Universitat Ramon Llull

TESI DOCTORAL

Títol: **Marc integrador de les capacitats de *Soft-Computing* i de *Knowledge Discovery* dels Mapes Autoorganitzatius en el Raonament Basat en Casos.**

Realitzada per l'**Albert Fornells Herrera**

en el Centre d'**Enginyeria i Arquitectura La Salle**

i en el Departament d'**Informàtica**

Dirigida per la **Dra. Elisabet Golobardes i Ribé**

C. Claravall, 1-3
08022 Barcelona
Tel. 936 022 200
Fax 936 022 249
E-mail: urlsc@sec.url.es
www.url.es

Resum

El Raonament Basat en Casos (CBR) és un paradigma d'aprenentatge basat en establir analogies amb problemes prèviament resolts per resoldre'n de nous. Per tant, l'organització, l'accés i la utilització del coneixement previ són aspectes claus per tenir èxit en aquest procés. No obstant, la majoria dels problemes reals presenten grans volums de dades complexes, incertes i amb coneixement aproximat i, conseqüentment, el rendiment del CBR pot veure's minvat degut a la complexitat de gestionar aquest tipus de coneixement. Això ha fet que en els últims anys hagi sorgit una nova línia de recerca anomenada *Soft-Computing and Intelligent Information Retrieval* enfocada en mitigar aquests efectes. D'aquí neix el context d'aquesta tesi.

Dins de l'ampli ventall de tècniques *Soft-Computing* per tractar coneixement complex, els Mapes Autoorganitzatius (SOM) destaquen sobre la resta per la seva capacitat en agrupar les dades en patrons, els quals permeten detectar relacions ocultes entre les dades. Aquesta capacitat ha estat explotada en treballs previs d'altres investigadors, on s'ha organitzat la memòria de casos del CBR amb SOM per tal de millorar la recuperació dels casos.

La finalitat de la present tesi és donar un pas més enllà en la simple combinació del CBR i de SOM, de tal manera que aquí s'introdueixen les capacitats de *Soft-Computing* i de *Knowledge Discovery* de SOM en totes les fases del CBR per nodrir-les del nou coneixement descobert. A més a més, les mètriques de complexitat apareixen en aquest context com un instrument precís per modelar el funcionament de SOM segons la tipologia de les dades. L'assoliment d'aquesta integració es pot dividir principalment en quatre fites: (1) la definició d'una metodologia per determinar la millor manera de recuperar els casos tenint en compte la complexitat de les dades i els requeriments de l'usuari; (2) la millora de la fiabilitat de la proposta de solucions gràcies a les relacions entre els clústers i els casos; (3) la potenciació de les capacitats explicatives mitjançant la generació d'explicacions simbòliques; (4) el manteniment incremental i semi-supervisat de la memòria de casos organitzada per SOM. Tots aquests punts s'integra sota la plataforma SOMCBR, la qual és extensament avaluada sobre *datasets* provinents de l'*UCI Repository* i de dominis mèdics i telemàtics.

Addicionalment, la tesi aborda de manera secundària dues línies de recerca fruït dels requeriments dels projectes on ha estat ubicada. D'una banda, s'aborda la definició de funcions de similitud específiques per definir com comparar un cas resolt amb un de nou mitjançant una variant de la Computació Evolutiva anomenada Evolució de Gramàtiques (GE). D'altra banda, s'estudia com definir esquemes de cooperació entre sistemes heterogenis per millorar la fiabilitat de la seva resposta conjunta mitjançant GE. Ambdues línies són integrades en dues plataformes, BRAIN i MGE respectivament, i són també avaluades amb els *datasets* anteriors.

Resumen

El Razonamiento Basado en Casos (CBR) es un paradigma de aprendizaje basado en establecer analogías con problemas previamente resueltos para resolver otros nuevos. Por tanto, la organización, el acceso y la utilización del conocimiento previo son aspectos clave para tener éxito en dicho proceso. No obstante, la mayoría de los problemas presentan grandes volúmenes de datos complejos, inciertos y con conocimiento aproximado y, consecuentemente, el rendimiento del CBR puede verse afectado debido a la complejidad de gestionar este tipo de conocimiento. Esto ha hecho que en los últimos años haya surgido una nueva línea de investigación llamada *Soft-Computing and Intelligent Information Retrieval* focalizada en mitigar estos efectos. Es aquí donde nace el contexto de esta tesis.

Dentro del amplio abanico de técnicas *Soft-Computing* para tratar conocimiento complejo, los Mapas Autoorganizativos (SOM) destacan por encima del resto por su capacidad de agrupar los datos en patrones, los cuales permiten detectar relaciones ocultas entre los datos. Esta capacidad ha sido aprovechada en trabajos previos de otros investigadores, donde se ha organizado la memoria de casos del CBR con SOM para mejorar la recuperación de los casos.

La finalidad de la presente tesis es dar un paso más en la simple combinación del CBR y de SOM, de tal manera que aquí se introducen las capacidades de *Soft-Computing* y de *Knowledge Discovery* de SOM en todas las fases del CBR para alimentarlas del conocimiento nuevo descubierto. Además, las métricas de complejidad aparecen en este contexto como un instrumento preciso para modelar el funcionamiento de SOM en función de la tipología de los datos. La consecución de esta integración se puede dividir principalmente en cuatro hitos: (1) la definición de una metodología para determinar la mejor manera de recuperar los casos teniendo en cuenta la complejidad de los datos y los requerimientos del usuario; (2) la mejora de la fiabilidad en la propuesta de soluciones gracias a las relaciones entre los clusters y los casos; (3) la potenciación de las capacidades explicativas mediante la generación de explicaciones simbólicas; (4) el mantenimiento incremental y semi-supervisado de la memoria de casos organizada por SOM. Todos estos puntos se integran en la plataforma SOMCBR, la cual es ampliamente evaluada sobre *datasets* procedentes del *UCI Repository* y de dominios médicos y telemáticos.

Adicionalmente, la tesis aborda de manera secundaria dos líneas de investigación fruto de los requerimientos de los proyectos donde ha estado ubicada la tesis. Por un lado, se aborda la definición de funciones de similitud específicas para definir como comparar un caso resuelto con uno de nuevo mediante una variación de la Computación Evolutiva denominada Evolución de Gramáticas (GE). Por otro lado, se estudia como definir esquemas de cooperación entre sistemas heterogéneos para mejorar la fiabilidad de su respuesta conjunta mediante GE. Ambas líneas son integradas en dos plataformas, BRAIN y MGE respectivamente, las cuales son también evaluadas sobre los *datasets* anteriores.

Abstract

Case-Based Reasoning (CBR) is an approach of machine learning based on solving new problems by identifying analogies with other previous solved problems. Thus, organization, access and management of this knowledge are crucial issues for achieving successful results. Nevertheless, the major part of real problems presents a huge amount of complex data, which also presents uncertain and partial knowledge. Therefore, CBR performance is influenced by the complex management of this knowledge. For this reason, a new research topic has appeared in the last years for tackling this problem: Soft-Computing and Intelligent Information Retrieval. This is the point where this thesis was born.

Inside the wide variety of Soft-Computing techniques for managing complex data, the Self-Organizing Maps (SOM) highlight from the rest due to their capability for grouping data according to certain patterns using the relations hidden in data. This capability has been used in a wide range of works, where the CBR case memory has been organized with SOM for improving the case retrieval.

The goal of this thesis is to take a step up in the simple combination of CBR and SOM. This thesis presents how to introduce the Soft-Computing and Knowledge Discovery capabilities of SOM inside all the steps of CBR to promote them with the discovered knowledge. Furthermore, complexity measures appear in this context as a mechanism to model the performance of SOM according to data topology. The achievement of this goal can be split in the next four points: (1) the definition of a methodology for setting up the best way of retrieving cases taking into account the data complexity and user requirements; (2) the improvement of the classification reliability through the relations between cases and clusters; (3) the promotion of the explaining capabilities by means of the generation of symbolic explanations; (4) the incremental and semi-supervised case-based maintenance. All these points are integrated in the SOMCBR framework, which has been widely tested in datasets from UCI Repository and from medical and telematic domains.

Additionally, this thesis secondly tackles two additional research lines due to the requirements of a project in which it has been developed. First, the definition of similarity functions ad hoc a domain is analyzed using a variant of the Evolutionary Computation called Grammar Evolution (GE). Second, the definition of cooperation schemes between heterogeneous systems is also analyzed for improving the reliability from the point of view of GE. Both lines are developed in two frameworks, BRAIN and MGE respectively, which are also evaluated over the last explained datasets.

'En lo puro no hay futuro
la pureza está en la mezcla
en la mezcla de lo puro
que antes que puro fue mezcla'

[Pau Donés, component del grup 'Jarabe de Palo']

Agraïments

Després de quatre llargs anys ha arribat el moment més esperat per 'tots', en el qual la família i els amics semblen tenir més ganes que un mateix de lliurar la tesi. Gràcies Carmen i pares pel vostre suport i paciència, us puc garantir que tinc més ganes jo que vosaltres que això s'acabi.

Al llarg de tot aquest temps he tingut la sort de conèixer i poder treballar amb un grup de persones fantàstic, gràcies a les quals aquest 'vaixell' en forma de tesi ha pogut arribar al seu port anomenat 'defensa'. Per aquest motiu, m'agradaria agrair a totes les persones que han contribuït d'alguna manera en aquesta tesi amb el seu granet de 'carbó'.

Primer de tot, vull agrair al Pep Martorell, a la Guiomar Corral, al Joan Camps, al Josep Maria Garrell, a l'Ester Bernadó, a la Núria Macià, al David Vernet, a l'Eva Armengol, al Xavier Vilasís i al Joan Martí la feina feta als articles publicats. Al Franscesc Teixidó i a l'Albert Orriols que hagin muntat el clúster 'Coronita'. Al David Nettleton i a la Tallulah Foster els hi agraeixo molt els seus consells per polir l'anglès dels articles. Finalment, l'agraïment més gran el vull dedicar a la persona que ha dirigit el 'vaixell' en aquests quatre anys. Elisabet, ha sigut un privilegi haver-te tingut com a amiga i com a directora de tesi. No té preu el teu sacrifici per estar sempre disponible, tot i el teu infinit '*inbox*' i la teva complexa agenda.

També voldria agrair al Departament d'Universitats, Recerca i Societat de la Informació (DURSI) el seu suport mitjançant una beca per a la formació de personal investigador (2004FIR 00365, 2005FIR 00237, 2006 FIC 00433, 2007 FIC 00976), així com el suport per part del '*Ministerio de Ciencia y Tecnología*' amb els projectes HRIMAC (TIC 2002-04160-C/02-02), ANALIA (CIT-390000-2005-27) i MID-CBR (TIN 2006-15140-C03-03). Tots aquests suports no haurien estat possibles sense la col·laboració d'Enginyeria i Arquitectura La Salle, i per tant, agraeixo molt el seu suport i ajut durant tot aquest temps.

Índex

Índex de figures

Índex de taules

Índex d'algorismes

Capítol 1

Introducció

El Raonament basat en casos és una branca de la Intel·ligència artificial basada en resoldre problemes nous a partir d'altres prèviament resolts i, per tant, un dels aspectes claus és la gestió de la seva experiència. La finalitat de la present tesi és potenciar l'organització i l'accés d'aquest coneixement per construir sistemes CBR més robusts i ràpids davant de grans volums de dades complexes i incertes. Per aconseguir aquesta fita, es planteja l'organització de la memòria mitjançant una tècnica connexionista no supervisada de clustering anomenada Mapa Autoorganitzatiu o de Kohonen, a través de la qual s'introduiran les seves capacitats *Soft-Computing* i de *Knowledge Discovery* per nodrir totes les fases del CBR del coneixement amagat a les dades del problema.

1.1 Marc de treball

El programa de doctorat on s'emmarca la meva recerca és el de les **Tecnologies de la Informació i les Comunicacions i la seva gestió** d'**Enginyeria i Arquitectura La Salle** (EALS) de la **Universitat Ramon Llull** (URL). Concretament, la seva realització ha estat dins el **Grup de Recerca en Sistemes Intel·ligents** (GRSI).

El GRSI és un grup de recerca creat a l'any 1994 dins d'EALS que centra la seva recerca al voltant de la Intel·ligència artificial i, més concretament, a l'aprenentatge artificial a través dels paradigmes del Raonament basat en casos, la Computació evolutiva i el *Soft-Computing* en general. L'activitat del grup es focalitza principalment en resoldre problemes de classificació, diagnòstic, i predicció en diferents àmbits, on destaquen els entorns mèdics i telemàtics. A més a més, l'activitat del grup i la seva consolidació van ser reconeguts per la Generalitat de Catalunya a l'any 2002 (2002 SGR-155) i revalidada al 2005 (2005 SGR-302).

La meva col·laboració en el GRSI va començar a l'any 2000. El motiu de començar a treballar amb ells va ser que des de petit sempre havia tingut curiositat per conèixer com era possible integrar comportaments intel·ligents en els ordinadors, sobretot arrel de sèries com la protagonitzada per David Hasselhoff en el cotxe fantàstic. La col·laboració va desembocar en la realització del meu treball final de carrera, el qual va estar centrat en l'estudi de la rellevància dels atributs de les dades. El treball va estar dirigit per la **Dra. Maria Salamó**. També va ser dins el marc del GRSI on vaig realitzar el projecte final de carrera. En aquest cas, es va desenvolupar un entorn docent per l'assignatura d'*Intel·ligència artificial* que permetés als seus alumnes desenvolupar jugadors d'escacs 'intel·ligents'. El projecte va ser dirigit per la **Dra. Elisabet Golobardes**, la qual des de llavors ha estat la meva tutora i directora de tesi.

Al llarg del doctorat he participat activament en els projectes **HRIMAC** (*Herramienta de Recuperación de Imágenes Mamográficas por Análisis de Contenido para el asesoramiento en el diagnóstico del cáncer de mama*, TIC 2002-04160-C02-02) i **MID-CBR** (*Un Marco Integrador*

para el Desarrollo de Sistemas de Razonamiento Basado en Casos, TIN 2006-15140-C03-03), i he col·laborat puntualment en el projecte **ANALIA** (CIT-390000-2005-27). Els tres projectes han estat finançats pel ***Ministerio de Ciencia y Tecnología***.

Finalment, la meva ubicació dins d'aquest marc de treball no hauria estat possible sense el suport que he tingut aquests quatre anys per part del **Departament d'Universitats, Recerca i Societat de la Informació** (DURSI) mitjançant una beca per a la formació de personal investigador (2004FI00365, 2005FIR00237, 2006FIC-0043, 2007FIC-00976), al *Ministerio de Ciencia y Tecnología* pel seu finançament en els esmentats projectes, així com a l'ajuda i suport que he rebut dels membres del GRSI, d'EALS i de la URL.

Aquest capítol introductori presenta el marc de recerca on va néixer la motivació d'aquesta tesi, així com els objectius que van fixar-se. El capítol finalitza amb una descripció de l'estructura d'aquesta memòria.

1.2 Marc de recerca

Una de les grans diferències entre les persones i la resta d'éssers vius és la capacitat per raonar, la qual ens permet pensar, avaluar i actuar segons certs principis per assolir una certa fita. Davant d'un problema nou o una situació conflictiva, una persona busca en la seva experiència situacions semblants per poder plantejar una resposta que s'adapti, amb més o menys èxit, al nou escenari. Aquest procés és precisament el fonament bàsic d'una de les famílies de l'aprenentatge artificial: l'**Aprenentatge analògic**.

1.2.1 L'Aprenentatge analògic

L'Aprenentatge analògic engloba les tècniques de la Intel·ligència artificial fonamentades en la identificació d'analogies entre problemes nous i d'altres prèviament resolts, per resoldre'ls a través d'un procés basat en extrapolar els passos o la metodologia aplicada anteriorment. Segons la manera com es seleccionin les situacions resoltes, es comparin respecte el nou problema, i es desenvolupi la nova solució entre d'altres aspectes, poden identificar-se diferents variants:

Raonament basat en exemples. Realitza tasques de classificació a partir d'exemples (Porter, 1986).

Raonament basat en instàncies. Classifica mitjançant conceptes que crea (Aha i Kibler, 1991).

Raonament basat en memòries. Es centra en la cerca d'informació de manera paral·lela (Kitano, 1993).

Raonament basat en casos. Modifica i adapta les solucions prèvies mitjançant un coneixement de fons. Aquests tipus de nuclis estan basats en regles i teories psicològiques (Riesbeck i Schank, 1989).

Raonament basat en analogies. Són sistemes basats en la reutilització d'informació. Resolen problemes nous a partir d'altres prèviament resolts, però que poden pertànyer a dominis diferents (Cabelli, 1988; Hall, 1989; Veloso i Carbonell, 1993a).

La variant en la qual es centra aquesta tesi és la del Raonament Basat en Casos (*Case-Based Reasoning* - CBR) (Kolodner, 1993).

1.2.2 El raonament basat en casos

Les bases del CBR en la Intel·ligència artificial es troben en els treballs realitzats per Roger Schank (Schank, 1982) sobre memòries dinàmiques i les tasques de recordar patrons i situacions anteriors per resoldre problemes nous i aprendre d'ells.

La figura 1.1 il·lustra una situació real on és necessari aplicar un raonament per resoldre un problema nou. Imaginem que un expert ha d'analitzar una mamografia d'un pacient per determinar el risc de ser cancerígena. De manera general, el raonament o metodologia que segueix l'expert pot descomposar-se en els passos següents:

1. L'expert detecta les característiques més rellevants de la mamografia amb la finalitat de caracteritzar el problema nou (**fase de preprocessament**). El conjunt de les característiques detectades permeten descriure la situació (**cas**).

2. L'expert cerca mamografies diagnosticades prèviament, tant per ell com pels llibres de medicina, que tinguin característiques semblants a les del nou problema (**fase de recuperació**).

3. Emet un diagnòstic tenint en compte els diagnòstics del conjunt de mamografies que havia recopilat com a semblants (**fase d'adaptació**).

4. Demana una segona opinió del diagnòstic a companys seus amb la finalitat de validar el seu diagnòstic (**fase de revisió**).

5. A partir del diagnòstic realitzat pren les notes pertinents per recordar el cas nou resolt, ja que aquest li pot ser útil en un futur (**fase d'emmagatzematge**).

Al marge de la fase de preprocessament que és comuna per qualsevol mètode d'aprenentatge, els altres quatre punts descriuen les quatre fases del cicle de vida del CBR (Aamodt i Plaza, 1994). A més a més, de la mateixa manera que les persones guarden les seves experiències al cervell, aquests sistemes disposen d'una estructura anomenada **memòria de casos** on guarden els casos prèviament resolts. Per tant, pot afirmar-se que aquest tipus d'aprenentatge esquematitza bastant bé el procés de raonament de les persones.

1.2.3 El paper clau de l'experiència en el raonament basat en casos

La figura 1.2 il·lustra el cicle de vida del CBR proposat per Aamodt&Plaza (Aamodt i Plaza, 1994) a través del qual es relacionen les fases introduïdes a l'apartat anterior. L'èxit del raonament, tant

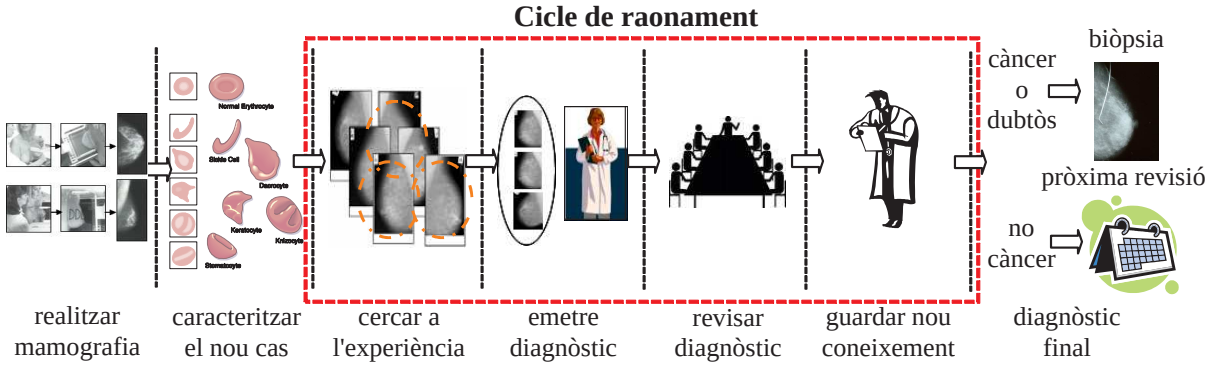

Figura 1.1: L'emissió d'un diagnòstic està fortament arrelat a l'experiència de l'expert, la qual constitueix la base a partir de la qual es pren la decisió. Aquest és precisament el fonament del CBR.

de l'expert com del sistema, està directament lligat a la capacitat d'ambdós per portar a bon terme cadascuna de les quatre fases. Els aspectes claus a tenir en compte a cadascuna d'elles són:

Criteri de recuperació. No tota experiència és útil. Determinar quins casos de l'experiència són seleccionats per ser avaluats, així com decidir el criteri per avaluar la similitud entre el cas nou i els resolts prèviament, és vital per obtenir la informació base a partir de la qual s'afronta el problema nou.

Criteri de fiabilitat. Cada domini té una complexitat implícita i un nivell de risc propi segons el 'preu a pagar' per equivocar-se. És vital establir mecanismes per a la definició de criteris que ajudin a garantir la fiabilitat de la proposta.

Criteri de validació. La validació d'una solució requereix de la intervenció d'un expert. No obstant, en molts dominis l'expert només pot donar arguments a favor o en contra de la solució, sense assegurar categòricament el resultat. Per tant, és necessari que l'expert disposi d'eines per entendre el perquè el problema s'ha resolt d'una determinada manera.

Criteri de manteniment del coneixement. La capacitat de resoldre problemes està fortament lligada a l'experiència de la que es disposa. Cal vetllar per mantenir consistent aquest coneixement, tant incloent casos nous resolts, com eliminant aquells que confonen al sistema perquè tenen soroll, o bé, ja no són certs perquè el domini ha canviat.

Tots els punts anteriors tenen el mateix denominador comú: l'experiència del sistema emmagatzemada a la memòria de casos. Les propietats desitjables que hauria de tenir aquesta memòria de casos són les següents:

Compacta. No ha de contenir casos redundants ni amb soroll perquè poden distorsionar la realitat i confondre al sistema en el procés de recuperació dels elements més semblants.

Representativa. No es pot resoldre tot allò del que no es té constància. És necessari disposar de casos representatius dels diferents aspectes característics del domini per tal de no tenir una visió parcial de la realitat.

Reduïda. La velocitat amb la qual el sistema respon està relacionada amb el nombre d'elements dels quals disposa. La mida de la memòria ha de permetre la resposta del sistema en un temps raonable segons les restriccions de l'àmbit d'aplicació.

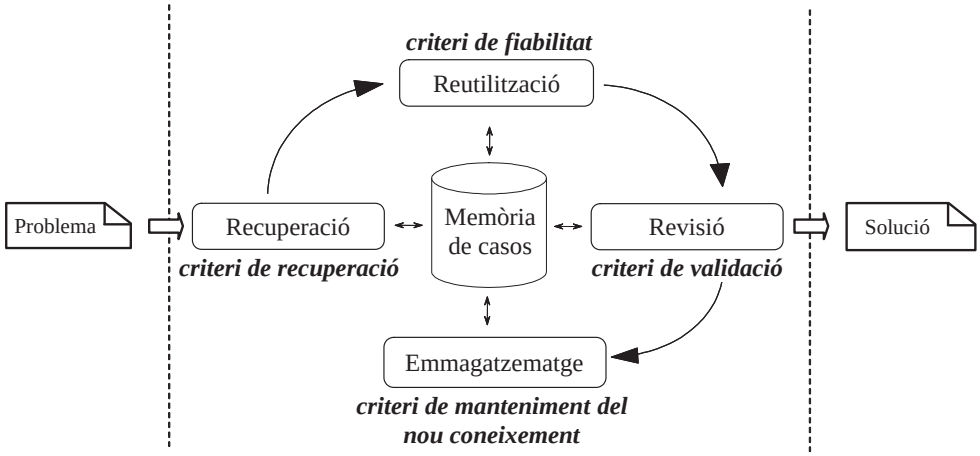

Figura 1.2: Aspectes claus dins el cicle del CBR proposat per Aamodt&Plaza (Aamodt i Plaza, 1994). Al llarg de la tesi ens referirem a la fase de reutilització en un sentit més ampli com a fase d'adaptació.

Aquestes tres propietats estan estretament relacionades entre elles, podent-se resumir en la premissa següent: **disposar del conjunt mínim de casos independents capaços de representar completament el domini**. Per tant, el contingut, l'organització, i l'accés de la memòria de casos determina el rendiment del sistema en termes de la capacitat per resoldre problemes nous, així com del temps necessari en proporcionar la resposta.

No obstant, sovint la realitat amb la que ens enfrontem no ens permet disposar de memòries amb aquestes característiques, degut a que la realitat normalment és complexa, imprecisa, i parcialment desconeguda. Per aquest motiu, en els últims anys ha sorgit una línia de recerca focalitzada en l'aplicació d'estratègies provinents del camp del *Knowledge Discovery* per potenciar el contingut, l'estructura i l'accés a la memòria de casos gràcies a les capacitats d'aquest tipus de tècniques, a través de les quals és possible:

Determinar la rellevància dels atributs per conèixer quins aspectes de les dades són importants, i quins es poden ignorar ja que només introdueixen soroll.

Entendre les relacions entre les dades per identificar, detectar i esborrar redundàncies, així com possibles inconsistències.

Identificar patrons dins les dades per tal d'indexar o jerarquitzar les seves relacions.

A la literatura poden trobar-se moltes estratègies aplicades per a dur a terme això des de diferents punts de vista i paradigmes. D'una banda, estan les estratègies que organitzen els casos mitjançant estructures de dades on els valors possibles dels atributs són els protagonistes. *K-d trees* (Wess et al., 1994) defineix una estructura organitzativa en forma d'arbre on cada node representa cadascun dels atributs. Segons el valor dels atributs va creant subarbres fins que hi ha tants nivells com atributs. Al nivell de les fulles es troben els casos agrupats segons els valors dels atributs. El principal desavantatge d'aquesta agrupació és el tractament dels valors desconeguts degut a l'estructura en forma d'arbre que obliga a tenir els valors dels atributs per arribar a les fulles. Aquesta limitació es solventa amb organitzacions basades en estructures en forma de graf. *Case Retrieval Nets* (Lenz et al., 1996) defineix un graf on cada node representa una combinació atribut-valor. La selecció dels casos es produeix a partir d'un procés d'activació que propaga la similitud entre els valors dels atributs del nou cas respecte els nodes, i entre els propis nodes del graf. *Decision Diagrams* (Nicholson et al., 2006) segueix l'estratègia *K-d trees* però amb un graf directe que solventa les restriccions dels arbres.

D'altra banda, les relacions també poden definir-se a nivell de cas, com per exemple relacionar els casos segons la similitud entre ells (Schaaf, 1995; Yang i Wu, 2001), o bé, a partir de similituds definides a partir del coneixement del domini com en el sistema CRASH (Brown, 1994).

Un punt de vista diferent als anteriors és el d'organitzar la memòria a través de la definició d'índexs o patrons fent servir un ús intensiu del coneixement. Un exemple d'això és el sistema BankXX (Rissland et al., 1993), on es defineixen conceptes legals per realitzar les cerques heurístiques. D'altra banda, els índexs es poden construir a partir d'un ús restringit del coneixement, és a dir, a partir de l'anàlisi intensiu de les dades del problema. Un exemple d'aquesta vessant és el sistema ULIC (Vernet i Golobardes, 2003), on els índexs són els centroides generats a partir de l'algorisme de clustering X-means (Pelleg i Moore, 2000).

En qualsevol de les diferents estratègies citades i, independentment del punt de vista utilitzat, el que es busca sempre és el mateix: filtrar els casos que no són interessants per millorar (1) la capacitat resolutiva i (2) el temps de resposta, perquè només es fa servir la part de l'experiència útil i rellevant.

1.3 Motivació: dominis complexos i amb coneixement incert

La millor manera de tractar dominis complexos[1] que presenten incertesa és fer-ho a través de tècniques que hagin estat pensades per treballar amb aquesta tipologia de dades. Les tècniques *Soft-Computing* − les quals engloben principalment les famílies de la Lògica difusa (Zadeh, 1965), els *Rough Sets* (Pawlac, 1991), la Computació evolutiva (Holland, 1975), el Raonament probabilístic (Pearl, 1988) i les Xarxes neuronals (Bishop, 1995) − a diferència de les tècniques tradicionals de *Hard Computing*, permeten el tractament de coneixement imprecís, incert, parcialment vertader i aproximat, és a dir, amb el tipus de coneixement de la majoria dels problemes reals. Per aquest motiu, s'ha fet necessari incorporar el seu ús a l'hora de gestionar grans volums de dades complexes i incertes, prenent força una nova línia de recerca anomenada *Soft-Computing and Intelligent Information Retrieval* (Crestani i Pasi, 2000; Cordón i Herrera, 2003).

Els avantatges d'aquest tipus de tècniques s'han posat de manifest en treballs previs realitzats dins el GRSI a l'hora d'abordar alguns aspectes puntuals del CBR davant de situacions complexes:

Definició de funcions de similitud. La fase de recuperació compara el grau de semblança entre el cas nou i el cas resolt a través d'una mètrica anomenada funció de similitud. La dificultat resideix en què les funcions tradicionals no funcionen igual de bé en tots els dominis. A (Camps et al., 2003; Golobardes et al., 2001) es va proposar aprofitar les capacitats *Soft-Computing* de la Computació evolutiva per definir automàticament funcions de similitud específiques a un domini.

Ponderació d'atributs. No totes les característiques que descriuen un problema tenen la mateixa rellevància. A (Salamó et al., 2000) va estudiar-se com aprofitar la teoria dels *Rough Sets* per determinar la importància dels atributs.

Manteniment de la memòria de casos. No tots els casos que es resolen han de guardar-se, ja que sinó les propietats de la memòria de casos es degraden si s'introdueix soroll o casos redundants. A (Salamó i Golobardes, 2001; Salamó i Golobardes, 2002; Salamó i Golobardes, 2003b; Salamó i Golobardes, 2003a; Salamó i Golobardes, 2004a; Salamó i Golobardes, 2004b; Salamó i Golobardes, 2004c) es va avaluar la teoria dels *Rough Sets* com una mètrica per avaluar la rellevància dels casos dins de la memòria de casos, per determinar quan és interessant emmagatzemar-los, no tenir-los en compte, o bé, convé esborrar casos antics.

Per tant, la seva utilització per organitzar la memòria de casos del CBR sembla un bon punt de partida per iniciar el camí cap a la construcció de sistemes CBR més robusts i tolerants al soroll ja que serà possible recuperar la part de l'experiència més adient per resoldre problemes nous.

Dins de l'ampli ventall de tècniques amb capacitats *Soft-Computing* i de *Knowledge Discovery* destaquen els Mapes de Kohonen o Mapes Auto-Organitzatius (*Self-Organizing Maps* - SOM) (Kohonen, 1984), els quals són una de les tècniques més utilitzades (Kaski et al., 1998b; Oja et al., 2003) dins el camp de les Xarxes neuronals. SOM és una tècnica no supervisada de clustering basada en identificar grups de dades (clústers) a través de la construcció d'un patró o model que reflecteix les característiques que tenen en comú el conjunt de casos. Aquesta propietat és molt útil per definir les relacions entre les dades i, d'aquesta manera, indexar el contingut de la memòria de casos per (1) seleccionar només els casos interessants i evitar tenir en compte casos redundants; i (2) reduir el temps computacional del sistema perquè només es fa servir una part de l'experiència.

La utilització de les xarxes neuronals per millorar el rendiment del CBR no és nou. A la literatura poden trobar-se aplicacions en entorns per a la predicció de bancarrotes (Hongkyu i Ingoo, 1996), pel diagnòstic i solució online de caigudes de sistemes informàtics (Jha et al., 1999),

[1] La complexitat de les dades es refereix a la separabilitat de classes i al poder discriminant dels atributs, i no a la seva representació en estructures de dades.

pel càlcul del cost de *software* (Carolyn et al., 2000), o per pressupostar projectes de construcció (Essam i Ahmed, 2001). En el cas específic de combinar el CBR amb SOM, s'han combinat per la predicció de valors d'accions (Kim i Han, 2001), per la predicció de venta de llibres (Chang i Lai, 2005), o bé, per tasques de validació d'estructures (Mujica et al., 2005). Per tant, la combinació de sistemes CBR amb algorismes de les xarxes neuronals és una unió que ha tingut força èxit. No obstant, els treballs que hi ha a la literatura només fan referència a sistemes que fan servir SOM en una part concreta del CBR, normalment només a la fase de recuperació, per resoldre problemes específics com en els casos de les cites anteriors. Això fa que hi hagi un ampli conjunt d'aspectes 'oblidats' a la literatura, els quals seria interessar abordar:

Fase de recuperació. L'organització de la memòria de casos amb SOM permet indexar el seu contingut per realitzar recuperacions selectives. L'estratègia de recuperació consisteix en cercar el clúster que millor modela el cas d'entrada i, a continuació, fer servir només els casos del clúster seleccionat per aplicar el CBR. Gràcies a això s'aconsegueix millorar notablement el temps de la fase de recuperació, el qual és directament proporcional a la mida de la memòria de casos explorada. A més a més, el mètode evita seleccionar casos sorollosos o que no tenen res a veure amb l'exemple d'entrada gràcies a la definició dels patrons.

Aquesta estratègia suposa que la informació que es busca es troba sempre al clúster seleccionat com a millor. No obstant, això no ha de perquè succeir si els clústers definits no són prou representatius degut a la incertesa del domini. Davant d'això que cal fer? Seleccionar més clústers? Utilitzar un percentatge de casos de cada clúster? No hi ha cap procediment que estableixi els criteris per seleccionar el nombre òptim de clústers ni de casos a recuperar d'aquests. Cal una metodologia que ajudi a decidir aquests criteris segons les necessitats de l'usuari i la capacitat dels clústers per representar la complexitat de les dades.

Fase d'adaptació. Aquesta fase és l'encarregada de proposar la nova solució a partir dels casos recuperats a la fase anterior. En aquest cas, la fase d'adaptació es realitza fent servir els casos recuperats i sense que SOM intervingui.

Perquè no s'aprofita la relació entre els casos i el patró del clúster al que pertany per definir graus de similitud o de pertinença entre ells com es fa a la lògica difusa? L'anàlisi d'aquesta relació pot ajudar a saber quins casos són 'robusts' i quins són 'incerts' respecte al patró que els modela, aspecte que pot millorar la fiabilitat de la proposta de solucions. Aquest aspecte és molt important sobretot en dominis crítics, com per exemple a l'àmbit de la medicina on el preu dels errors són habitualment massa alts.

Fase de revisió. La fase valida mitjançant un expert si la solució proposada és correcte. El problema resideix en què els experts sovint tenen moltes dificultats per realitzar aquesta tasca perquè no coneixen la resposta (si la coneguessin no caldria un CBR que els hi proposés). Per això, és important dotar al sistema de mecanismes que l'ajudin a prendre aquesta decisió.

Dins del context de revisió/anàlisi de dades, SOM té un paper molt destacat a la literatura gràcies a la seva capacitat per agrupar dades. Aquesta capacitat és aprofitada pels experts per analitzar visualment les dades, i extreure'n conclusions respecte les seves relacions. Seria interessant dotar als sistemes CBR que organitzen la memòria amb SOM, de la capacitat d'oferir explicacions als usuaris respecte el perquè les dades s'agrupen de determinada manera. Això permetria a l'usuari entendre millor les relacions entre les dades i, conseqüentment, comprendre perquè certes dades són recuperades.

Fase d'emmagatzematge. A partir de les fases anteriors s'obté la solució correcta al nou problema, quedant com a últim pas decidir si el cas resolt s'ha de guardar o no a l'experiència per tal de millorar les futures prediccions.

Les aplicacions on es fa servir SOM per organitzar la memòria de casos no contemplen la integració de coneixement nou de manera incremental i supervisada perquè l'entrenament de SOM és no supervisat i no incremental. L'única solució és reentrenar el mapa des de zero amb la conseqüent despesa computacional. Per tant, calen estratègies per mantenir el coneixement de la memòria de manera incremental i amb l'ajuda de l'expert.

És a partir d'aquestes necessitats no cobertes d'on neix la present tesi.

1.4 Objectius de la tesi

La finalitat de la tesi és analitzar, definir i implementar un marc complert que integri SOM en tots els aspectes del cicle de vida del CBR, i on les capacitats referents al *Knowledge Discovery* i al *Soft-Computing* de SOM s'aprofitin per construir sistemes CBR més robusts i tolerants al soroll. El fet de nodrir al sistema del coneixement descobert de les relacions ocultes entre les dades permetrà al CBR adaptar-se millor a les necessitats de cada problema.

Tenint en compte les reflexions de l'apartat anterior, la figura 1.3 il·lustra el marc integrador de SOM en el CBR, anomenat SOMCBR, on cada fase del CBR aprofita les capacitats de SOM. Per portar a terme la definició d'aquest marc cal assolir les fites següents:

Definir una metodologia per establir com recuperar els casos segons els requeriments de l'usuari i la complexitat de les dades. SOM segmenta la topologia de les dades en grups de casos que comparteixen propietats semblants i, on cada grup està representat per un patró que modela aquell conjunt. Aquest darrer element és el que permet indexar el contingut de la memòria.

Aquest punt pretén definir una metodologia que, tenint en compte les diferents maneres de recuperar els casos i el seu rendiment associat, ens ajudi a conèixer sota quina tipologia de dades l'estratègia proporcionarà els resultats desitjats.

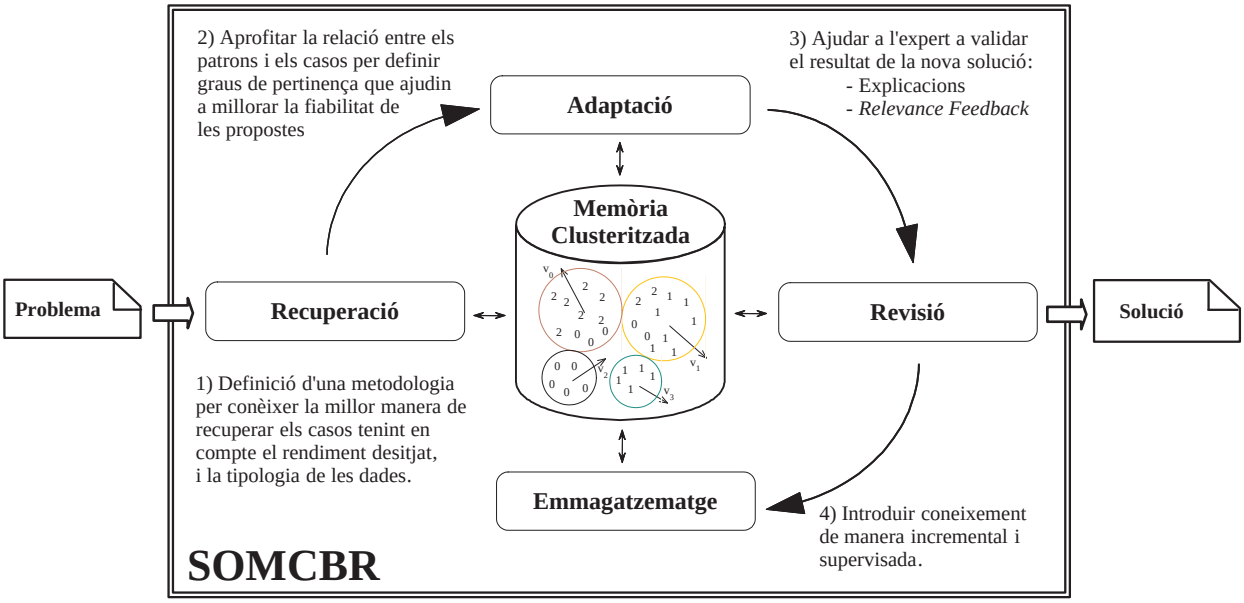

Figura 1.3: La tesi pretén definir un marc integrador, anomenat SOMCBR, que aprofiti les capacitats de *Soft-Computing* i de *Knowledge Discovery* de SOM per potenciar totes les fases del CBR.

Incrementar la fiabilitat de la predicció a partir dels veïns més robusts. La proposta de la nova solució depèn dels casos recuperats a la fase anterior. No obstant, no tots els casos són igual d'importants.

Aquest punt centra el seu objectiu en definir una estratègia per proposar solucions basades en el grau de pertinença dels casos respecte el patró que els modela per tal de detectar quins casos són més fiables i, conseqüentment, han de considerar-se més rellevants a l'hora de proposar la nova solució.

Suport a l'expert en el procés de revisió dels resultats. La figura de l'expert és vital a la fase de revisió per tal de poder validar el resultat.

Aquest punt té com a finalitat ajudar a l'expert a comprendre els motius pels quals determinats casos són recuperats a partir de la generació d'explicacions simbòliques que descriguin les relacions entre les dades identificades per SOM.

Introducció i manteniment de nou coneixement en la memòria clusteritzada. El sistema va adquirint noves experiències a mesura que resolt problemes nous, aspecte que li permet millorar la seva futura capacitat resolutiva. No obstant, SOM no ha estat concebut per ser actualitzat amb coneixement nou.

La tasca en aquest cas consisteix en definir una estratègia per mantenir el coneixement de la memòria de manera incremental i que tingui en compte el *feedback* de l'expert. Des d'aquest punt de vista, el manteniment ha de comprendre tant la gestió dels casos, com del sistema d'indexació de SOM sense que les propietats del sistema es vegin reduïdes.

Implementació de les aportacions en un marc integrador anomenat SOMCBR. A partir dels punts anteriors, cal implementar una plataforma, anomenada SOMCBR (*Self-Organization Map in a Case-Based system*), que integri totes les contribucions anteriors. Aquest objectiu és transversal als anteriors ja que es desenvolupa paral·lelament.

Validació de les contribucions. Els punts implementats a la plataforma SOMCBR han de ser avaluats mitjançant un ampli joc de dades provinent de l'*UCI Repository* (Asuncion i Newman, 2007), així com de dades reals provinents de dos dels projectes on s'ha emmarcat la tesi:

HRIMAC (TIC2002-04160-C02-02). El projecte aborda el desenvolupament d'una eina per ajudar a l'expert en el procés de diagnòstic de càncer de mama. Concretament, l'eina ha de permetre recuperar imatges mamogràfiques de diferents bases de dades públiques segons certs criteris topològics. L'anàlisi d'aquests casos, ja patològics, ajuda al radiòleg a millorar la interpretació de la mamografia i, conseqüentment, a diagnosticar amb més garanties d'èxit. Una altra utilitat important és que permet exemplificar casos prototips de les diferents simptomologies registrades dins l'anàlisi d'imatges mamogràfiques, aspecte molt important per formar a nous experts. En aquest projecte es col·labora amb el departament de Visió per Computador de la Universitat de Girona i amb l'Hospital Universitari Dr. Josep Trueta.

MID-CBR (TIN2006-15140-C03-03). La finalitat del projecte és desenvolupar un marc conceptual que ajudi a definir com integrar sota un mateix marc els desenvolupaments de sistemes de Raonament basat en casos. En el projecte es col·labora amb l'Institut d'Investigació d'Intel·ligència Artificial (IIIA) del *Consejo Superior de Investigaciones Científicas* (CSIC) i amb la Universitat Complutense de Madrid (UCM). A més a més, el projecte compta amb el recolzament de dues EPO's. D'una banda, amb la *Fundació*

Clínic per a la Recerca Biomèdica del Hospital Clínico de Barcelona s'estudia el desenvolupament d'una eina per diagnosticar i/o establir pronòstics de càncer de melanoma. D'altra banda, amb l'empresa ISECOM es treballa amb el desenvolupament d'una eina per la detecció de vulnerabilitats telemàtiques.

ANALIA (CIT-390000-2005-27). L'objectiu és incorporar tècniques de la Intel·ligència artificial i de la mineria de dades als resultats proporcionats pel sistema CONSENSUS. CONSENSUS (FIT-360000-2004-81) és un sistema de detecció de vulnerabilitats desenvolupat per EALS amb la col·laboració de l'empresa ISECOM. L'eina està destinada a ajudar als professionals de la seguretat a detectar de forma automàtica les vulnerabilitats que poden existir en una xarxa, així com en els dispositius que la componen. CONSENSUS visualitza grans volums d'informació provinents de diferents testejos de seguretat, però no inclou prestacions addicionals que ajudin a l'expert en seguretat a la seva anàlisi. Per això, amb ANALIA es pretén millorar la fase d'anàlisi posterior al testeig i ajudar a l'analista de seguretat a l'extracció de conclusions mitjançant un processament previ de la informació obtinguda.

De la mateixa manera que l'objectiu anterior, aquest també és transversal als anteriors al desenvolupar-se en paral·lel amb els altres.

Al marge dels objectius anteriors vinculats amb la definició d'un marc que integri SOM en tots els aspectes del cicle del CBR, els requeriments del projecte HRIMAC han fet que s'hagin abordat de manera secundària dues línies de treball desvinculades al SOMCBR:

Disseny de funcions de similitud específiques al domini. Aquesta línia ja havia estat oberta anteriorment dins el grup. A la fase de recuperació cal definir com comparar un cas nou respecte un altre de la memòria de casos a través d'una funció de similitud, la qual mesura la seva semblança. No obstant, les funcions tradicionals no funcionen bé per a dominis complexos, i cal definir com comparar dos casos tenint en compte les particularitats del domini.

Disseny d'esquemes de cooperació específics al domini. Quan una persona ha de prendre una decisió important, demana l'opinió a d'altres per tal d'assegurar-se de prendre la decisió correcte. El mateix principi es pot traslladar al món de la Intel·ligència artificial. Aquest punt es centra en estudiar com definir esquemes de cooperació específics al domini per tal de millorar la fiabilitat dels resultats.

Ambdues línies es fonamenten en l'optimització de funcions a través d'una variant de la Computació evolutiva anomenada Evolució de gramàtiques (*Grammar Evolution* - GE) (Ryan et al., 1998). A banda de la definició teòrica d'una aproximació per abordar cada problemàtica, la seva validació s'ha fet a partir de *datasets* provinents de l'HRIMAC i l'*UCI Repository* (Asuncion i Newman, 2007), els quals han estat avaluades sobre les plataformes implementades BRAIN i MGE respectivament. Actualment aquestes línies de recerca són dues tesis en curs al GRSI. D'aquestes dues línies de recerca, a la redacció de la tesi només s'ha tingut en compte el disseny de funcions de similitud perquè és un aspecte relativament proper al CBR. En canvi, la segona línia al estar totalment desvinculada del CBR s'ha omés a la present memòria. Malgrat això, la feina feta en aquesta vessat es pot trobar redactada a l'article (Fornells et al., 2006a) adjuntat.

1.5 Estructura de la memòria

La memòria de la tesi s'estructura en cinc parts. La primera part engloba els fonaments teòrics dels tres paradigmes de la Intel·ligència artificial en els quals es fonamenta la tesi. Els capítols 2, 3

i 4 descriuen els principis del Raonament basat en casos, els Mapes autoorganitzatius, i la relació entre la Programació genètica i l'Evolució de gramàtiques respectivament.

La segona part engloba la definició, la implementació i l'avaluació del marc integrador de SOM en el CBR. Els capítols 5, 6, 7 i 8 fan referència a les contribucions de la fase de recuperació, d'adaptació, de revisió i d'emmagatzematge respectivament. El capítol 9 recull aplicacions directes dels conceptes estudiats en aquesta part sobre els projectes HRIMAC i ANALIA. Finalment, el capítol 10 descriu a alt nivell la plataforma SOMCBR.

La tercera part aborda el disseny de funcions de similitud específiques per un domini. El capítol 11 proposa i avalua el sistema híbrid entre les variants de la Computació evolutiva i el CBR. El capítol 12 descriu a alt nivell les plataformes implementades per abordar aquesta part.

La quarta part engloba la cloenda de la recerca, la qual està composada pel capítol 13 on es resumeixen les aportacions i el treball realitzat, així com les línies de treball futur.

Finalment, la cinquena i última part està composada per un conjunt d'apèndixs que recullen temes d'interès que estan relacionats amb alguns capítols de la memòria. L'apèndix A conté les diferents sigles emprades a la tesi. Els apèndixs B i C descriuen les dades dels projectes HRIMAC i ANALIA. L'apèndix D descriu diverses metodologies d'experimentació. L'apèndix E resumeix les tècniques de preprocessament dades aplicades en els experiments. L'apèndix F descriu les funcions de distància més habituals. L'apèndix G tracta la relació de les mètriques de complexitat i el SOMCBR. Finalment, l'apèndix H descriu els principals conceptes de les tècniques de *Relevance Feedback*.

Resum

El treball realitzat s'emmarca dins el Grup de Recerca en Sistemes Intel·ligents (GRSI) sota el programa de doctorat de les Tecnologies de la Informació i les Comunicacions i la seva gestió d'Enginyeria i Arquitectura La Salle (EALS) de la Universitat Ramon Llull (URL). La seva realització no hauria estat possible sense el suport del Departament d'Universitats, Recerca i Societat de la Informació (DURSI) mitjançant una beca per a la formació de personal investigador (2004FI00365, 2005FIR00237, 2006FIC-0043, 2007FIC-00976), al *Ministerio de Ciencia y Tecnología* pel seu finançament en els projectes HRIMAC, ANALIA i MID-CBR, així com a l'ajuda i suport que he rebut dels membres del GRSI, d'EALS i la URL.

El context de la tesi és el del Raonament Basat en Casos (CBR). El CBR és un paradigma basat en resoldre problemes nous a partir d'altres resolts prèviament. El seu rendiment, en termes de capacitat resolutiva i temps de resposta, està molt lligat a l'experiència representada mitjançant una estructura anomenada memòria de casos. Per tant, la gestió d'aquest coneixement és un aspecte vital per la construcció de sistemes CBR robusts davant de grans volums de dades complexes i incertes. D'altra banda, el Mapa autoorganitzatiu (SOM) és una tècnica no supervisada de clustering de la família de les xarxes neuronals que, gràcies a les seves capacitats de *Soft-Computing* i de *Knowledge Discovery*, pot extraure coneixement amagat a les dades per adaptar-se millor a elles. Aquestes virtuts han estat aprofitades per diversos autors per organitzar la memòria de casos del CBR amb la finalitat de millorar el seu rendiment. No obstant, hi ha molts aspectes que romanen oberts.

La finalitat de la tesi és definir i implementar un marc que integri les capacitats de *Soft-Computing* i de *Knowledge Discovery* de SOM a totes les fases del CBR. El resultat final d'aquest marc serà una plataforma anomenada SOMCBR, la qual serà avaluada mitjançant un ampli joc de dades provinents de l'*UCI Repository* i de les dades mèdiques i telemàtiques de dos dels projectes on s'emmarca la tesi: HRIMAC i ANALIA.

A més a més, la tesi aborda dos temes addicionals basats en l'optimització de funcions mitjançant una variant de la Computació evolutiva anomenada *Grammar Evolution* degut als requeri-

ments del projecte HRIMAC. Per cadascun d'aquests subobjectius s'implementa una plataforma, la qual és validada per *datasets* provinents de l'*UCI Repository* i el projecte HRIMAC.

Part I

Fonaments teòrics

Capítol 2

El Raonament basat en casos

El Raonament basat en casos és una branca de l'Aprenentatge analògic basada en resoldre problemes nous a partir de la identificació d'analogies amb d'altres prèviament resolts. Aquest capítol descriu a grans trets els seus orígens i fonaments per disposar d'una visió global del seu funcionament, aspecte vital pel seguiment d'aquesta tesi.

2.1 Fonaments i orígens del CBR

El Raonament Basat en Casos (*Case Base Reasoning* - CBR) (Kolodner, 1993) és una tècnica fonamentada en el principi bàsic de raonar a partir dels records. Donat un problema nou, cerca dins la seva experiència situacions semblants a través de les quals es puguin establir analogies per resoldre el nou problema.

La base del CBR en la Intel·ligència artificial es troba en els treballs realitzats per Roger Schank sobre memòries dinàmiques i les tasques de recordar patrons i situacions anteriors amb la finalitat de resoldre problemes i aprendre d'ells (Schank, 1982).

El primer sistema que pot considerar-se CBR fou desenvolupat per Janet Kolodner al 1983 (Kolodner, 1983) a la Universitat de Yale. Aquest sistema, anomenat CYRUS, era un sistema de preguntes i respostes sobre viatges i trobades basat en el model de memòria dinàmica de Schank i en la teoria del MOP (*Memory Organization Packet*) (Schank, 1982). Aquest model va posar les bases a partir de les quals es van realitzar molts desenvolupaments basats en aquesta idea, com per exemple MEDIATOR (Kolodner et al., 1985), PERSUADER (Sycara, 1988), CHEF (Hammond, 1989), JULIA (Hinrichs, 1992) o CASEY (Koton, 1989). D'altra banda des de la Universitat de Texas, Bruce Porter proposava un enfocament basat en l'aprenentatge de conceptes per realitzar tasques de classificació a partir d'aquests. El sistema desenvolupat s'anomenava PROTOS, i estava caracteritzat per integrar coneixements de domini general i específic en un mateix model de memòria estructurada. Seguint aquesta línia de recerca van desenvolupar-se molts altres sistemes, on els més importants van realitzar-se en l'àmbit legal sobretot gràcies als treballs d'Edwina Rissland i el seu grup de la Universitat de Massachusetts. Fruit del seu esforç va desenvolupar-se el sistema HYPO (Ashley, 1991), el qual era un sistema que interpretava els casos precedents i aconsellava a la gent dels jutjats. Aquest sistema va evolucionar en un nou sistema, anomenat CABARET (Rissland i Skalak, 1991), que feia servir un enfocament on els sistemes basats en casos es combinaven amb d'altres basats en regles. Dins també d'aquesta línia de dominis legals, altre sistema rellevant fou GREBE (Branting i Porter, 1991).

Tot i que en els inicis va ser la part americana la que tibava més fort, poc a poc des d'Europa van començar a sorgir contribucions importants centrades sobretot en el desenvolupament de sistemes experts i en l'adquisició del coneixement. Els primers treballs van ser realitzats per Klaus D. Althoff i altres membres de la Universitat de Kaiserslautern (Althoff, 1989), concretament en el

desenvolupament del sistema MOLTKE per a diagnòstics tècnics. Als 90 Enric Plaza i Ramon López De Mántaras van estudiar la seva aplicació en els diagnòstics mèdics (Plaza i López de Mantarás, 1990), i Beatriz López va desenvolupar el sistema BOLERO (López i Plaza, 1990) per estudiar mètodes basats en casos per raonament estratègic. D'altra banda a Aberdeen, el grup d'en Derek Sleeman va estudiar la utilització dels casos pel refinament del coneixement base amb la implementació del sistema REFINER, el qual fou desenvolupat per Sunil Sharma (Sharma i Sleeman, 1988). A la Universitat de Trondheim, Agnar Aamodt i els seus col·legues de Sintef van estudiar els aspectes de l'aprenentatge del CBR en el context de l'adquisició de coneixement general i el manteniment del coneixement particular amb el sistema GREEK (Aamodt, 1991). Des de la ciència cognitiva, els primers treballs sobre el raonament analògic els va realitzar Mark Keane a Dublin (Keane, 1988). Seguint aquest camí, el grup de Gerhard de la Universitat de Freiburg va centrar-se en l'estudi dels models cognitius en el projecte EVENTS (Strube i Janetzko, 1990).

Pel que fa respecte la zona d'Àsia, les activitats es centren a l'Índia (Venkatamaran et al., 1993) i al Japó (Kitano, 1993). En aquest últim hi ha un gran interès en buscar aproximacions d'arquitectures paral·les.

En qualsevol cas, durant tot aquest temps els sistemes CBR han anat evolucionat de manera diferent segons les necessitats del tipus d'aplicació on s'havia d'aplicar, i segons el tipus de coneixement amb el que treballaven. D'una banda, les aplicacions del CBR poden classificar-se principalment en els tipus següents:

Classificació. La finalitat és assignar una classe al nou problema a resoldre (Koton, 1989).

Configuració de sistemes. Consisteix a identificar i agrupar elements d'un entorn per aconseguir un funcionament concret (Bareiss, 1988).

Disseny. A partir d'un conjunt de requeriments s'han de definir els mètodes/tècniques que millor poden acomplir les necessitats plantejades (Navinchandra, 1991).

Tutories. L'objectiu és avaluar als estudiants mitjançant la presentació de casos apropiats (Farrel, 1987).

Planificació. Són situacions on cal produir una seqüència d'accions per tal d'assolir un objectiu (Hammond, 1989).

Interpretació basada en casos. A partir d'un conjunt de regles o premisses cal realitzar interpretacions que ajudin a trobar relacions (Ashley, 1991).

Adquisició del coneixement. L'objectiu és ajudar a extraure i classificar coneixement d'un domini per facilitar la comprensió de l'expert (Sharma i Sleeman, 1988).

D'altra banda, tradicionalment s'han distingit dos tipus de sistemes CBR segons la manera com aquests adquireixen el coneixement del problema:

Sistemes CBR amb ús intensiu de coneixement (*Knowledge Intensive*). Són sistemes que adquireixen coneixements del domini que no estan implícits a les pròpies dades. Per aquest motiu no necessiten que la base de dades del problema sigui gran, ja que fan un ús limitat dels processos d'aprenentatge de les dades. El coneixement d'alt nivell que gestionen requereix de formalismes sofisticats, com per exemple *frames* (Minsky, 1975) o lògiques de descripció (*description logics*) (Brachman i Schmolze, 1985). Són algorismes que s'engloben dins de la família dels Sistemes Basats en el Coneixement (*Knowledge Base Systems* - KBS).

Els fronts oberts actualment en aquest tipus de sistemes es poden resumir en (1) incorporar ontologies en el CBR així com l'ús de metodologies de modelatge de coneixement (*Knowledge*

Modeling) a l'hora de desenvolupar sistemes CBR, (2) desenvolupar tècniques de recuperació de casos amb ús intensiu de coneixement (*Knowledge-Intensive Retrieval*) i, (3) el desenvolupament de tècniques de reutilització i adaptació que siguin independents del domini.

Sistemes CBR amb ús restringit de coneixement (*Data Intensive*). També anomenats sistemes CBR amb ús intensiu de dades, són sistemes que extrauen exclusivament el coneixement de la base de dades del problema. L'avantatge d'aquest tipus de sistemes és que són capaços d'aprendre. Per contra, requereixen que la base de dades del problema sigui prou representativa per no tenir una visió parcial del problema. L'ús del coneixement es restringeix només a seleccionar un formalisme per representar els casos, el qual normalment és fa en forma de vector d'atributs i, en algun cas excepcional, mitjançant *frames*. Són sistemes que pertanyen al camp de l'aprenentatge artificial (*Machine Learning* - ML).

Els fronts oberts actualment són (1) la gestió de grans volums de dades i (2) la incorporació de tècniques *Soft-Computing* dins del procés de raonament.

Tot i aquesta classificació 'purista', és la combinació d'ambdós punts de vista el que permet la construcció de sistemes CBR més potents gràcies a la combinació dels seus punts forts: aprendre i integrar coneixement del domini. Aquest és precisament el marc conceptual que es proposa al projecte MID-CBR (TIN 2006-15140-C03), [1] on la classificació unidimensional anterior es converteix en dues dimensions fruit de la combinació del tipus d'aprenentatge i del coneixement utilitzat tal com mostra la figura 2.1.

Els dos sistemes anteriorment explicats estan ubicats als quadrants 1 i 4, els quals representen la configuració de tenir només una de les dues propietats. Per contra, el quadrant 2 fa referència a la tendència dels nous sistemes comentats abans, els quals combinen l'ús intensiu de coneixement i aprenentatge. Els fronts oberts de cadascuna de les famílies de tècniques romanen vigents, tot i que la suma d'ambdues obre noves qüestions que poden arribar a ser complexes de resoldre, com per exemple, resoldre satisfactòriament com combinar l'aprenentatge a partir d'exemples i el coneixement del domini. No obstant, els sistemes CBR ofereixen eines conceptuals per definir tècniques de recuperació, reutilització i manteniment per acabar afrontant amb èxit aquestes dificultats.

Finalment, el quadrant 3 representa sistemes CBR amb ús restringit de coneixement i d'aprenentatge. Aquests solen ser sistemes orientats a tasques de cerca com la planificació o la configuració basada en casos, ja que sovint fan un ús restringit del coneixement (defineixen només

[1] MID-CBR és un projecte coordinat pel Dr. Enric Plaza on participen l'IIIA, la UCM i el GRSI.

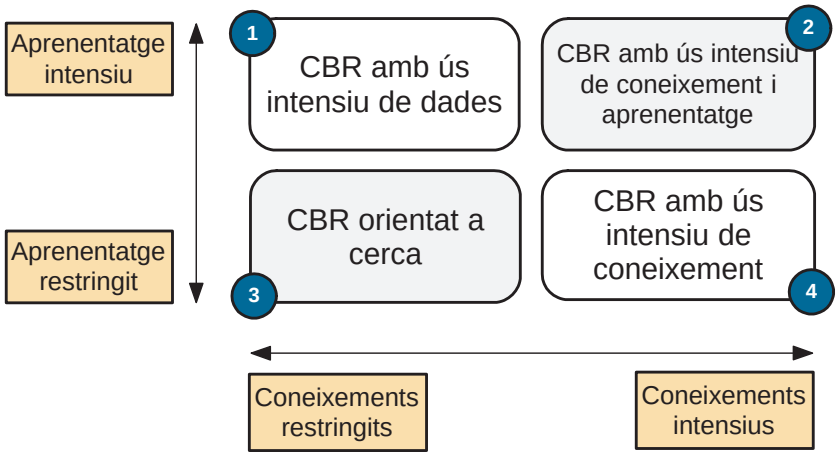

Figura 2.1: Els sistemes CBR actuals es poden classificar en 4 tipus segons el coneixement que facin servir, i el tipus d'aprenentatge que realitzin.

un formalisme de representació dels casos), i (2) poden treballar amb un nombre petit de casos (fins i tot sense base de casos (Veloso i Carbonell, 1993a)). Aquests aspectes poc coneguts van ser analitzats per Hanks & Weld (Hanks i Weld, 1992), els quals van demostrar que l'adaptació i generació de plans pot veure's com un procés de cerca. La gran utilitat d'aquests sistemes és que substitueixen els coneixements del domini per un procés de cerca adaptat a la tasca a resoldre, i l'aprenentatge es fa servir per millorar l'eficiència en el procés de resolució de problemes. La dificultat en aquest cas és què no es pot retenir una gran quantitat de casos. Els fronts oberts en aquest camp són la definició de tècniques de recuperació que facin servir formalismes més complexes, tècniques per reduir el volum de dades i, tècniques de reutilització capaces de reduir la combinatòria en el procés d'adaptació.

D'altra banda, també es proposa l'aparició d'una nova dimensió transversal a les dues anteriors fruït de l'àmbit d'aplicació dels sistemes que representen els casos de manera diferent a la concepció tradicional. Dos exemples d'això són els camps de la robòtica i de la gestió de documents. Tradicionalment, un cas és descriu com a una entitat discreta i representada en el seu propi formalisme de representació. No obstant, en entorns dinàmics com a la robòtica les entrades del sistema són valors canviants de l'entorn i no un cas fixe, i en entorns textuals els casos han de derivar-se de l'anàlisis de documents.

L'enfocament on està ubicat el GRSI (TIN 2006-15140-C03-03), i també la present tesi, és el del quadrant 1: sistemes CBR amb ús intensiu de dades. Per aquest motiu l'enfocament i la concepció del CBR al llarg de la tesi es farà sempre des d'aquest punt de vista, a excepció del capítol de la fase d'explicacions on ens desplaçarem lleugerament cap al quadrant 2. A continuació, s'introdueixen les fases del CBR sota aquesta percepció.

2.2 Algorisme del CBR

La descripció del cicle de vida del CBR es pot fer de moltes maneres, tot i que la més coneguda és la de la Fig. 2.2, la qual va ser proposada per en Agnar Aamodt i l'Enric Plaza (Aamodt i Plaza, 1994). Ells descriuen el CBR com un procés cíclic de quatre fases que giren al voltant de l'experiència del sistema, la qual s'emmagatzema en una estructura anomenada **memòria de casos**. Davant d'un nou problema, el sistema realitza els passos següents:

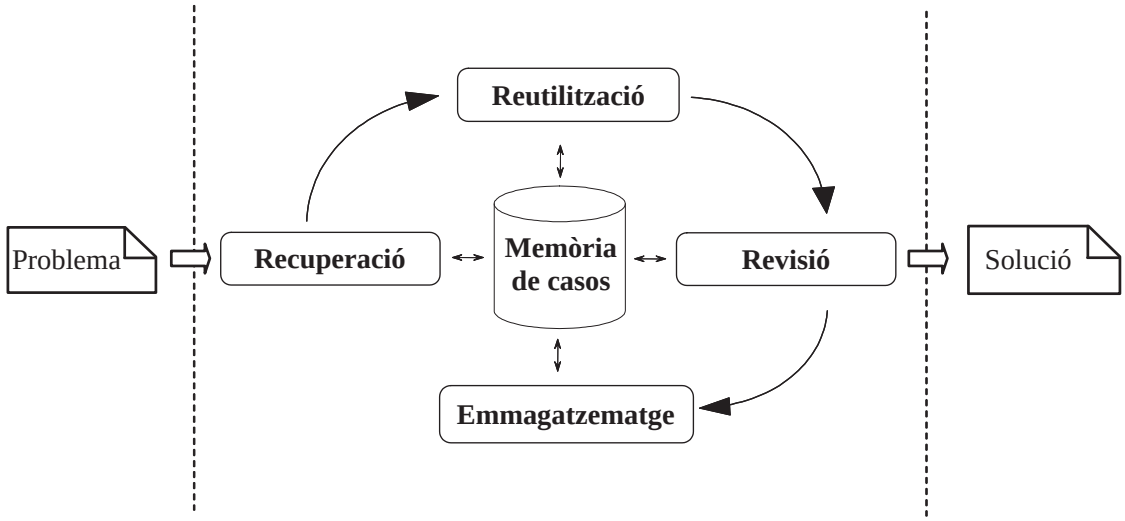

Figura 2.2: Cicle de vida del CBR proposat per en Agnar Aamodt i l'Enric Plaza (Aamodt i Plaza, 1994). Al llarg de la tesi ens referirem a la fase de reutilització en un sentit més ampli com a fase d'adaptació.

Fase de recuperació o *Retrieve.* Es busquen els casos més semblants de la memòria de casos respecte el problema nou. Dins d'aquest procés juguen un paper clau **l'organització de la memòria** i les **funcions de similitud**. El primer aspecte és important perquè condiciona la informació que es consulta, i el segon és important perquè defineix el grau de semblança entre el problema nou i els emmagatzemats a la memòria. L'èxit d'aquesta fase influeix en el desenllaç del raonament ja que la resta de fases depenen dels seus resultats.

Fase d'adaptació o *Reuse.* Es proposa una solució a partir dels casos recuperats a la fase anterior. L'adaptació pot ser des de molt simple (assignar la mateixa resolució que el/s cas/os recuperat/s) fins a molt complexa (aplicar estratègies de planificació per arribar a objectius comuns).

Fase de revisió o *Revise.* Es valida si la solució proposada és correcta a través de la figura de l'expert. No obstant, l'expert sovint no pot afirmar que la resolució sigui certa al 100%, només pot donar la seva opinió respecte si la resposta proposada té sentit i no és un disbarat. Això és degut a que si sapigués la resposta, no caldria un sistema CBR que li donés. En alguns casos el paper de l'expert pot automatitzar-se amb l'ajuda de regles o heurístiques específiques al domini.

Fase d'emmagatzematge o *Retain.* Es produeix el que pot qualificar-se com l'aprenentatge pròpiament dit. Cal definir quins criteris han d'acomplir-se per tal d'incorporar el nou coneixement. L'aplicació d'una política incorrecte pot desestabilitzar la memòria de casos, impossibilitant la resolució dels casos nous. Per tant, cal assignar una política segons el domini que s'està estudiant, la qual mantingui dins de les possibilitats una memòria reduïda, compacte i representativa.

De la mateixa manera que en qualsevol altre sistema d'aprenentatge, el CBR requereix d'una **fase prèvia per preparar i processar les dades**. A partir de la definició del problema a resoldre, cal obtenir i quantificar les característiques més rellevants que descriuen el problema. Molts cops aquestes valoracions no seran objectives ja que dependran de la percepció de l'expert, o de la fiabilitat de la mesura realitzada. A més a més, un cas està descrit per molts tipus de característiques i els seus valors tenen sovint soroll o valors desconeguts. Per tant, tots aquests aspectes s'han de tenir en compte a l'hora de treballar. L'apèndix E detalla les tècniques de processament emprades a la tesi.

Al llarg dels punts següents es dóna una visió global dels diferents components introduïts en aquesta secció. En cap cas la finalitat és oferir una visió exhaustiva de totes les variants existents a la literatura, només es vol donar un 'cop d'ocell' del que hi ha per ubicar al lector en el context de la tesi.

2.3 La memòria de casos

La memòria de casos és l'element al voltant del qual gira tot el cicle del raonament. Hi ha dos aspectes rellevants que cal tenir presents: (1) com es representa l'experiència i (2) com s'organitza. La definició d'aquests paràmetres determinarà en gran part la viabilitat de l'aplicació.

Un *cas* és descrit per Kolodner com '*una situació dins un context que representa una experiència que ens ofereix una lliçó fonamental per a aconseguir la meta del raonament*', és a dir, un cas és la representació del problema amb la seva respectiva solució mitjançant un conjunt d'*atributs*. Des del nostre punt de vista, el de l'ús intensiu del coneixement, un cas es representa normalment mitjançant un conjunt d'atributs. No obstant, poden haver-hi representacions basades en grafs o arbres si la definició del cas és complexa, o bé, no és constant.

D'altra banda, l'organització i l'accés als casos dins la memòria pot ser molt variada. Aquest aspecte influirà a totes les fases del cicle, però sobretot a les fases de recuperació i emmagatzematge. En el primer cas, perquè condiciona la manera de cercar els casos. En el segon cas, perquè condiciona la manera d'actualitzar el coneixement. Les configuracions més habituals són:

- Estructures que organitzen els casos de manera seqüencial en una llista (Ashley, 1991).
- Estructures en forma de graf que relacionen els casos en funció del valor dels atributs (Lenz et al., 1996; Nicholson et al., 2006).
- Estructures que relacionen els casos segons les seves similituds (Schaaf, 1995; Yang i Wu, 2001).
- Estructures que indexen els casos a partir de patrons (Vernet i Golobardes, 2003; Chang i Lai, 2005).
- Estructures jeràrquiques a través dels valors dels atributs (Schank, 1982; Wess et al., 1994).
- Estructures semàntiques on els casos s'agrupen en conceptes (Bareiss, 1988; Porter et al., 1990; Brown, 1994).

Per tant, la definició de la representació dels casos i l'organització de la memòria juguen un paper clau dins el rendiment del sistema, ja que condicionen l'accés a la informació buscada. A banda de disposar d'una organització i d'una definició correcte del problema, hi ha un altre aspecte que cal tenir molt present: la consistència i completesa de les dades. Per aquest motiu, sempre cal tractar prèviament les dades per 'netejar-les' mitjançant algunes de les tècniques que s'exposen a l'apartat següent.

2.4 Fase de preparació de dades

Resoldre un problema no és només aplicar un mètode sobre unes dades, cal seleccionar la configuració més adient tenint en compte la naturalesa de les dades a tractar. Per aquest motiu, preprocessar i corregir les dades són aspectes claus a tractar abans d'avaluar un mètode. Els principals aspectes a tenir en compte amb les dades a abans d'abordar un problema són els següents (Witten i Frank, 2005):

Selecció i/o ponderació d'atributs. Per evitar redundàncies, simplificar càlculs i millorar el rendiment, cal seleccionar només els atributs que són significatius i importants perquè la resta només aporten soroll. A més a més, cal determinar la seva contribució a la solució del problema.

Normalització d'atributs. Cadascun dels atributs numèrics disposa de rangs de valors diferents. Per aquest motiu, cal traslladar els seus valors als mateixos rangs per evitar esbiaixar els resultats. Respecte els valors nominals o basats en cadenes de caràcters, és recomanable aplicar processos d'estandarització per unificar conceptes semblants que estan representats de manera diferent (e.g. Dr. és el mateix que Doctor).

Compatibilitat d'atributs. Els casos poden estar composats per atributs de tipus diferents. Això fa necessari conversions dels tipus de les dades per poder aplicar determinades tècniques o anàlisis que només funcionen amb un tipus de dades.

Validesa de les dades. Per evitar mesures incorrectes o inconsistències en els resultats, és molt important detectar i eliminar el soroll de les dades, així com gestionar els valors desconeguts.

Creació d'atributs sintètics. Poden haver-hi situacions on sigui interessant crear nous atributs a partir dels existents, els quals siguin més útils o més entenedors. Per exemple, si tenim dues dates potser és més interessant tenir un atribut que indiqui la diferència d'anys enlloc de les dues dates, ja que pel sistema serà més útil i fàcil de gestionar aquest tipus d'informació. No existeixen regles que indiquin quan s'ha de realitzar això donat que depèn directament del domini i del tipus de problema.

Un cop tractades les dades segons la seva naturalesa i la manera com es vol abordar el problema, s'inicia l'aplicació de les fases del CBR descrites a l'apartat 2.3: recuperació, adaptació, revisió i emmagatzematge.

2.5 Fase de recuperació

La fase de recuperació és l'encarregada de cercar el conjunt de casos de la memòria que són més semblants al problema d'entrada. Per tal de realitzar aquest procés es requereix la definició d'una mètrica, anomenada funció de similitud, que avalua el grau de similitud entre el cas nou i els emmagatzemats. És important distingir entre els conceptes '**funció de recuperació**' i '**funció de similitud**'. La funció de recuperació defineix els criteris per recuperar casos de la memòria de casos (i.e., els que tinguin una similitud més gran d'un cert valor), i la funció de similitud mesura el valor de la similitud (i.e., 0.4). No obstant, alguns autors han desenvolupat aproximacions que, trencant l'estricta missió conceptual de la funció de similitud, mesuren també l'adaptabilitat del cas. Aquest enfocament juga un paper important en els sistemes de recomanació (Smyth i Mckenna, 1998; McSherry, 2003). Dins d'aquest l'àmbit, destaca una proposta on la diversitat dels casos recuperats s'integra com a element 'mesurador' (McSherry, 2002).

D'altra banda, els casos poden analitzar-se des d'un punt vista **local** (centrant la valoració en un parell d'atributs importants), o bé, **global** (valorant la semblança tenint en compte tota la informació dels casos) (Mougouie et al., 2003).

La manera com es realitza la cerca també és molt important per l'èxit d'aquesta fase. La cerca pot orientar-se principalment a **cerques sintàctiques** o a **semàntiques**, és a dir, a buscar casos que tenen la mateixa estructura d'informació, o bé, casos que tenen un significat similar. Aquests plantejaments estan estretament relacionats amb el quadrant de la figura 2.1 on s'ubica el sistema, ja que el tipus de cerca depen de la manera com són adquirides i representades les dades, així com del tipus de problema que es vol resoldre (de disseny, basats en analogies, de classificació, etc.). En el nostre cas, les cerques són sintàctiques ja que les funcions de similitud es centren en comparar de manera intensiva les dades a partir d'una estructura simple (vegeu l'apèndix F). L'extrem oposat el trobaríem a les cerques realitzades a partir de conceptes (Diaz i Calero, 2003), les quals fan un ús intensiu del coneixement. En aquest context, això suposa haver de definir mecanismes més complexes per poder tractar aquest coneixement de més alt nivell, com per exemple mitjançant la definició d'ontologies (Diaz i Calero, 2001).

En qualsevol cas, el procés de recuperació dels casos de la memòria es regeix per les etapes següents:

1. **Identificar característiques**. A partir del coneixement del domini del problema o d'un expert, cal saber quines característiques són les més rellevants. Aquest serà el coneixement que definirà el 'cas'.

2. **Cerca inicial.** Es busquen el conjunt de casos candidats plausibles, és a dir, s'estudia quins casos poden ser interessants a estudiar i comparar segons les seves característiques.

3. **Cerca.** A partir dels candidats anteriors, s'estudia més a fons cadascun dels casos tenint en compte totes les restriccions del problema.

4. **Selecció.** A partir de les cerques anteriors, es retornen els casos més similars. La selecció pot no tornar res si no hi ha un mínim de semblança.

Finalment, és important que la fase de recuperació permeti tant **especialitzar** com **generalitzar** (Bridge i Ferguson, 2002). S'entén per generalitzar la capacitat per resoldre el màxim nombre de casos a partir de casos generals de la memòria de casos. En canvi, especialitzar es refereix a la capacitat de resoldre els casos crítics a partir d'excepcions de casos emmagatzemats en la memòria.

Resumint, la manera de cercar a la memòria i la definició de la funció de similitud són aspectes molt importants per a portar a bon terme aquesta fase. Ambdós aspectes s'aborden amb profunditat a la segona i tercera part de la tesi.

2.6 Fase d'adaptació

La fase d'adaptació s'encarrega de proposar una solució al nou problema mitjançant la informació dels casos recuperats a la fase anterior. És molt important diferenciar les semblances entre el cas recuperat i el nou, i analitzar quina part de la solució anterior és reaprofitable per proposar la solució al nou cas. Aquesta tasca serà més o menys complexa segons el tipus de problema. De manera general, les tasques d'aquesta fase es poden separar en dos tipus (Wilke et al., 1998; Wilke i Bergmann, 1998):

Tasques analítiques. Es realitzen en problemes de classificació, diagnosi o suport a la presa de decisions. Es produeix quan la solució dels casos recuperats s'expressa mitjançant un element anomenat classe, i cal decidir quina de les diferents classes retornades és la correcte pel nou cas. L'estratègia més comuna per decidir-ho és a partir de realitzar tècniques de votació basades en els K veïns més semblants (*K-Nearest Neighbourn*, K-NN) (Cover i Hart, 1967; Dasarathy, 1991).

Tasques sintètiques. Es realitzen en problemes de configuració, disseny o planificació. En aquest cas, la solució dels casos recuperats és una traça que indica els passos realitzats per solventar el problema. D'aquesta manera, la tasca a realitzar consisteix en combinar parts de les solucions per obtenir-ne una de nova. Al mateix temps, dins d'aquest enfocament es poden diferenciar dues famílies de tècniques:

- **Adaptació transformacional.** Es centra en l'equivalència de la solució amb els casos recuperats. A partir de la solució d'un cas recuperat, la va transformant a partir de l'aplicació d'un conjunt d'operadors fins a obtenir una solució vàlida (Heinrich i Kolodner, 1991; Veloso i Carbonell, 1993a; Veloso i Carbonell, 1993b). També hi ha aproximacions que fan servir les estructures de les solucions de més d'un cas (Wilke et al., 1998; Wilke i Bergmann, 1998).
- **Adaptació generativa**. També coneguda com adaptació derivacional, es basa en augmentar la representació del cas amb coneixement detallat de les decisions preses a la resolució del cas (opcions, justificacions, etc.). Aquesta informació es fa servir per reinstanciar la traça del procés de la solució en el nou context per generar la nova solució. A l'entorn de la planificació destaca l'algorisme DerUCP (Tsz-Chiu et al., 2002), el qual serveix per adaptar plans usant generació derivacional (Carbonell, 1986).

Aquesta manera tan diferent d'abordar la problemàtica va portar als investigadors a definir el concepte de reutilització (*reuse*) (Aamodt i Plaza, 1994) per diferenciar la primera aproximació de la segona, on realment sí que hi ha una adaptació. Tot i aquesta divisió, hi ha enfocaments que

combinen els dos punts de vista. Aquest és el cas de la **Construcció Adaptativa** (*Constructive Adaptation*) (Plaza i Arcos, 2002), la qual és un mètode orientat a tasques de configuració on la solució és un conjunt de relacions entre elements. El mètode realitza una cerca aplicant una heurística *best-first* en un espai de solucions parcials fent servir la informació dels casos solucionats per guiar la cerca.

2.7 Fase de revisió

És la fase encarregada de revisar la solució proposada en la fase anterior i, en el cas que no sigui correcte, corregir la solució si és possible.

Habitualment cal disposar del suport d'un expert per decidir si el problema s'ha resolt correctament, o bé, disposar d'algun sistema o conjunt de regles específiques pel domini en qüestió que possibilitin la validació de la solució. No obstant, aquests mecanismes són sovint molt complexes de definir ja que si existissin no caldria definir aquest sistema CBR.

2.8 Fase d'emmagatzematge

És la fase responsable de l'aprenentatge del sistema mitjançant l'aplicació d'una política que estableix els criteris per mantenir el coneixement de la memòria a partir dels nous casos resolts. Les accions a realitzar es poden dividir de manera general (1) no emmagatzemar res, (2) emmagatzemar si el nou cas és diferent respecte els que hi ha, o bé, (3) emmagatzemar si s'ha classificat incorrectament (Golobardes, 1998).

Independentment de la política que s'apliqui, aquesta ha de garantir que la memòria de casos sigui compacte, representativa, reduïda i consistent. Aquestes propietats garantiran que es disposi de la experiència necessària per poder resoldre qualsevol cas, que la informació es trobi en un temps raonable i, a més a més, que aquesta sigui vàlida. Això va originar que sorgís fa uns anys una línia de recerca focalitzada en la consistència de les dades emmagatzemades a la memòria, la qual s'engloba sota el terme de Manteniment de la Memòria de Casos (*Case Base Maintance* - CBM) (Nieto, 2001). Els objectius a assolir per part d'aquestes tècniques es poden dividir en:

- **Agrupar o eliminar casos redundants.** A mesura que el sistema adquireix nou coneixement poden aparèixer problemes de dades replicades o redundants que provoquen una reducció del rendiment degut a que s'augmenta el temps d'exploració de la memòria de casos i, al mateix temps, el sistema pot confondre's si hi ha ambigüitats. Per combatre això, cal eliminar aquesta informació extra, o bé, crear un nou cas més general que englobi els que són similars (casos sintètics).

- **Identificar i eliminar inconsistències.** Amb el pas del temps el coneixement pot evolucionar fins al punt que es descobreixen noves relacions que arribin a invalidar coneixements previs. Cal vetllar per detectar aquestes inconsistències que introduiran soroll i incertesa.

La definició de les accions pertinents per assolir els objectius anteriors estan molt vinculades a la política d'emmagatzematge utilitzada. En qualsevol cas, el rendiment pot avaluar-se a partir de les tres premisses següents plantejades per Smyth i Mckenna (Smyth i Mckenna, 1998):

- **Eficiència** (*performance*). El temps mig de solucionar un problema.

- **Competència** (*competence*). El rang de problemes solucionats.

- **Qualitat de la solució.** (*quality*). El nivell d'error de les solucions i la seva fiabilitat.

Aquestes tres mesures estan molt vinculades entre elles, i totes tenen a veure amb la 'qualitat' de la memòria en termes de com representativa, compacta i reduïda sigui. Per exemple, les memòries petites tindran un rendiment alt en detriment de la competència i viceversa. Per aquest motiu, cal arribar a solucions de compromís que potenciïn els tres paràmetres.

D'altra banda, els principals tipus de problemes que poden aparèixer en el procés de manteniment de la memòria de casos poden classificar-se en:

- **Dos tipus de casos** (*Two types of cases*). Es produeix en entorns on els casos són no estructurats i, conseqüentment, poden produir-se situacions on la disposició dels atributs canvia. Cal definir com establir les comparacions entre aquests tipus de casos.

- **Inconsistència de casos** (*The inconsistent-case problem*). L'adquisició de coneixement nou pot invalidar altre que estava aprés anteriorment. Cal disposar de mecanismes per revisar el coneixement.

- **Casos redundants** (*The redundant-case problem*). La política d'aprenentatge pot fer que s'aprenguin casos molt similars als que ja es tenen. Aquesta redundància només confon al sistema i incrementa el temps d'exploració de la memòria. Per tant, s'han d'eliminar.

- **Utilitat** (*The utility problem*). Quan el cost associat a buscar un cas no compensa el fet de fer-ho servir, vol dir que el cas no és útil perquè està incrementant de manera innecessària l'espai de cerca. En aquesta situació, cal esborrar el cas.

Finalment, alguns autors com Zhu i Yang (Zhu i Yang, 1999) separen les tècniques de mantenint en dues grans famílies:

- **Manteniment d'índexs.** Actualitzar els índexs per tal de tenir sempre indexada la informació de manera correcte i tenir un rendiment òptim.

- **Manteniment del contingut.** Destaquen dues tècniques principalment:

  - Les tècniques d'**editing** consisteixen en modificar la memòria de casos inicial per d'aquesta manera començar amb una memòria de casos 'òptima'.
  - Les tècniques d'**oblit** consisteixen en esborrar casos de la memòria per millorar el rendiment del sistema. Aquestes tècniques permeten eliminar casos redundants, inconsistents, etc. Dues tècniques importants d'oblit són:
    * Selecció aleatòria (*Random Selection*): s'esborren casos aleatòriament.
    * Heurística LxF (*LxF Heuristic*): elimina els casos que menys informació útil tenen i que menys cops s'han fet servir.

2.9 Consideracions abans de resoldre un problema

L'aplicació del CBR varia segons el seu domini d'aplicació perquè cada domini té particularitats pròpies que condicionen la definició de les quatre fases introduïdes al llarg d'aquest capítol. De manera general, els aspectes que cal tenir presents són els següents:

Anàlisi de les dades que representen un cas. La naturalesa de les dades d'entrada i la manera com ens són proporcionades és vital per resoldre els problemes. Per això, cal preprocessarles per tractar el soroll, els valors desconeguts, determinar la influència dels atributs en el resultat i normalitzar les dades entre d'altres operacions.

Organització de la memòria. Tot el sistema gira entorn la memòria de casos perquè aquesta conté l'experiència en base a la qual es resolen els nous problemes. És necessari una organització que possibiliti un accés eficient al coneixement, és a dir, que ajudi a filtrar el coneixement que no està relacionat amb el problema a resoldre. Aquest aspecte es tracta àmpliament a la segona part de la tesi.

Definició de la mètrica de comparació. Cal utilitzar funcions que mesurin de manera fiable els casos, tenint en compte la tipologia i la representació de les dades. Aquesta necessitat s'aborda a la tercera part de la tesi.

Recuperació dels casos més similars. A partir de la mètrica establerta, cal cercar els casos més adients. Abans d'iniciar la cerca és important definir sota quines condicions es pot considerar que un cas sigui vàlid, com per exemple, el valor mínim de similitud per considerar dos casos com a similars.

Adaptació dels casos. La proposta de la solució ha de garantir un mínim de fiabilitat segons el domini del problema. Per exemple, en el cas del domini mèdic és millor no classificar que fer-ho malament.

Revisió de la nova solució. L'expert ha de disposar de tota la informació que el sistema hagi fet servir per proposar la nova solució i, d'aquesta manera, que li resulti més fàcil la comprensió de la decisió presa.

Manteniment del coneixement. La consistència i completesa del coneixement condiciona el rendiment. Cal establir polítiques d'emmagatzematge i validació que ajudin a fer que la memòria sigui el més representativa, compacta i reduïda possible.

Resum

El CBR és fonamenta en la capacitat d'establir analogies entre problemes per tal de resoldre'n de nous. El seu cicle de vida està composat per les fases de recuperació, adaptació, revisió i emmagatzematge, les quals giren al voltant de l'experiència emmagatzemada a la memòria de casos. La fase de recuperació s'encarrega de retornar els casos més similars respecte el qual es vol resoldre. La fase d'adaptació proposa una solució en base als casos recuperats. La fase de revisió vàlida la solució proposada. Finalment, la fase d'emmagatzematge és l'encarregada de mantenir el coneixement de la memòria a partir dels nous casos resolts. Com amb qualsevol altra sistema d'aprenentatge, prèviament a executar el CBR cal preprocessar les dades per tal d'eliminar inconsistències i tractar el soroll entre d'altres aspectes.

El capítol ha remarcat dos elements del cicle que són crítics per portar a bon terme el funcionament del CBR. D'una banda, la memòria de casos és l'element que conté l'experiència a partir de la qual el sistema resol els problemes. Aquesta estructura per gestionar el coneixement idealment hauria de ser compacte, reduïda i representativa per tal de garantir que la informació que es necessita per resoldre el problema està present a la memòria, pot accedir-se ràpidament i, a més a més, no hi ha redundàncies que puguin confondre al sistema. L'inconvenient és que aquesta situació idíl·lica no es presenta massa sovint degut a la complexitat i incertesa dels problemes reals. Per aquest motiu, han sorgit diferents aproximacions en els últims anys per tal de potenciar l'accés i l'organització de la memòria. Aquesta mateixa línia és la que s'aborda a la segona part de la tesi, on es planteja potenciar totes les fases del CBR a partir del coneixement descobert per un Mapa autoorganitzatiu, la qual és una tècnica de clustering no supervisada que gràcies a les seves capacitats de *Soft-Computing* i de *Knowledge Discovery* és capaç d'extraure coneixement de les relacions ocultes entre les dades.

D'altra banda, la funció de similitud és una mètrica de comparació que mesura el grau de semblança entre dos casos per tal d'avaluar quins casos de la memòria són els més rellevants per resoldre el problema nou. La definició d'aquesta funció no és trivial perquè sovint requereix d'un coneixement profund del domini d'aplicació, aspecte molt difícil d'aconseguir en els problemes reals. Això fa que molt sovint es facin servir funcions de similitud de propòsit general, les quals tot i que no proporcionen els resultats desitjables, permeten afrontar el problema. La tercera part de la tesi aborda la definició de funcions de similitud de manera específica a un problema per tal de millorar la precisió del CBR a l'hora de resoldre un problema nou. Aquesta definició o cerca de la funció es realitza a partir d'una variant de la Computació evolutiva, anomenada Evolució de gramàtiques, la qual fa servir restriccions per guiar la cerca.

Capítol 3

Els Mapes autoorganitzatius

La majoria dels problemes reals presenten gran volums de dades complexes que, a més a més, presenten imprecisions i coneixement aproximat. Conseqüentment, els sistemes CBR requereixen dues característiques per a oferir un bon rendiment. La primera consisteix en oferir un temps de resposta reduït sense que això repercuteixi substancialment en la qualitat dels resultats. D'altra banda, és recomanable que el sistema disposi dels mecanismes necessaris per poder tractar aquest tipus de coneixement i, d'aquesta manera, ser més robust i tolerant al soroll. Aquest capítol presenta els Mapes autoorganitzatius com una estratègia de clustering no supervisada emmarcada dins de la família de tècniques *Soft-Computing*, gràcies a la qual és possible identificar agrupacions de dades representades per un patró. Aquesta propietat permet organitzar la memòria de casos per tal d'abordar els dos problemes anteriors.

3.1 Fonaments i orígens del SOM

Les xarxes neuronals (Bishop, 1995) han emergit en els últims anys com eines molt potents pel modelat estadístic aplicat al reconeixement de patrons, tant per tasques de classificació com de predicció. Aquestes disposen d'un conjunt de característiques que les fan molt atractives per ser utilitzades: (1) poden processar dades amb soroll o incompletes, (2) tenen una alta tolerància als errors si hi ha neurones espatllades (en el cas d'implementacions físiques), i (3) poden respondre en un temps molt breu gràcies al seu paral·lelisme implícit. Tot i que hi ha moltes variants dins aquesta família d'algorismes, destaquen per sobre de tot els Mapes autoorganitzatius o Mapes de Kohonen (*Self-Organizing Map* - SOM) (Kohonen, 1984; Kohonen, 2000), els quals s'han fet servir en centenars d'articles científics i aplicacions reals (Kaski et al., 1998b; Oja et al., 2003).

Al 1982 T. Kohonen va presentar SOM com un model de xarxa basat en determinades evidències cerebrals (Kohonen, 1982). Diversos estudis van trobar que en el còrtex dels animals superiors apareixen zones on les neurones detectores de trets característics es troben topològicament ordenades, de tal manera que les informacions captades de l'entorn a través dels òrgans sensorials es representen internament en forma de mapes bidimensionals. Per exemple, les neurones de l'àrea somatosensorial que reben senyals de sensors pròxims a la pell es troben distribuïdes de manera que representen com una mena de mapa de la superfície de la pell. Encara que l'organització neuronal està predeterminada genèticament, l'aprenentatge produït amb la nostra experiència influeix en la seva distribució final. Per tant, això suggereix que el cervell disposa de la capacitat inherent de crear mapes topològics de les informacions rebudes de l'exterior. D'altra banda, també s'ha observat que la influència de les neurones respecte les que té al seu voltant varia segons la distància entre elles, essent aquesta molt petita si les neurones estan allunyades. Tenint en compte aquest conjunt d'evidències, el model de xarxa presentat per Kohonen pretén representar de forma simplificada

la capacitat del cervell per definir mapes topològics a partir de les senyals rebudes de l'exterior (Kohonen, 1990).

En els dos apartats següents es descriu el funcionament de SOM, així com els aspectes més rellevants que s'han de tenir en compte per la seva correcta aplicació.

3.2 Algorisme de SOM

SOM es defineix com una tècnica no supervisada de clustering que projecta l'espai original de les dades en un altre més reduït. Al ser una tècnica no supervisada de Knowledge Discovery, ha de descobrir per ella mateixa els trets comuns, correlacions, i categories de les dades. En altres paraules, ha d'agrupar les dades semblants sota un mateix patró que ressalta les relacions ocultes entre les dades. Això permet facilitar la comprensió i visualització de les dades, sobretot quan aquestes presenten un alt número de dimensions. A més a més, les seves capacitats *Soft-Computing* li permeten tractar dades complexes, incertes i també amb soroll.

L'arquitectura de la xarxa s'organitza en dues capes tal com mostra la figura 3.1. La capa d'entrada s'encarrega de rebre i transmetre a la capa de sortida la informació procedent de l'exterior. Per aquest motiu la capa està representada per N neurones, una per cadascuna de les possibles variables d'entrada. D'altra banda, la capa de sortida s'encarrega de processar la informació i definir les agrupacions de les dades en el mapa. Tot i que normalment el mapa és de dues dimensions de $M \times M$ (com en el cas de la figura 3.1), també pot representar-se en una dimensió com una cadena de neurones, o bé, en tres dimensions amb un paral·lelepípede.

En qualsevol cas, el sentit de les connexions sempre és des de la capa d'entrada cap a la de sortida. Cada neurona d'entrada està connectada amb totes i cadascuna de les neurones de la capa de sortida mitjançant un pes tal com mostra la figura 3.1 (w_{im}: la neurona d'entrada i i la neurona de sortida m). Això permet construir un vector de N components anomenat vector director o de referència a través del qual es modela la categoria que representa la neurona m. D'altra banda, les neurones de la capa de sortida estan relacionades entre elles mitjançant unes connexions laterals que poden ser d'excitació o d'inhibició. Aquestes connexions defineixen el veïnatge de la neurona, aspecte que determina el rang d'influència entre les neurones. A la figura 3.2 es representa un exemple de la topologia hexagonal i una altra de la rectangular, on hi ha 6 i 8 veïns respectivament. Com pot observar-se, la tipologia d'interconnexió es manté regular al llarg de tot el mapa.

Tot seguit passarem a exposar com es realitza la construcció dels models. Primer de tot cal tenir en compte que no existeix un algorisme d'entrenament que sigui totalment estàndard, tot i que sí es poden identificar un conjunt de passos comuns. La idea principal és identificar les categories de les dades (una per neurona) a partir de la presentació de diferents exemples. De manera general el procés d'aprenentatge es desenvolupa en les etapes següents:

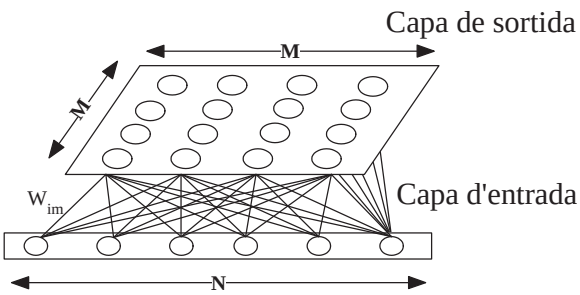

Figura 3.1: Arquitectura d'un mapa 2D.

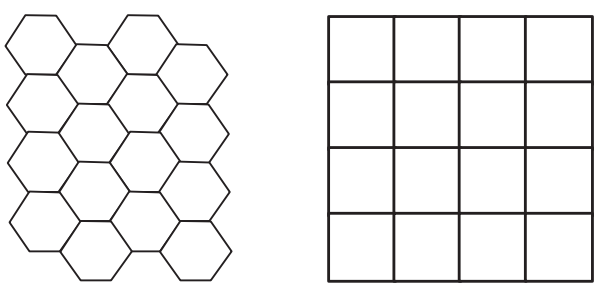

Figura 3.2: Topologies hexagonal i rectangular.

1. Els vectors directors són inicialitzats aleatòriament.

2. Es compara l'exemple d'entrada respecte els vectors directors de cadascuna de les neurones per determinar quina és la neurona que el representa millor. La guanyadora serà la que tingui els pesos més semblants.

3. El vector de la neurona guanyadora s'actualitza per tal de representar millor l'exemple d'entrada. D'aquesta manera, la neurona guanyadora respondrà amb més força quan es presenti un nou exemple semblant a ella.

   Al mateix temps, també s'actualitzen les neurones veïnes però amb menys intensitat. A diferència d'altres tècniques de clustering com per exemple el K-means (Hartigan i Wong, 1979; Kanungo et al., 2000), el nombre de neurones a modificar es calcula mitjançant una funció de veïnatge que canvia amb el pas del temps. Gràcies a això s'aconsegueix una ordenació topogràfica, en la qual la relació de proximitat dels vectors a l'espai original N es manté al ser projectats a l'espai reduït $M \times M$.

4. Els passos 2 i 3 es repeteixen per tots els exemples d'entrenament fins que l'error dels vectors directors per representar els exemples arriba a un cert valor llindar, o bé, transcorren un nombre màxim d'iteracions.

Pot observar-se com l'efecte global no és rés més que apropar de manera iterativa cadascun dels vectors de pesos de les neurones als diferents patrons de les dades, tenint en compte que hi haurà com a màxim tants patrons com neurones.

L'algorisme 3.1 descriu l'esquema específic d'entrenament que s'ha fet servir al llarg de la tesi per construir els mapes en dues dimensions, ja que no hi ha un esquema general estàndard. Abans de res, cal inicialitzar els vectors de pesos. Hi ha diferents criteris entre els quals destaquen

Algorisme 3.1: Definició de l'entrenament seguit a la tesis per construir un mapa 2D

Sigui MAX el nombre màxim d'iteracions a realitzar
Sigui E_{MAX} l'error màxim acceptat com a 0.01
Sigui M_m un node m d'un mapa amb $M \times M$ neurones amb topologia rectangular, i $\vec{v_m}$ el seu vector director
Sigui I una instància del conjunt d'entrenament, i $\vec{I}$ el vector que la representa
Sigui t la iteració actual
Sigui $\alpha(t)$ el factor d'aprenentatge a la iteració t, que té com a valor mínim α_{MIN}=0.01
Sigui $\beta(t)$ el factor de veïnatge a la iteració t, que té com a valor mínim β_{MIN}=1
Tots els vectors directors $\vec{v_m}$ del mapa s'inicialitzen aleatòriament entre [0..1]
t=0
$\alpha(0)$=0.6
$\beta(0)$=M
$error$=$+\infty$
Mentre $((t < MAX)\&(error > E_{MAX})$ **fer**
    $error$=0
    Per tot I *del conjunt d'entrenament* **fer**
        Sigui M_I el node al que millor s'adapta I segons l'equació 3.1
        Tots els nodes veïns de M_I amb un radi $\beta(t)$ s'ajusten mitjançant l'equació 3.2
        $error$=$error$+ $\|\overline{I} - \overline{M_I}\|$
    error=error/$(M \times M \times Instances)$
    Actualitzar $\alpha(t)$ i $\beta(t)$ amb les equacions 3.3 i 3.4 respectivament, si no han assolit els seus valors mínims
    t++

inicialitzar entre [0..1], entre [-1..1], o totes les components dels vectors a 0. En el nostre cas sempre s'inicialitza entre [0..1] perquè sempre es treballa amb dades normalitzades en aquest rang. A partir d'això comença l'entrenament de les neurones fins que transcorren MAX iteracions, o bé, s'aconsegueix que els vectors modelin les dades per sota d'un cert error. Tot i que no hi ha un criteri estàndard per fixar el nombre d'iteracions, aquest hauria de ser proporcional a la mida del mapa. Una dada de referència són 500 iteracions per neurona, tot i que amb 150 iteracions per neurona es resolen la majoria dels problemes (Kohonen, 1990). En aquest procés hi ha dues etapes: primer s'organitzen els vectors de pesos en el mapa, i després s'ajusten els elements d'una manera més precisa en el mapa. Aquesta adaptació s'aconsegueix amb la definició dels factors d'aprenentatge (mesura el grau d'influència a l'ubicar un nou exemple en el mapa, [0..1]) i el factor de veïnatge (mesura el radi de veïns als quals el nou element influeix, $[1..M]$). Inicialment es requereix considerar el major nombre possible de nodes, així com influir de la manera més agressiva sobre els vectors dels nodes del mapa. Per aquest motiu, els dos factors han de disposar de valors alts. En canvi, a mesura que l'entrenament avança es fa necessari estabilitzar el sistema. Conseqüentment, la influència dels dos factors es redueix seguint les equacions 3.3 i 3.4 fins a assolir un factor d'aprenentatge al voltant del 0.01, i un factor de veïnatge amb valor 1, el qual fa referència només als veïns immediats del voltant. Encara que l'algorisme descrit fa servir la distància euclidiana com a mètrica de comparació, pot fer-se servir qualsevol altre [1].

$$\forall m : 1\ m \leq M : \|\vec{I} - \vec{v_I}\| \leq \|\vec{I} - \vec{v_m}\| \tag{3.1}$$

$$\vec{v_m}(t+1) = \vec{v_m}(t) + \alpha(t) \cdot (\vec{I} - \vec{v_m}(t)) \tag{3.2}$$

$$\beta(t+1) = \beta(t) + (\beta_{MIN} - \beta(t)) \cdot \frac{t}{MAX} \tag{3.3}$$

$$\alpha(t+1) = \alpha(t) + (\alpha_{MIN} - \alpha(t)) \cdot \frac{t}{MAX} \tag{3.4}$$

on:

$\|\vec{A} - \vec{B}\|$ és el sumatori de les diferències al quadrat.

3.3 Consideracions abans de resoldre un problema

Els principals aspectes a tenir presents a l'hora de definir com ha de realitzar-se l'entrenament de SOM poden dividir-se en els punts següents:

Mida del mapa. La mida del mapa condiciona el nombre patrons que es poden modelar. Mides de mapes grans crearan molts models massa específics amb pocs elements, dificultant la tasca de generalitzar comportaments. En canvi, mides de mapa petites generaran pocs models massa genèrics amb moltes instàncies, fent que el clúster no representi cap comportament. Per tant, cal arribar a una mida de compromís entre les dades de les quals es disposa, i el nombre de patrons a definir. En el nostre cas, la decisió del mapa es realitza mitjançant una estratègia semblant a la del X-means (Pelleg i Moore, 2000). Aquesta consisteix a generar diferents configuracions de mides, i seleccionar aquella que té l'error més petit. És important tenir present que no necessàriament tots els models han de contenir elements. En aquest cas, la neurona és com si no estigués i es pot entendre com una discontinuïtat espaial.

Factor de veïnatge dels clústers. El veïnatge condiciona la construcció dels patrons perquè determina el radi d'influència d'un model respecte a la resta. Inicialment ha d'afectar a la

[1] En el cas de fer servir SOM amb un altre sistema, els dos han de fer servir la mateixa mètrica per evitar divergències en els criteris de similitud.

gran majoria dels nodes, i de manera progressiva ha de reduir-se fins a influenciar només als del voltant.

Factor d'aprenentatge dels clústers. El factor marca el ritme de la convergència de l'algorisme. Valors alts dificulten que l'algorisme evolucioni ja que poden fer que el procés d'aprenentatge sigui aleatori, i valors molt petits fan que trigui molt a evolucionar. Cal arribar a un valor de compromís, el qual estarà marcat per la tipologia de les dades.

Mètrica de comparació. És l'element encarregat de mesurar en quin grau un element s'assembla a un patró. La seva definició és complexa i està directament lligada a les peculiaritats del domini. Quan es fa servir SOM i CBR, les dues han de fer servir les mateixes mètriques.

Resum

Aquest capítol ha presentat SOM com una tècnica no supervisada de clustering basada en projectar l'espai original de les dades a un altre més reduït, on les característiques més rellevants queden resaltades. A grans trets, el procés de clustering pot resumir-se en els següents passos: (1) definir un nombre el nombre de clústers a trobar; (2) inicialitzar aleatòriament els patrons que modelen cada clúster; (3) per cada exemple d'entrada buscar el clúster que millor el representa; (4) actualitzar el patró del clúster seleccionat, així com el dels seus clústers veïns per tal de mantenir les distancies topològiques entre l'espai original i el reduït; (5) repetir els passos 3 i 4 fins de manera iterativa fins que l'error del mapa sigui inferior a un cert llindar, o bé, s'arriba al nombre màxim d'iteracions. A més a més, en tot aquest procés els veïns implicats i el grau d'actualització dels patrons va canviant de més a menys fins a assolir una situació estable.

D'altra banda, les característiques més importants de SOM són: (1) preserva la topologia original de les dades; (2) no té cap problema si l'espai original té un nombre elevat de dimensions; (3) incorpora la selecció de característiques de manera implícita; (4) encara que una classe tingui pocs exemples aquests no es perden; (5) permet visualitzar les dades d'una manera fàcil; i (6) s'autoajusta de manera autònoma per ajustar-se bé a les dades. D'altra banda, els inconvenients del mètode són que el resultat final dels clústers està condicionat per l'ordre en el que s'agafen els elements, i no és trivial definir els paràmetres de la seva configuració.

Com es veurà a la segona part de la tesi, SOM permetrà millorar el rendiment del CBR en termes de temps de resposta i de capacitat resolutiva, ja que nodrirà totes les fases del CBR amb el coneixement descobert de les relacions ocultes entre les dades.

Capítol 4

La Programació genètica i l'Evolució de gramàtiques

El domini dels problemes reals està composat normalment per coneixement imprecís, incert, parcialment vertader i aproximat. Això fa difícil establir com mesurar la similitud entre dos elements. Això fa que sovint sigui necessari l'aplicació de tècniques basades en *Soft Computing* per gestionar aquest tipus de coneixement. Aquest capítol introdueix la Programació genètica i l'Evolució de gramàtiques com dues estratègies basades en la Computació evolutiva, gràcies a les quals es poden optimitzar/ajustar funcions/programes de manera específica per un domini. Aquests conceptes seran posteriorment utilitzats pel disseny de funcions de similitud pel CBR a la tercera part de la tesi.

4.1 Fonaments i orígens de la GP i la GE

La Computació evolutiva és un paradigma que engloba els algorismes d'aprenentatge que imiten el procés natural de l'evolució de les especies per tal d'aprendre. Els dos principis fonamentals de l'evolució natural són:

- **La selecció natural**. Fou introduït per Darwin al 1830, i intenta explicar com la vida evoluciona a partir de petits canvis i de la selecció dels millors individus.
- **El concepte d'herència**. Fou introduït per Mendel al 1865, i explica com els fills són el resultat de la combinació de les característiques dels seus pares, formant un individu que hipotèticament hauria de ser millor.

Aquests dos principis es reprodueixen en la Computació evolutiva de la manera següent:

- **La selecció natural** es realitza a partir d'una funció (funció de *fitness*) que mesura quins individus són els millors de la població. Els individus seleccionats són els que en principi tenen més probabilitats de sobreviure.
- **L'herència** es realitza creuant el material genètic dels pares a partir de la seva representació interna generant així un nou individu, en el qual poden introduir-se petits canvis aleatoris.

L'evolució natural ha trobat espècies (solucions) d'una gran complexitat genètica que resolen el problema de la supervivència en el nostre món. Conceptualment, i tenint com a prova les espècies actuals, es considera que el paradigma de l'evolució natural és molt potent. Al conceptualitzar l'evolució natural a la Computació evolutiva implícitament s'està fent una cerca directa de la

solució, de manera que en entorns on els espais de solucions possibles són molt grans i/o complexos resulta més eficient que la cerca exhaustiva o aleatòria. La cerca es considera directa perquè la població es guia cap a la solució desitjada a través de la funció de *fitness*.

Les variants d'algorismes dins la Computació evolutiva es diferencien principalment per la representació dels individus de la població, per la definició i aplicació dels operadors genètics, i per la finalitat que es persegueix. De totes les vessants, les més representatives són les següents:

Algorismes Genètics (*Genetic Algorithm* - GA). És l'enfocament americà, i va ser proposat per John Holland i el seu equip de la universitat de Michigan al 1970. La seva finalitat és l'optimització de funcions.

Estratègies Evolutives (*Evolutive Estrategies* - EE). És l'enfocament europeu dels GA i té com a objectiu la parametrització de sistemes.

Programes d'Evolució (*Program Evolution* - PE). Els Programes d'Evolució són un refinament dels GA proposat per Michalewicz en 1994, que tenen el mateix objectiu que els GA però s'apliquen sobre problemes on cal una representació més complexa.

Programació Evolutiva (*Evolutive Programming* - EP). La Programació Evolutiva està fonamentada en els GA's, i fou proposada per Lawrence J. Fogel al 1960. El seu objectiu es centra en l'optimització de funcions combinatòries de valors reals on la superfície d'optimització és abrupta i, per tant, presenta solucions òptimes locals.

Els paradigmes de la Programació Genètica (GP) i l'Evolució de Gramàtiques (GE) que estudiarem en aquest capítol es consideren com un subgrup dels GA. Aquests dos paradigmes tracten d'aconseguir un dels principals reptes de la Informàtica: que els ordinadors siguin capaços de resoldre problemes sense haver estat programats explícitament per fer-ho. És a dir, aconseguir que un ordinador faci el que ha de fer sense dir-li exactament com.

La GP va començar a ser introduïda al 1987 per Koza i el seu grup (Koza, 1992). L'objectiu de la seva proposta era fer evolucionar individus que representessin programes que resolien un cert problema. Aquesta nou enfocament va implicar una revisió tant de la representació com dels operadors dels GA, ja que la representació tradicional basada en cadenes de bits era insuficient per representar les relacions que poden aparèixer entre els blocs d'un programa. Concretament, els arbres van ser la representació escollida per representar els programes: els nodes representen els elements del programa (variables, constants, operacions, etc.), i les arestes les relacions entre els blocs. A més a més, amb aquesta representació era directe obtenir el programa.

Malgrat les bones intencions d'aquesta tècnica, un dels seus punts febles és que per problemes amb una certa complexitat l'espai de cerca esdevé massa gran, fent que la cerca del programa esdevingui *NP-Hard*. Per compensar aquesta 'excessiva flexibilitat' que sovint feia inviable el problema, van anar apareixent propostes basades en introduir restriccions per acotar la cerca. En el cas de la GP proposada per Koza, aquesta tasca no era gens trivial ja que requeria una adaptació dels operadors que, a més a més de ser costosa, no sempre era viable com més endavant s'exposarà.

Als anys 90, Ryan and O'neill (Ryan et al., 1998) van proposar la GE com un enfocament revolucionari que a diferència de la resta, es centrava en treballar directament sobre els GA i no sobre la GP per incorporar restriccions. La idea principal d'aquesta tècnica és aprofitar els avantatges de la representació i el cicle clàssic dels GA, però a l'hora d'aplicar l'avaluació dels individus transformar la cadena de bits en un programa. Aquesta transformació es basa en un procés de mapeig genotip-fenotip, on la cadena de bits es mapeja sobre una gramàtica en *Backus Naur Form* (BNF) que modela les propietats del programa a construir. Per tant, el gran avantatge d'aquest enfocament és que separa l'espai de cerca del de solucions, fent més fàcil la definició i modelatge de les restriccions.

La GP i la GE poden aplicar-se a molts tipus de problemes, tot i que les seves principals aplicacions són:

Optimització de funcions. A partir d'un conjunt de paràmetres amb els seus rangs respectius, ajusta els valors més acurats per aconseguir modelar un cert comportament.

Cerca de funcions/programes. Troba el programa/funció que satisfà un certs requeriments.

4.2 Algorisme de la GP i la GE

La GP i la GE són paradigmes que tenen com a objectiu evolucionar un conjunt d'individus que representen programes seguint les lleis de l'evolució natural i de l'herència. L'aplicació d'aquests principis sobre els individus produirà una nova i generalment millor població de programes. Ambdues estratègies es basen en el cicle de vida dels GA, el qual es defineix com un seguit de fases que simulen els efectes de l'evolució natural de les espècies tal com indica la figura 4.1.

De la mateixa manera que amb el CBR, cal aplicar un pas previ per preprocessar les dades per tal de preparar-les, així com per garantir la seva consistència. Les fases que composen l'esquema general del cicle dels GA són les següents:

Inicialització. Crea la primera generació d'individus, els quals seran els progenitors a partir dels quals es generaran la resta d'individus de les successives generacions. És important que aquesta fase garanteixi una certa diversitat en la població per tal de disposar de la varietat genètica necessària per generar l'individu que resol el problema.

Avaluació. Cada individu disposa d'un grau d'adaptació al medi que es mesura mitjançant una puntuació representada per un valor de *fitness*. Aquesta fase avalua aquest valor per mesurar el seu grau d'adaptació: a millor grau, major capacitat per resoldre el problema.

Selecció. La nova generació es construeix a partir del material genètic dels individus de la generació actual. Aquesta fase determina quins individus són els més indicats per traspassar el seu material genètic. De manera general, classifica els individus en quatre tipus segons el seu *fitness*: els que sobreviuen i els que no, i els que es reprodueixen i els que no.

Aplicació d'operadors primaris. És la fase on es generen els individus de la pròxima generació com a resultat de la recombinació de dos individus de la població actual (creuament), o directament d'un individu aïllat (reproducció).

Aplicació d'operadors secundaris. Aquesta fase és l'encarregada de produir petits canvis aleatoris que ajudin a introduir millores.

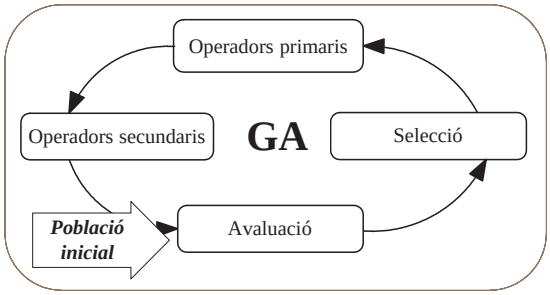

Figura 4.1: Esquema general de les etapes del cicle de vida dels GA.

Algorisme 4.1: Algorisme del cicle de vida d'una població en un GA.

Funció *GA()* **és**
  Siguin P, P' poblacions d'individus
  Sigui MAX_P la mida màxima de la població
  Sigui MAX_GEN el nombre màxim de generacions a evolucionar
  Inicialització de la població
  Avaluació de tots els individus de la població
  generacioActual=0
  Repetir
    //Construcció de la població de la nova generació;
    Sigui S el conjunt d'individus **seleccionat** de la població P
    $P'=\emptyset$
    //A continuació s'apliquen els operadors sobre els individus de S
    Mentre *(mida(P')<MAX_P)* **fer**
      Sigui *pOp* un **operador primari** seleccionat aleatòriament
      Aplicar l'operador *pOp* en el/s individu/s de S seleccionat/s
      Sigui *sOp* un **operador secundari** seleccionat aleatòriament
      Aplicar l'operador *sOp* sobre els nous individus anteriors
      Aplicar la política de reemplaçament entre els pares i fills
      Guardar els individus que passen en P'
    Avaluació dels individus de la població P' que han sofert modificacions
    Aplicació de millores sobre P' (per exemple elitisme)
    P=P'
    generacioActual++
  Mentre *((generacioActual< MAX_GEN) & (trobadaSolucio()==fals))*
  retorna *Individu amb el millor fitness*

Per tant, a partir d'una **població inicial** aleatòria d'individus s'**avalua** per cadascun d'ells el seu grau d'adaptació. Aquesta mesura s'anomena *fitness*, i determinarà en gran part quins individus sobreviuen, moren, o es reprodueixen segons l'estratègia que s'estigui aplicant. De tota la població es **seleccionen** un conjunt d'individus de la població a partir dels quals es creen nous individus com a fruit de la recombinació genètica mitjançant els operadors primaris (**creuament i reproducció**). Posteriorment, s'introdueixen petits canvis aleatoris (**mutació**) sobre els nous individus amb la finalitat de provocar millores espontànies. Tot aquest procés es repeteix fins que s'obté un individu que té un *fitness* màxim, o bé, quan han transcorregut un nombre màxim de generacions tal com mostra l'algorisme 4.1. Al llarg dels punts següents aquestes fases s'aborden des del punt de vista de la GP i de la GE, tenint en compte les peculiaritats de cada representació.

4.3 La representació dels individus

4.3.1 Representació de gens basada en arbres en la GP

La GP va ser concebuda com un plantejament revolucionari que pretenia crear programes de manera automàtica sense dir com fer-los partint dels conceptes de la CE. Aquest nou punt de vista va implicar una redifinició de la representació dels individus per tal de representar els programes, ja que la representació lineal era insuficient: calia una estructura que permetés relacionar amb una certa jerarquia un nombre indeterminat d'elements. Concretament, l'estructura en forma d'arbre n-ari va ser l'escollida: els nodes eren els potencials elements, i les branques definien les seves jerarquia. A més a més, en el cas dels nodes es podien distingir dos tipus:

- **Nodes funció**: són els nodes intermitjos, i representen les operacions que poden realitzar-se.

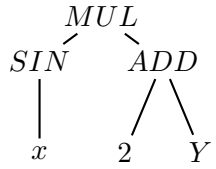

Figura 4.2: Exemple d'individu en la GP representat mitjançant un arbre n-ari.

```
(*(+ 2 x) (sin x))
```

Figura 4.3: Representació d'un individu en la GP usant Lisp.

Exemples de nodes: $+$, $*$, *sinus*.

- **Nodes terminal**: són els nodes fulles, i representen les constants o variables del programa sobre les quals s'apliquen les operacions dels nodes funció. Exemples de terminals: 2, x.

La figura 4.2 mostra un exemple de la representació d'un programa, en aquest cas una funció, representat en forma d'arbre. Koza va trobar en la programació funcional, concretament en Lisp (Steele, 1990), una manera molt fàcil de representar els individus. La figura 4.3 representa el mateix exemple en Lisp. Tot i que aquesta representació té com avantatge que els individus poden executar-se directament, té l'inconvenient de necessitar una adaptació especial dels operadors primaris i secundaris per tal de respectar l'estructura de l'arbre. Si a més a més es volen gestionar restriccions, el procés d'adaptació es torna encara més complex.

4.3.2 Representació de gens basada en estructures lineals en la GE

La GE persegueix la mateixa finalitat que la GP: evolucionar individus que representen programes fins aconseguir la definició d'un que resolgui el problema. No obstant, la GE afronta el problema fent servir una estratègia totalment diferent: separar l'espai de cerca del de solucions.

L'espai de cerca s'explora fent servir la mateixa representació que la dels GA, és a dir, es basa en una representació lineal de bits que s'agrupen cada X elements per definir *codons* (enters). Al fer servir la mateixa representació que GA, no cal realitzar cap tipus d'adaptació dels operadors i totes les millores aplicables als GA ho són també per la GE. És en la fase d'avaluació on els enters que formen l'individu es mapegen sobre una gramàtica BNF, amb la finalitat de construir un programa a partir d'aquests enters. Aquest mapeig genotip-fenotip es detallarà més endavant. Només cal tenir present que el nombre de codons de l'individu ha de ser suficientment gran com per permetre representar el programa.

4.4 Inicialització de la població

La fase d'inicialització és l'encarregada de crear la primera generació d'individus a partir dels quals es cerca la solució del problema.

Tant en la GP com en la GE existeixen dues famílies d'inicialitzacions: **aleatòries** i **no aleatòries**. Amb les aleatòries es pretén crear individus sense cap restricció, i amb les no aleatòries existeixen algunes condicions. Normalment es fan servir les inicialitzacions aleatòries ja que al no saber com és la solució, no pot suposar-se res. En canvi, pot ser interessant partir d'una població prèviament creada per algun expert si el problema és molt complex o es coneixen certs aspectes, o simplement garantir que tots els individus tinguin un *fitness* amb un mínim de qualitat (**decimació**). En qualsevol cas, els individus generats han de tenir la suficient diversitat genètica per garantir que pot trobar-se una bona solució en un temps raonable.

4.4.1 Construcció d'arbres en la GP

A la fase d'inicialització cal definir la forma dels arbres, és a dir, el seu nombre de nodes i de branques. Segons Koza les combinacions que poden produir-se poden resumir-se en:

- ***Full***. Tots els individus tenen la profunditat màxima en totes les seves branques, és a dir, tots els arbres seran complerts.
- ***Grow***. Els individus de la població no tenen perquè ser complets.
- ***Ramped half and half***. Els individus tenen la mateixa probabilitat de ser inicialitzats *full* o *grow*, quedant repartits en grups segons el seu nivell de profunditat.
- ***Ramped grow***. De la mateixa manera que abans, es divideix la població en grups en funció del seu nivell de profunditat, però ara tots seguint el mètode *grow*.
- ***Ramped full***. Igual que abans, però ara tots els individus s'inicialitzen amb el mètode *full*.

4.4.2 Definició dels elements del genotip en la GE

Els individus en la GE es representen mitjançant cadenes de bits de la mateixa manera que en GA. L'única diferència és que en la GE aquests bits s'agrupen cada Y elements per formar codons, de tal manera que a efectes pràctics l'individu es representa mitjançant un conjunt d'enters. L'única restricció a l'hora d'inicialitzar un individu és garantir que els enters que formen part tinguin un rang que estigui entre $[0..2^Y - 1]$. La definició de Y és un punt important que es comentarà més endavant.

De la mateixa manera que en la GP existeixen diferents maneres de crear els arbres, en la GE es podria fer un símil amb els noms segons la quantitat de terminals que inicialment apareixen.

4.5 Avaluació dels individus

4.5.1 Execució de l'arbre n-ari en la GP

Una de principals virtuts de la representació de la GP és que els programes que representen els individus són directament executables. Independentment si representa una funció o un programa, l'avaluació consisteix en substituir els terminals de les fulles a mesura que es va recorrent l'arbre del node arrel fins les fulles.

4.5.2 Execució de l'expressió lineal en la GE

GE no permet una execució directa dels individus que representen els individus a diferència del que succeeix amb la GP, ja que els individus són només cadenes d'enters. Per tal d'obtenir el programa que representa l'individu cal aplicar un procés de mapeig genotip-fenotip mitjançant una gramàtica BNF que modela les propietats/forma que hauria de tenir el programa a generar. Una gramàtica BNF està formada per un tupla $\{T, N, P, S\}$ on:[1]

- T (*Terminals*): Són els elements que poden aparèixer en un programa.
- N (*Non-Terminals*): És el nom de les produccions que hi ha en la gramàtica.
- P (*Productions*): Per cada producció, es defineixen un conjunt de regles.

[1]Els termes 'terminals' i 'no terminals' tenen significats diferents en la GP i la GE.

- S (*Starting production*): És la producció inicial de la gramàtica.

L'algorisme 4.2 detalla el procés de mapeig genotip-fenotip. El primer pas consisteix en agrupar els bits de l'individu en Y elements per obtenir un conjunt de codons (enters), i partir d'una expressió formada pels no terminals de la producció inicial (S). De manera iterativa, es van substituint els no terminals de l'expressió mitjançant l'equació 4.1, la qual reemplaça el no terminal actual pels elements associats a la regla que s'obté d'aplicar l'equació. Aquest procés es repeteix fins que l'expressió només conté terminals, i per tant, pot avaluar-se.

$$nova\ regla = resta\ de\ \frac{codon}{\#regles\ del\ no\ terminal} \tag{4.1}$$

El valor Y que agrupa els bits en codons depèn del nombre de regles de la producció més gran. Per exemple, si el codon està format per 8 bits només podrà seleccionar regles entre 0 i 255. A més, cal tenir en compte que el nombre de bits condiciona el procés de selecció de regles ja que afecta a la probabilitat de seleccionar una regla. Per exemple, si tenim 3 regles (A, B i C) i fem servir 2 bits ([0..3]) la combinació de valors 00 i 11 seleccionarà A, 01 seleccionarà B, i 10 seleccionarà C. Per tant, A tindrà més probabilitats de ser seleccionada que B i C. Aquesta problemàtica pot ser afrontada des des dos punts de vista: (1) Introduir regles buides, *introns*, per tal de permetre que totes siguin seleccionades amb la mateixa probabilitat; (2) Incrementar el nombre de bits per minimitzar l'impacte.

És important dimensionar correctament el nombre de codons d'un individu, ja que sinó poden produir-se situacions on s'hagin fet servir tots els codons i l'expressió que representa el programa encara tingui no terminals. Per evitar aquesta situació, la GE defineix una operació anomenada *wrapper* per crear un 'fenomen de solapament', de tal manera que els codons es tornen a reaprofitar quan ja s'han fet servir. Al mateix temps això genera un altre problema, i és que poden produir-se situacions on s'entri en un bucle infinit i l'avaluació no acabi mai. Aquest problema es resol amb la limitació del nombre de cops que s'aplica l'operació de *wrapping*.

Per tal d'aclarir aquest procés veurem l'exemple següent. Suposem que volem transformar l'individu {3, 5, 1, 4, 5, 6} en un programa a partir de la gramàtica BNF de la figura 4.4. Segons l'algorisme 4.2 inicialment es parteix dels no terminals de la producció inicial S, i de manera iterativa es van reemplaçant els no terminals fent servir l'equació 4.1 fins que tots els elements es converteixen en terminals. Aquest procés es detalla a la figura 4.5.

N = { <main>, <var>, <op>, }
T = {x, y, +, - }
S = {<main> }
P =
<main> → <var> <op> <var>
<var> → x | y
<op> → + | -

Figura 4.4: Exemple de la gramàtica BNF aplicada en l'exemple de la figura 4.5.

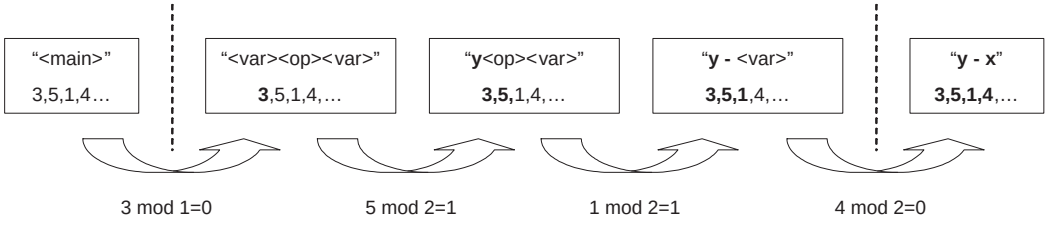

Figura 4.5: Exemple de traducció d'individu en programa en la GE.

Algorisme 4.2: Procés de mapeig genotip-fenotip.

```
Sigui G una gramàtica BNF representada per una matriu on el nombre de files representa les regles
de totes les produccions, i les columnes els seus elements de les regles
Sigui MAX_CODON el nombre màxim de codons d'un individu
Sigui MAX_WRAP el nombre màxim d'operacions de wrapping permeses
Sigui individual un array d'enters [MAX_CODON]
Sigui program un Vector de cadenes de caràters que guarda la traducció
Sigui fiTraduccio un booleà que indica la fi del procés de traducció
Siguin indexCodon, indexProgram, indexFiProgram, reglaSeleccionada, numElements, codon,
numNonTerminals, produccio, numWrap enters
//Inicialitzem l'individu
fiTraduccio=fals
numWrap=indexCodon=0
program[0]=S
indexFiProgram=numNonTerminals=1
Mentre (!fiTraduccio) fer
    indexProgram=0
    Mentre (indexProgram<indexFiProgram) fer
        Si (program[indexProgram] no és un Terminal) llavors
            codon=individual[indexCodon]
            produccio=program[indexProgram]
            reglaSeleccionada=codon MOD getNumRules(produccio)
            numElements=getNumElements(G[reglaSeleccionada])
            //Reemplaça els no terminals pels elements de la producció escollida
            program.remove(indexProgram)
            Per (int i=0;i<numElements;i++) fer
                program.add(indexProgram + i, getElement(i, G[reglaSeleccionada]))
            indexProgram=indexProgram+numElements-1
            indexFiProgram=indexFiProgram+numElements-1
            indexCodon++
            numNonTerminals=+getNonTerminals(G[reglaSeleccionada])-1
            Si (indexCodon==MAX_CODON) llavors
                //Aplicació de l'operador de wrapping. Es reusen els codons
                indexCodon=0
                numWrap++
                Si (numWrap==MAX_WRAP) llavors
                    //L'individu és massa llarg i ha de ser penalitzat
                    fiTraduccio=cert
        indexProgram++
    Si (numNonTerminals==0) llavors
        fiTraduccio=true
```

4.5.3 Càlcul del *fitness* dels individus

El *fitness* representa el grau d'adaptació d'un l'individu al problema: a millor *fitness*, millor solució. La forma de calcular el *fitness* depèn de cada problema, així com de la manera com evoluciona. Les equacions de càlcul pel *fitness* més habituals són les següents:

Càlcul del *raw fitness*. Mesura la quantitat d'error d'un individu al donar la solució, és a dir, la diferència entre el que dóna i el que hauria de donar (en el cas de que aquesta es conegui). Els individus més ben considerats seran aquells que tinguin el *fitness* més baix. El càlcul del *raw fitness* de l'individu i a l'instant t ve donat per la fórmula 4.2.

$$r(i,t) = \sum_{j=1}^{N_e} |(\text{Valor retornat per l'individu } i \text{ pel cas } j) - (\text{Valor correcte pel cas } j)| \quad (4.2)$$

Càlcul del *standarized fitness*. La manera com es calcula el *raw fitness* fa que el millor individu tingui el *fitness* més baix, i això pot no interessar a tots els problemes. Per exemple, en un problema d'optimització de costos sí que interessen els valors mínims i, per tant, en aquest cas el *standarized fitness* és directament el *raw fitness* (vegeu l'equació 4.2). En canvi, en altres problemes interessa calcular un valor màxim. En aquest cas el *standarized fitness* de l'individu i a l'instant t ve donat per la fórmula 4.2.

$$s(i,t) = \textit{fitness} \text{ que hauria de tenir el millor individu} - r(i,t) \quad (4.3)$$

Càlcul de l'*adjusted fitness*. L'*adjusted fitness* es calcula a partir del *standarized fitness*. El càlcul consisteix en mapejar el valor del *standarized fitness* entre 0.0 i 1.0. Quan el *standarized fitness* tendeix a 0 es calcula de la manera següent:

$$a(i,t) = \frac{1}{1 + s(i,t)} \quad (4.4)$$

Càlcul del *normalized fitness*. El *normalized fitness* es calcula a partir del *adjusted fitness* fent servir l'equació 4.5. El *fitness* dels individus està comprès entre 0.0 i 1.0, però la seva suma ha de ser igual a 1.0. Els millors individus tenen els valors més alts.

$$n(i,t) = \frac{a(i,t)}{\sum_{k=1}^{\text{mida de la població}} a(k,t)} \quad (4.5)$$

4.6 Estratègies per la selecció d'individus

La fase determina quins individus són els escollits per propagar el seu material genètic cap a la nova generació. Les estratègies de selecció més habituals són les següents:

Aleatòria (*Random*). La selecció aleatòria escull els individus de forma aleatòria sense tenir en compte el *fitness* ni cap altre paràmetre diferencial. La probabilitat de seleccionar un individu bo és la mateixa que la de seleccionar un de dolent, així que s'obtindria el mateix efecte que si exploréssim aleatòriament tot l'espai de cerca. Gairebé no s'utilitza.

***Greedy Over-Selection*.** Aquesta selecció té l'avantatge que molts cops permet reduir el nombre d'iteracions requerides per a que l'algorisme convergeixi. Tot i que els millors individus són els que tenen en principi més probabilitats de ser seleccionats, aquest mètode dóna més possibilitats als que no ho són. El mètode segueix aquests passos: (1) a partir del *normalized fitness* es divideix la població en dos grups. El primer conté el 20% dels millors individus; el segon l'altre 80%; (2) Es fa la selecció donant el 50% de probabilitats a cada grup, de manera que els individus del primer grup tenen moltes més probabilitats. La selecció dins d'un dels grups és *fitness-proporcionate*.

Proporcional al *fitness* (*Fitness proportionate*). Els individus s'escullen a partir del càlcul d'una probabilitat proporcional a l'avaluació del individu, és a dir, les seves possibilitats són directament proporcionals a la qualitat del *fitness*. A vegades passa que les diferències

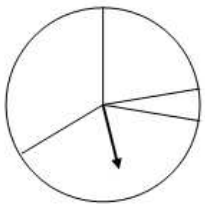

Figura 4.6: Exemple de selecció per ruleta.

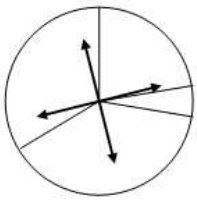

Figura 4.7: Exemple de selecció SUS.

entre els *fitness* dels individus d'una mateixa població són molt grans i el *fitness* d'un individu destaca per sobre de la resta, provocant com a resultat que en la següent generació la majoria de components siguin descendents d'aquests. Per tant, cal mecanismes per garantir una diversitat per afrontar la cerca de la solució amb un mínim de garanties.

Selecció per ruleta (*Roulette Wheel Selection*). El mètode assigna una importància diferent a cada individu segons la relació del seu *fitness* respecte el *fitness* global de tota la població. A partir d'aquesta puntuació es construeix com una ruleta, de tal manera que a cada individu se li assigna un interval. Tot seguit, es generen tants números aleatoris com mida tingui la població, i es seleccionen aquells individus on 'cau' el número generat dins el seu interval. La figura 4.6 mostraria una ruleta formada per 4 individus, on cadascun té un rang diferent segons el valor del seu *fitness*.

Selecció SUS (*Stochastic Universal Sampling*). Aquest mètode és semblant a la selecció per ruleta, però en aquest cas es disposa d'un conjunt virtual d'agulles equidistants que permeten seleccionar tots els individus al mateix temps. A la figura 4.7 es té la mateixa població que abans juntament amb 4 agulles. Dels quatre individus a seleccionar en l'exemple passaran la selecció tres dels individus diferents i el quart serà un duplicat. Amb aquest mètode s'aconsegueix una població més diversa que en el cas anterior.

Selecció per rang (*Rank selection*). En els dos mètodes anteriors poden aparèixer problemes quan les diferències entre els valors de *fitness* són molt grans. Si un individu té el 90% del sumatori dels valors de *fitness* és molt poc probable que algun dels altres individus sigui seleccionat. La selecció per rang ordena els individus del més dolent al més bo assignant a cadascun una posició. El *fitness* serà aquesta posició dividida pel sumatori de les posicions del rànquing. D'aquesta manera la diferència dels *fitness* queda atenuada, deixant més oportunitat als individus més dolents. Un cop hem assignat la probabilitat s'utilitza un mètode similar a la ruleta o a SUS per seleccionar els individus. Aquest mètode té com a desavantatge que requereix més temps per fer convergir l'algorisme perquè els individus que no són tan bons tenen més probabilitats de sobreviure. No obstant, aquest fenomen permet mantenir una diversitat en la població que ajuda a trobar millor la solució.

Selecció per torneig (*Tournament selection*). La selecció per torneig consisteix en enfrontar els individus uns contra els altres en diferents rondes per determinar els que tenen millor *fitness*. El procés finalitza quan hi ha N vencedors, essent N la mida de la població. El mètode proporciona més possibilitats als individus de *fitness* 'normal' que en els altres mètodes perquè únicament competeixen contra un sol individu de la població. Si dos individus bons s'enfronten, només un és seleccionat per a la següent generació. Això no és un problema perquè al disputar-se diferents rondes el risc de perdre individus bons és petit.

4.7 Operadors genètics

Els operadors genètics són els encarregats d'explorar l'espai de cerca de manera estocàstica. Es divideixen en dos grups en funció de la seva importància i aplicació:

Operadors primaris. S'apliquen sobre els individus de la generació actual per crear descendents (fills) basats en el seu material genètic (pares). Els operadors primaris són comuns per la GP i la GE, i es distingeixen dos tipus:

- **Reproducció**. La reproducció permet que l'individu passi a formar part de la següent generació sense que la seva informació sigui alterada, és a dir, es copia directament a la generació següent. És un operador asexual ja que només intervé un individu per generar el nou. S'aplica de la mateixa manera tant a la GP com a la GE.
- **Creuament**. Generalment és l'operador més predominant durant el cicle. El seu funcionament es basa en la reproducció biològica sexual que combina dos pares per traspassar la seva informació als fills de forma creuada. Els fills que es generen poden ser millors o pitjors que els pares.

  L'aplicació d'aquest operador està condicionat per la representació dels individus, ja que no és el mateix creuar dues estructures en forma d'arbre n-ari (en la GP), que lineals (en la GE). En el creuament es distingeixen dues fases:

  1. Selecció dels punts de tall dels pares que determinen el material genètic a intercanviar. L'intercanvi d'informació normalment es fa a un punt, és a dir, es selecciona un punt del primer pare i un altre del segon. No obstant, també poden fer-se creuaments de segments de material genètic delimitats per dos punts. La selecció dels punts pot fer-se:
     - **Aleatòria.** Els punts són seleccionats sense cap criteri.
     - **Creuaments intel·ligents.** Els punts són seleccionats com a base a algun criteri sobre el tipus de solució que es busca.
     - **Creuaments sensibles al context.** Segons el significat del punt inicial de tall, es busca un punt concret en l'altre pare.
     - **Creuament depenent de la profunditat.** Segons la profunditat s'escullen uns punts o altres.
  2. Intercanviar la informació a partir dels punts seleccionats.

  A la GP el creuament està lligat a una sèrie de condicions, com per exemple respectar la profunditat màxima de l'arbre i/o la relació entre els terminals i els no terminals per tal que sintàcticament l'arbre sigui vàlid. En canvi, a la GE no hi ha cap restricció sintàctica en el creuament al basar-se en una representació independent de la solució. Només cal respectar la mida màxima que pot tenir l'individu (O'Neill i Ryan, 2000).

Operadors secundaris. S'apliquen esporàdicament sobre els fills amb la finalitat de provocar canvis aleatoris i significatius per introduir millores. Els operadors secundaris més rellevants són els següents:

- **Mutació**. La mutació és un operador asexual que modifica una part seleccionada aleatòriament de l'individu per generar-ne un de nou. L'operador intenta contrarestar la possible homogeneïtzació de la població degut a la selecció. Permet introduir més riquesa genètica afegint funcions que s'havien perdut i que poden tenir un paper important en la generació de la solució final.

L'operador s'aplica de manera diferent a la GP i a la GE. En la GP el punt de mutació pot ser un node intern (funció) o una fulla (terminal). La seva aplicació eliminarà el node del punt de mutació i inserirà un subarbre generat aleatòriament. En canvi en la GE, la mutació implica canviar aleatòriament un enter per un altre (sempre dins el rang), aspecte que canviarà tota l'estructura del programa. La introducció de canvis aleatoris pot desencadenar que la cerca guiada es converteixi en aleatòria. Per evitar-ho aquest efecte la seva probabilitat d'aplicació ha de ser baixa.

- **Permutació.** L'operador permutació és un operador asexual de la GP que s'encarrega d'intercanviar les branques d'un node funció.
- **Edició.** L'edició és un operador asexual de la GP que opera sobre un únic individu per obtenir-ne un de nou, on es simplifica alguna branca seguint algun criteri específic.

  Es considera un mitjà per simplificar les expressions durant l'execució i millorar el rendiment. L'edició pot simplificar les expressions que no tinguin un efecte sobre el context o tinguin arguments constants.
- **Encapsulació.** L'encapsulació és un operador asexual de la GP que identifica automàticament subarbres útils i els substitueix per un node amb un codi especial que el representa. Les noves funcions encapsulades es guarden com nous nodes.
- **Modificació de constants.** Un dels problemes més difícils de gestionar en la GP és buscar funcions que tenen valors constants, perquè aleshores l'espai de cerca esdevé infinit a l'existir infinites constants. Aquesta problemàtica s'afronta des de l'aplicació de diferents estratègies de les quals destaquen:
  - **Perturbació de constants** (*Constant perturbation*). És una tècnica introduïda per Spencer (Spencer, 1994) que consisteix en modificar el valor de la constant en un $\pm 10\%$.
  - **Mutació numèrica** (*Numeric mutation*). Altera les constants a partir d'una distribució uniforme dins d'un cert rang a partir del valor de la constant (Evett i Fernandez, 1998).
- **Poda.** Els individus en la GE no fan servir necessàriament tots els seus gens en el procés de mapeig. Per exemple, per generar l'expressió $X * X + X$ només calen 5 gens, però l'individu que la representa possiblement en té més. La no utilització de tota la informació genètica pot repercutir negativament sobre l'operador de creuament degut a que pot succeir que només es creuïn informacions no útils, les quals no implicaran cap millora. Aquest efecte està il·lustrat a la figura 4.8. Per potenciar el creuament útil d'informació l'operador poda 'retalla' la informació que no es fa servir.
- **Duplicació.** Consisteix en realitzar una còpia d'un o més gens que han estat utilitzats de manera exitosa durant el procés d'evolució per potenciar la seva utilització. La presència de còpies de certs fragments de material genètic té diversos avantatges com per exemple evitar que es degradin/perdin degut a mutacions, o permetre que evolucioni de diferents maneres per trobar variacions millors basades en aquestes.
- **Canvis en la probabilitat dels operadors.** La influència dels operadors sobre els individus no té perquè ser la mateixa quan la població acaba de ser generada que quan ja està adaptada a un problema.
- **Generació de noves poblacions.** Quan no hi ha cap canvi en el millor individu de la població durant un cert nombre de generacions possiblement és perquè a la població li manca la suficient varietat genètica per continuar evolucionant. És en aquestes situacions quan pot ser interessant generar una nova població conservant el millor individu actual amb la finalitat de trobar-ne un de millor.

Un cop es disposa dels nous individus generats a partir dels operadors, el pas següent és decidir quins individus passen a formar part de la nova generació i quins moren, ja que aquests poden representar tant solucions bones com dolentes.

4.8 Polítiques de reemplaçament

Les polítiques de reemplaçament decideixen quins individus passen a la següent generació (entre els pares i els fills). Les principals polítiques es poden dividir en:

Aleatòria. A partir dels pares i dels fills, es seleccionen aleatòriament els individus perquè passin a formar part de la població de la generació següent.

Worst. Els fills substitueixen els pitjors pares. La població de la pròxima generació estarà formada pels millors individus entre els pares i els fills.

Generacional. La població de la nova generació està formada només pels fills.

Steady-State. L'objectiu és mantenir els millors individus de la població antiga i de la nova.

És important tenir en compte que l'elecció d'una estratègia o una altra influenciarà en la convergència i l'èxit de l'algorisme i, per tant, cal seleccionar-la segons la complexitat del problema.

4.9 Criteri d'acabament

Els criteris d'acabament de l'algorisme es poden dividir principalment en dos tipus:

Percentatge d'error acceptat. Normalment no es busca la millor solució possible sinó una aproximada per acotar l'espai de cerca i permetre resoldre el problema en un temps computacional raonable. Aquesta aproximació s'estableix a partir del criteri d'error màxim que l'usuari està disposat a acceptar.

Nombre màxim de generacions. L'usuari estableix el nombre de generacions que està disposat a esperar sense trobar l'individu que solucioni el problema amb l'error acceptat.

El primer criteri sovint no és trivial de definir perquè si es conegués com és la solució ja no s'aplicarien aquest tipus de tècniques. Per això, el segon criteri és el més habitual.

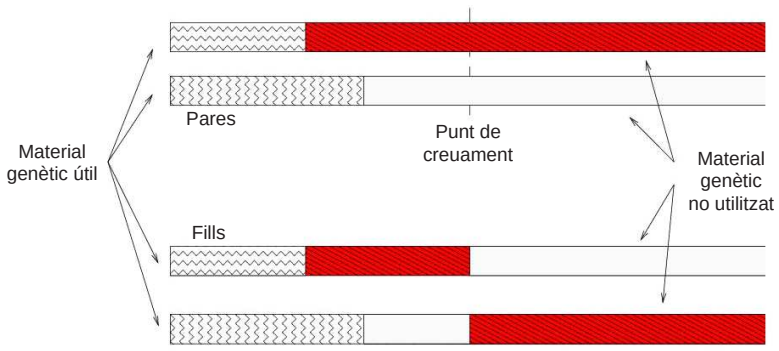

Figura 4.8: No tota la informació intervé en el creuament dels dos individus.

4.10 La GP versus la GE

Encara que la GP i la GE es basen en els principis dels GA per definir de manera automàtica un programa, cadascuna afronta el problema a la seva manera. Les seves principals diferències són:

Representació dels individus. La GP es basa en una representació no lineal i estructurada en forma d'arbre n-ari que permet d'una manera senzilla representar els elements del programa, així com la jerarquia entre ells. D'altra banda, la GE es basa en una representació lineal d'enters. Aquestes representacions condicionen la resta de fases de l'algorisme.

Espai de cerca. Els individus en la GE estan representats per enters sense cap sentit, en canvi, un cop són mapejats adquireixen significat. Aquesta separació de l'espai de cerca del de solucions permet una manipulació més senzilla i eficaç per part dels operadors. En canvi, els individus en la GP al ser directament les solucions necessiten sempre d'un procés de validació cada cop que s'han de modificar. Per tant, la GE simplifica la gestió dels individus.

Avaluació del programa. La representació de la GP permet obtenir el programa a executar directament els individus sense cap pas intermedi. En canvi, en la GE és necessari un mapeig genotip-fenotip sobre una gramàtica BNF per tal d'obtenir el programa que representa. Per tant, amb la GP aquest pas és més ràpid.

Operadors i restriccions. Ambdues representacions permeten afegir restriccions en els individus. En el cas de la GE això és transparent a l'algorisme ja que només s'apliquen els canvis sobre la gramàtica BNF, la qual s'utilitza només a l'hora d'executar l'individu. En canvi, en la GP afegir restriccions implica adaptar tots els operadors per garantir que els individus siguin vàlids. Per tant, la gestió de restriccions és més simple, ràpida i potent en la GE perquè és transparent.

En qualsevol cas, hi ha dos aspectes que influeixen sobre el temps necessari de cada aproximació per trobar una solució al problema:

Temps necessari per avaluar una generació. La transparència de les restriccions en la GE fa que l'execució d'una generació sigui normalment més ràpida. El coll d'ampolla en la GE es troba en el procés de mapeig de l'individu a la solució, i en la GP es troba en l'aplicació dels operadors.

Convergència de l'algorisme per trobar la solució. Tot i que amb la GE les solucions es troben abans que amb la GP perquè l'espai de cerca es redueix, si es realitzen restriccions massa estrictes pot succeir que la millor solució no es trobi.

Per tant, la complexitat, restriccions i tipologia del problema decantaran la balança cap a una aproximació o l'altra.

4.11 Consideracions abans de resoldre un problema

Per aplicar la GP o la GE sobre un domini cal prendre prèviament decisions sobre com definir i adaptar el problema, així com els paràmetres de l'execució. Les definicions que cal fer són:

Representació del programa. Independentment de l'enfocament, cal determinar quins són els elements del programa que poden intervenir, és a dir, les variables, constants, arguments i operacions. A més a més, també cal decidir com es relacionen entre ells els elements.

En la GP això s'engloba dins el conjunt de terminals i funcions així com en les restriccions a tenir en compte a l'hora d'aplicar els operadors. La seva elecció és fonamental per aconseguir trobar la solució, i aquesta ha de garantir les propietats següents:

- **Clausura**. Requereix que cada una de les funcions del conjunt de funcions sigui capaç d'acceptar tots els possibles valors o dades retornades per qualsevol funció del conjunt de funcions.
- **Suficiència**. El conjunt de funcions i terminals ha de ser suficientment representatiu perquè la solució es pugui arribar a trobar, és a dir, estigui en el domini que es representa pel conjunt de nodes.
- **Universalitat**. El conjunt de nodes ha de ser suficientment limitat perquè l'espai de cerca sigui assequible i les solucions es puguin arribar a trobar.

En canvi, en la GE la forma i el comportament del programa es representa mitjançant la gramàtica BNF, la qual modela els elements i l'ordre que han de seguir. Una definició incorrecta o massa restrictiva pot provocar que no es disposi de la suficient flexibilitat per trobar la solució.

La funció de *fitness*. Per la majoria de problemes el *fitness* es la mesura de l'error produït al executar el programa, és a dir, ens indica com de bo és el nostre programa. Si l'error és zero, s'ha trobat el programa ideal.

En canvi, als problemes d'optimització de control el *fitness* és la quantitat de temps, diners, etc. que es necessita per assolir l'objectiu. Si es tracta de reconèixer figures, es mesurarà segons el nombre de figures classificades correctament.

Per a d'altres problemes, és més apropiat fer servir un *fitness* multiobjectiu que combini factors com ara la correctesa, la parsimònia o l'eficiència.

Paràmetres de control d'execució. Per tal d'ajustar l'algorisme a cada problema, és convenient fer ús d'una sèrie de paràmetres que controlen l'execució de l'algorisme principal. Els més generals són:

- Mida de la població. Indica el número d'individus que té la població.
- Mida inicial dels individus. Indica la mida màxima dels individus a l'inici, i es mesura en nombre d'instruccions.
- Mida màxima dels individus. Indica la mida inicial permesa per a un individu en nombre d'instruccions.
- Freqüència dels operadors. Marca la probabilitat que s'apliqui cada operador.
- Mesura de l'error. La mesura de l'error determina com de bo és un programa.
- Inicialització de la població. Indica el mètode per inicialitzar la població.

A mesura que s'analitzin altres operadors i criteris d'execució apareixen més paràmetres i criteris de com ajustar cadascun d'ells per obtenir un correcte funcionament.

Condicions d'acabament. Indica els criteris que s'han de complir per finalitzar l'execució de l'algorisme. L'acabament pot ser degut a dues raons:

- Es troba un programa amb error acceptable dins d'un marge de tolerància.
- S'arriba a un nombre límit d'iteracions fixat per l'usuari. La solució s'obté de l'individu amb el millor *fitness*.

Els dos primers passos corresponen a especificar la representació dels individus, i els tres següents a l'execució de l'algorisme.

Resum

La CE és un paradigma basat en els principis de l'evolució natural i les lleis d'herència. De les diferents variants, la GP i la GE destaquen per tenir com a finalitat un dels grans reptes a l'àmbit de la Intel·ligència artificial: fer programes que defineixin programes.

Ambdues aproximacions estan basades en el cicle dels GA, el qual evoluciona una població d'individus al llarg d'un conjunt de generacions fins trobar la millor solució, o bé, una aproximada. El desenvolupament d'una generació es divideix en els passos següents: (1) es mesura el grau d'adaptació dels individus al medi amb el càlcul del *fitness*; (2) es tria el conjunt d'individus a partir dels quals es construirà una nova població; (3) es manipula el material genètic dels individus a partir dels operadors primaris i secundaris per obtenir nous individus; finalment, (4) es seleccionen els individus que passaran a formar part de la nova generació a partir dels individus de l'anterior generació i els nous generats amb els operadors.

Tot i que ambdues estratègies persegueixen el mateix objectiu, fan servir mecanismes molt diferents per aconseguir-ho. La GP es basa en disposar d'individus representats per arbres que representen directament el programa que es busca. La gran avantatge d'això és que es disposa sempre del programa a executar, però fa complexa la manipulació dels individus perquè cal garantir en tot moment la integritat i consistència del programa representat en l'arbre. D'altra banda, la GE disposa d'individus representats per cadenes d'enters que són transformats en programes mitjançant un procés de mapeig genotip-fenotip a partir d'una gramàtica BNF, la qual modela la forma que hauria de tenir el programa. Això simplifica la gestió de restriccions, però una definició massa restrictiva de la gramàtica pot impossibilitar trobar la solució.

Per tant, el paper de les restriccions per reduir l'espai de cerca i evitar que el problema esdevingui *NP-Hard* és el factor més rellevant per seleccionar una de les dues aproximacions.

Part II

Contribucions a l'organització de la memòria de casos del CBR

Capítol 5

Fase de recuperació

L'organització de la memòria de casos mitjançant clústers permet millorar l'eficiència i l'eficàcia de la fase de recuperació perquè permet ignorar la part del coneixement que no té res a veure amb el problema nou a resoldre. Aquesta recuperació selectiva té dues etapes: (1) seleccionar el conjunt de clústers que tenen un patró semblant al problema nou, i (2) recuperar un conjunt dels casos dels clústers. Tot i que això redueix la selecció de casos sorollosos i millora el temps necessari per explorar la memòria, l'agressivitat de l'estratègia pot conduir a resultats negatius segons la tipologia i complexitat de les dades. Això és degut a que la geometria de les dades condiciona la capacitat dels clústers per representar les dades. Aquest capítol presenta una metodologia per ajustar la fase de recuperació tenint en compte els dos aspectes anteriors: l'agressivitat de la reducció de l'espai de cerca i la complexitat de les dades. D'aquesta manera, serà possible estimar la resposta del sistema. Encara que aquesta metodologia es planteja de manera general per qualsevol mètode de clustering, la seva avaluació es realitza mitjançant SOM al ser la tècnica escollida en aquesta tesi. Finalment, la capacitat de SOM per organitzar la memòria es compara amb una altra proposta basada en l'algorisme X-means.

5.1 Motivació: trobar als escollits

La majoria dels problemes reals estan caracteritzats per disposar grans volums de dades, les quals sovint presenten imprecisions, incertesa i coneixement aproximat. És per aquest motiu que es fa necessari definir mecanismes que donin al sistema les capacitats necessàries per tal (1) seleccionar només els casos interessants i evitar tenir en compte casos redundants, i (2) reduir el temps computacional del sistema en cercar aquesta informació. És en aquest context on va plantejar-se l'organització del coneixement del sistema mitjançant SOM al capítol 1, ja que les seves capacitats de *Knowledge Discovery* i de *Soft-Compting* el feien un bon candidat per tractar amb aquest tipus de coneixement

El procés de recuperació en aquest context requereix de dues etapes: (1) seleccionar el conjunt de clústers que representen el cas d'entrada, i (2) recuperar un conjunt de casos dels clústers seleccionats. Tot i que això permet assolir les dues fites anteriorment esmentades, cal donar resposta a dues qüestions que defineixen l'agressivitat del mètode en quant a la reducció de l'espai de cerca desitjada:

- quants clústers es seleccionen? i sota quin/s criteri/s?
- quants casos es recuperen de cada clúster? i sota quin/s criteri/s?

La definició d'aquests paràmetres és crítica perquè el seu impacte depèn de la capacitat dels clústers per representar les dades, és a dir, de la seva geometria i complexitat.

Aquest capítol planteja com abordar la fase de recuperació d'una memòria de casos clusteritzada a partir de la definició d'una metodologia que tingui en compte d'una banda el rendiment desitjat per l'usuari (un compromís entre les capacitats resolutives i el temps de resposta) i, d'altra banda, la tipologia (complexitat) de les dades. Tot i que aquesta metodologia es planteja de manera general per qualsevol mètode de clustering, la seva avaluació es realitza mitjançant SOM. D'altra banda, la capacitat de SOM per organitzar el coneixement es compara amb una altra proposta desenvolupada al GRSI anomenada ULIC (Vernet i Golobardes, 2003), la qual es basa en l'algorisme X-means (Pelleg i Moore, 2000)

L'estructura d'aquest capítol és la següent. El punt 2 proposa la metodologia per modelar la recuperació d'una memòria clusteritzada. El punt 3 personalitza la metodologia sobre SOM per un ampli joc de dades. El punt 4 compara SOMCBR amb ULIC. Finalment, les conclusions i línies futures conclouen aquest capítol centrat a la fase de recuperació.

5.2 Metodologia per definir la recuperació més adient vers els requeriments i la complexitat del problema a tractar

Aquest apartat descriu els diferents components d'una metodologia per modelar la fase de recuperació d'una memòria de casos clusteritzada tenint en compte l'agressivitat desitjada per l'usuari i la complexitat de les dades. Primer, es presenta el mapa d'estratègies com una taxonomia de les diferents maneres en les quals la recuperació pot portar-se a terme tenint en compte el nombre de clústers i casos que es fan servir. A continuació, es presenta una proposta de *scatter plot* per analitzar d'una manera ràpida i fàcil els rendiment de totes les configuracions definides a la taxonomia anterior sobre un conjunt ampli de *datasets*. Finalment, es defineix una segmentació dels *datasets* segons la correlació entre la seva complexitat i el seu impacte respecte el mètode de clusterització.

El rendiment al llarg del capítol es considera com la relació entre la reducció del temps emprat en recuperar els casos més semblants, i el màxim increment d'error acceptat respecte l'error comès al fer servir tots els casos.

5.2.1 Mapa d'estratègies

El mapa d'estratègies és una taxonomia de les diferents maneres de recuperar casos d'una memòria clusteritzada. Tal com mostra la figura 5.1, poden considerar-se dos factors. D'una banda, el **factor dels clústers seleccionats** fa referència al nombre de clústers que es fan servir de la memòria de casos en el procés de recuperació. La selecció es fa tenint en compte el grau de similitud entre el patró que representa el clúster i el cas nou a resoldre. Aquesta similitud pot ser calculada mitjançant el complement de la distància Euclidiana normalitzada (vegeu l'equació 5.1), tot i que altres mètriques poden fer-se servir. Un valor similar a 1 indica que el clúster M_m és semblant respecte el cas d'entrada c_i. En canvi, un valor proper a 0 indica que són diferents. El símbol N representa el nombre d'atributs.

$$similitud(c_i, M_m) = |1 - distància(c_i, M_m)| = \left| 1 - \sqrt{\frac{\sum_{n:1}^{N}(c_i(n) - M_m(n))^2}{N}} \right| \qquad (5.1)$$

Tal com mostra la figura 5.1, s'identifiquen tres zones segons el nombre de clústers seleccionats. La zona delimitada per les àrees número 1 i 2 es refereix a situacions on només es selecciona el clúster més semblant. En canvi, la zona que defineixen les àrees 5 i 6 fa referència a una situació oposada on es seleccionen tots els clústers. Finalment, la situació intermitja es troba a la zona compresa per les àrees 3 i 4, on només una part dels clústers són seleccionats. En qualsevol cas, la selecció d'una d'aquestes situacions depèn de tres aspectes: (1) la capacitat dels clústers per

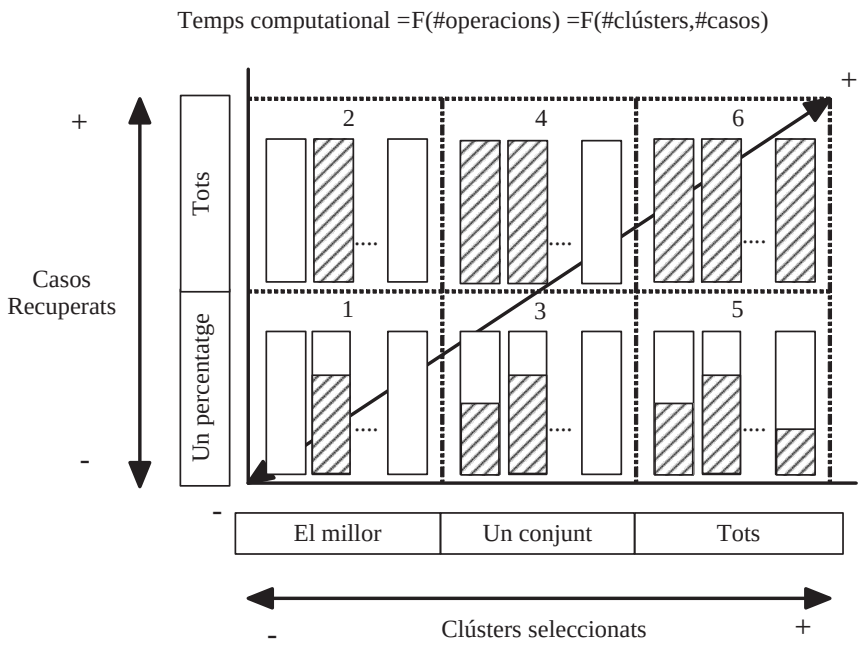

Figura 5.1: El mapa d'estratègies divideix les estratègies de recuperació en sis àrees. Cadascuna d'elles està definida per la combinació dels dos factors: el nombre de clústers seleccionats, i el nombre de casos recuperats de cada clúster. Els rectangles representen clústers, i l'àrea ratllada són els casos utilitzats de cadascun. La fletxa diagonal marca l'increment del temps computacional degut a l'increment de casos utilitzats.

representar les dades; (2) la millora de temps computacional esperada; i (3) la màxima reducció del percentatge d'encerts acceptada deguda a la reducció dels casos emprats. Per exemple, si es desitja una alta reducció del temps computacional s'han de seleccionar menys clústers per fer servir menys casos. No obstant, això pot degradar la capacitat resolutiva del sistema si els clústers no són representatius. Per tant, la selecció del nombre de clústers és un compromís entre els aspectes 2 i 3, els quals estan altament influenciats per la capacitat de modelar la complexitat de les dades (aspecte 1). Una manera d'automatitzar aquesta tasca és a partir de la definició d'un valor llindar (ϑ) que estableixi la mínima similitud acceptada entre el clúster M_m i el cas d'entrada C per considerar el clúster com a 'interessant'.

D'altra banda, el **factor dels casos recuperats** representa la quantitat de casos a recuperar de cadascun dels clústers que pot ser (1) un percentatge arbitrari, o bé, (2) tots els casos del clúster. Aquest aspecte marca la diferència entre les àrees de les zones anteriorment definides. Aquest estat intermig possibilita una reducció del temps computacional mantenint la capacitat per explorar altres clústers. Pot observar-se que l'àrea número 6 representa la situació on tots els clústers i tots els casos són seleccionats, com si d'un CBR amb una cerca lineal de la memòria de casos es tractés ($Tots\_Tots$).

La selecció del percentatge dels casos del clúster pot fer-se de moltes maneres, tot i que és obvi que cal tenir present la bondat del clúster: quan més s'assembla el clúster al cas d'entrada, més casos ha d'aportar. La primera proposta per definir el percentatge de casos és mitjançant la definició d'una relació lineal entre la contribució del clúster i la seva bondat. L'equació 5.2 resumeix aquest comportament com el quocient entre la similitud del cas nou c_i respecte el clúster M_m entre el sumatori de les diferències de les similituds del cas nou respecte tots els clústers.

$$\%\ dels\ casos\ de\ M_m = \frac{similitud(c_i, M_m)}{\sum_{m \in K_M} similitud(c_i, m)} \cdot 100 \tag{5.2}$$

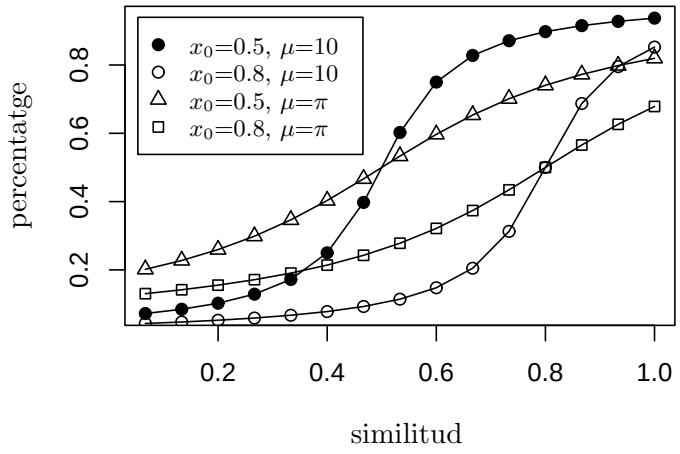

Figura 5.2: Representació gràfica de l'equació 5.3. Els arguments μ i x_0 ajusten la funció segons el gradient i el punt d'inflexió desitjat. Els valors $x_0 = 0.8$ i $\mu = 10$ són la configuració més restrictiva, i els valors $x_0 = 0.5$ i $\mu = 10$ la menys. Les altres dues configuracions són situacions intermitges.

Un altre enfocament pot ser el de definir una relació no lineal on s'accentuïn les situacions on el clúster és molt representatiu i on no. Aquest comportament pot modelar-se mitjançant la funció *arctangent*, la qual proporciona un comportament lineal per valors intermitjos del seu domini, i no lineal a mesura que arriba als extrems. Altres aspectes a tenir presents són la possibilitat de modelar el pendent de la corba, així com el moment a partir del qual es recompensa que el clúster sigui representatiu (punt d'inflexió). Aquestes consideracions queden reflectides a l'equació 5.3 proposada, on els paràmetres x_0 i μ representen la pendent de la corba i el punt d'inflexió respectivament. A més a més, per tal de traslladar el domini de l'arctangent de [-$\pi/2, \pi/2$] a [0, 1] cal dividir entre π i afegir 0.5.

$$\%\ de\ casos\ de\ M_m = 0.5 + \frac{arctg(\mu * (similitud(c_i, M_m) - x_0))}{\pi} \cdot 100 \tag{5.3}$$

La figura 5.2 mostra com els paràmetres μ i x_0 de l'equació 5.3 determinen el percentatge de la contribució d'elements. Valors alts de μ i x_0 impliquen una selecció més restrictiva. Contràriament, valors baixos condueixen a situacions de baixos nivells de restriccions.

Finalment, aquesta última equació pot normalitza-se amb la finalitat d'ajustar (incrementat o decrementant) la quantitat total de casos recuperats. Aquest aspecte es produeix si la similitud global entre el cas d'entrada i els clústers és globalment molt alta o baixa. Aquesta variant de l'equació 5.3 és l'equació 5.4.

$$\%\ of\ cases\ from\ M_m\ (normalized) = \frac{\%\ of\ cases\ from\ M_m}{\sum_{m \in K_M} \%\ of\ cases\ from\ m} \cdot 100 \tag{5.4}$$

Exemple. La figura 5.3 il·lustra com la combinació dels dos factors determina el grau de dispersió a través del qual el sistema explora els clústers en els quals s'han organitzat els casos. La part esquerra representa una memòria de casos clusteritzada en 9 clústers, on cadascun d'ells conté 100 casos i el seu nivell de bondat respecte el nou cas d'entrada està representat dins el clúster. La part dreta descriu el comportament de dotze estratègies sobre la memòria de la part dreta, fent servir diferents configuracions a nivell de nombre de clústers seleccionats, i a nivell de casos recuperats de cada clúster fent servir les equacions 5.2, 5.3, i 5.4. A més a més, a cada combinació dels factors

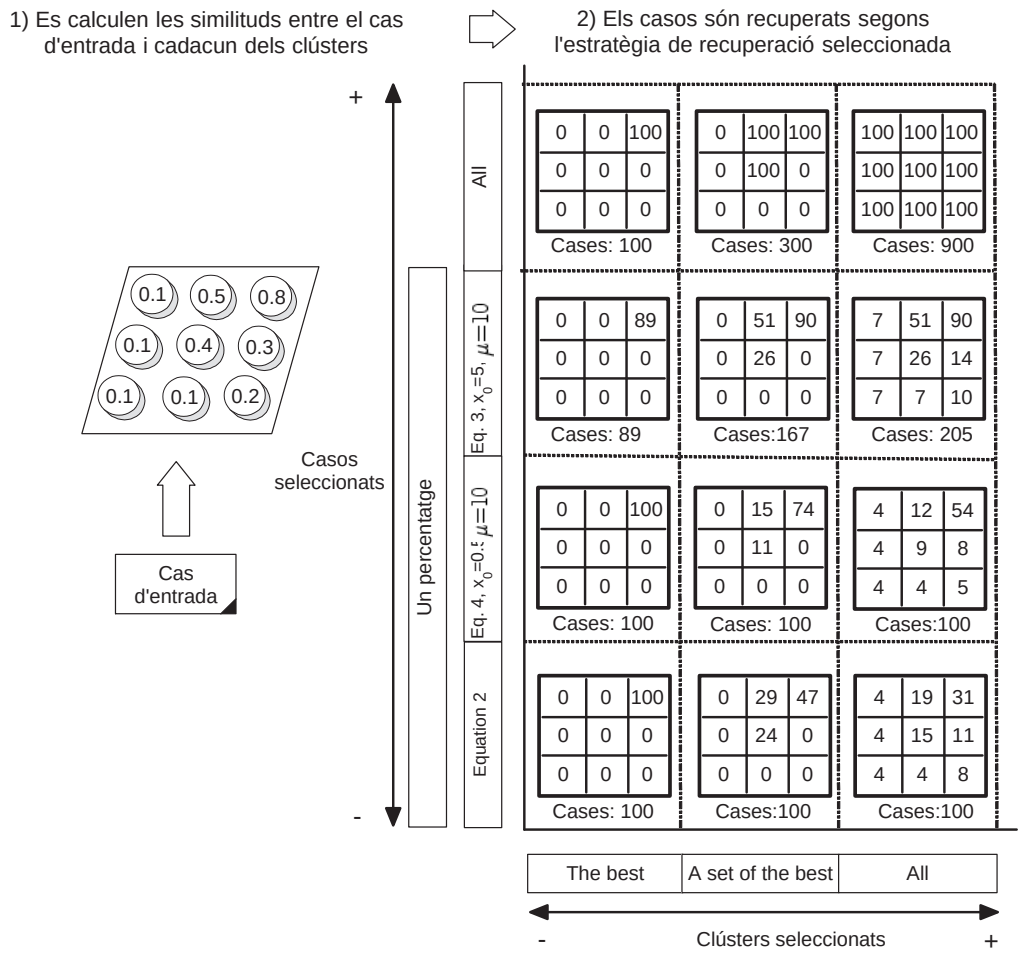

Figura 5.3: La part esquerra de la figura exemplifica una memòria de casos clusteritzada en 9 parts, i la part dreta mostra la dispersió en la qual explora la memòria segons l'estratègia seleccionada. Cada matriu de cada àrea correspon als casos recuperats de cadascun dels 9 clústers. Un valor zero a la casella indica que no s'ha seleccionat el clúster. El nombre total de casos recuperats a cada estratègia apareix a la part inferior de la matriu.

hi ha una matriu que representa el nombre de casos recuperats de cadascun dels clústers, i on el valor 0 significa que el clúster no ha estat seleccionat. La definició d'ambdós factors dependrà del rendiment desitjat per l'usuari, el qual estarà condicionat per la capacitat dels clústers per representar la complexitat de les dades.

Suposem una situació on es vol un temps de cerca reduït però que mantinguin les capacitats resolutives. Si els clústers són representatius, la millor estratègia consisteix en seleccionar només el clúster més similar respecte el nou cas, ja que aquest contindrà tota la informació interessant. No obstant, els clústers poden presentar una manca de precisió deguda a la complexitat de les dades. En aquest escenari, la millor solució és seleccionar més d'un clúster. Encara que aquesta decisió afecta al temps, aquest fet pot compensar-se aplicant estratègies que recuperen només una part dels casos dels clústers. En aquesta situació, les estratègies basades en les equacions 5.2 o 5.4 proporcionen el mateix nombre d'elements però de diferents clústers. Pot observar-se com l'estratègia 5.4 al no estar normalitzada proporciona un nombre d'elements diferent.

La situació extrema apareix quan la bondat del clúster respecte el cas d'entrada és reduïda, i es necessari una exploració completa de tots els clústers. En aquest cas, les estratègies que fan servir les equacions 5.2, 5.3, i 5.4 per seleccionar el nombre de casos dels clústers permeten explorar la memòria fent servir diferents graus de dispersió però sense fer servir tota la memòria.

5.2.2 Avaluació del rendiment de les estratègies de recuperació de casos

El següent pas després de definir la taxonomia de les estratègies és analitzar i comparar el seu rendiment sobre un conjunt ampli de *datasets* de diferents tipologies. No obstant, l'anàlisi dels resultats pot arribar a ser molt feixuc degut a la gran quantitat de resultats que s'obtenen. És per aquest motiu que s'ha desenvolupat un mètode àgil i visual per comparar el rendiment entre les estratègies a través de la representació gràfica 2D definida a la figura 5.4 (Martorell, 2007).

L'eix de les x fa referència al percentatge mig de casos de la memòria utilitzats per trobar l'element més semblant donada una estratègia. En el cas de la figura, l'estratègia $S2$ i $S3$ consulten de mitja només el 25% i el 15% de la memòria respectivament. Per tant, el complement d'aquest valor ens indica la reducció del nombre d'operacions respecte explorar tota la memòria, 75% i 85% respectivament. En el cas de la figura $S1$ seria l'estratègia de referència, la qual equivaldria a *Tots_Tots* segons la taxonomia de l'apartat anterior. La relació es representa en escala logarítmica amb la finalitat de millorar la visualització de les relacions on hi ha una reducció mitja-alta dels casos, ja que és l'escenari habitual on s'ubiquen les estratègies. Un valor proper a 0 indica que no hi ha reducció en el nombre d'operacions, i un valor negatiu indica millora.

D'altra banda, l'eix de les y avalua com de bé funciona una estratègia sobre D *datasets*. S'ha descartat fer servir la mitja del percentatge d'error de l'estratègia sobre els D *datasets* per dos motius: (1) La mesura no és robusta als *outliers* perquè aquests introdueixen soroll a la mitja; (2) És difícil mesurar quins mètodes són millors degut a que les mitges dels errors poden ser molt semblants. No obstant, aquestes dues problemàtiques poden ser pal·liades si fem servir el rànquing mig enlloc de la mitja dels errors. Si considerem E estratègies avaluades sobre D *datasets*, $R_{e,d}$ és el rànquing (ordenació entre 1 i E dels algorismes segons el percentatge d'error més baix) de l'estratègia e respecte el dataset d. A partir d'això, R_e és el rànquing mig de l'estratègia e calculat com el sumatori dels rànquings mitjançant l'equació 5.5.

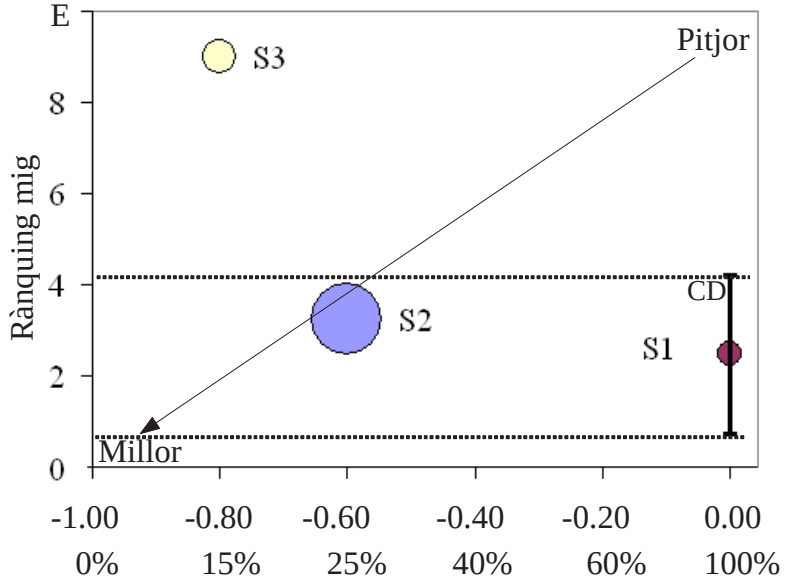

Figura 5.4: Representació gràfica per comparar el rendiment de diferents estratègies ($S1$, $S2$, i $S3$). L'eix de les x mesura la millora del temps computacional com el quocient del nombre d'operacions d'una estratègia (S2 o S3) respecte la de referència (S1), és a dir, el percentatge de casos utilitzats de la memòria. L'eix de les y representa el rànquing mig de les estratègies sobre els D *datasets*. Les dues línies horitzontals delimiten la zona on les estratègies tenen un rànquing estadísticament equivalent (CD).

$$R_e = \frac{\sum_{d=1}^{D} R_{e,d}}{D} \tag{5.5}$$

Valors propers a 1 indiquen que l'estratègia sol ser la millor sobre els D *datasets*, mentre que valors propers a E indiquen l'efecte oposat. Com es pot veure a la figura 5.4 el rànquing mig està representat per unes boletes que tenen radis diferents. En aquest cas, el radi representa la desviació estàndard de R_i: a major mida, més gran és la variabilitat del seu comportament respecte el rànquing. Tenint en compte això veiem com l'estratègia $S1$ és millor que $S2$, i $S2$ millor que $S3$. No obstant, això ens planteja una nova qüestió: dues estratègies poden ser equivalents en quant a rànquings? Aquesta qüestió es tracta a partir de la definició del *Critical Distance* (CD).

CD és una mesura que defineix la mínima distància a partir de la qual es pot considerar que la diferència de rànquing entre dos R_i és significativa donat un cert nivell de confiança. La zona que delimita el CD es representa mitjançant dues línies discontinues horitzontals dins els quals les estratègies es consideren equivalents des del punt de vista de rànquing. A l'exemple de la figura, $S1$ i $S2$ tenen un rànquing equivalent, però ambdues són significativament diferents respecte $S3$. El valor del CD està basat en el mètode de Bonferroni-Dunn (Demsar, 2006), i el seu valor s'obté a través de l'equació 5.6, on Z és un valor estadístic obtingut a partir del nivell de significància desitjat (Martorell, 2007).

$$CD = Z\sqrt{\frac{E(E+1)}{D}} \tag{5.6}$$

Finalment, veiem perquè el rànquing mig és una mesura de la bondat que ens evita els dos problemes anteriorment citats. Pel que fa als *outliers*, aquests tenen un efecte més baix sobre la mitja si una estratègia funciona molt malament per un problema. A més a més, possiblement totes les estratègies es veuran afectades i el rànquing per aquell problema no distarà molt sobre la mitja de problemes. D'altra banda, el fet de treballar amb rànquings fa que la distància entre els mètodes sigui de com a mínim de 1 a cada dataset i no de dècimes com en el cas del percentatge d'error, aspecte que accentua les diferències significatives entre els resultats a l'hora de comparar dos algorismes amb valors absoluts similars (Sheskin, 1997).

5.2.3 L'impacte de la complexitat de les dades

L'esquema gràfic anterior permet d'una manera ràpida i àgil comparar el rendiment d'un conjunt d'estratègies. No obstant, aquest anàlisi no té en compte un aspecte clau per l'assoliment de l'èxit de les estratègies: la capacitat dels clústers per representar els patrons de les dades, aspecte que està directament relacionat amb la complexitat d'aquestes. Per això, aquest punt proposa una segmentació de les dades segons la seva complexitat i, d'aquesta manera, analitzar el gràfic anterior per cadascun d'aquests grups per tal d'obtenir conclusions més fiables.

L'estudi de la complexitat fa referència a la caracterització de la complexitat intrínseca del dataset, i a l'estudi del seu impacte sobre el rendiment del classificador (Basu i Ho, 2006). D'una banda, la complexitat de les dades està principalment relacionada a tres causes: (1) l'ambigüitat de classe, (2) la complexitat de la frontera, i (3) la disparitat del conjunt d'entrenament. No obstant, degut a la dificultat de determinar els aspectes 1 i 3, els estudis actuals es centren en l'aspecte 2. Ho & Basu (Ho i Basu, 2002) van proposar al 2002 un espai de mesures per identificar els diferents aspectes de la complexitat de la frontera basat en el poder discriminant dels atributs, la separabilitat de classes i la topologia de les classes. D'altra banda, no totes les mètriques estan igualment correlacionades amb el rendiment del classificador. Per aquest motiu, es fa necessari un estudi per veure quines d'elles poden aportar-nos major informació per discernir entre les tipologies de dades existents segons el rendiment del classificador.

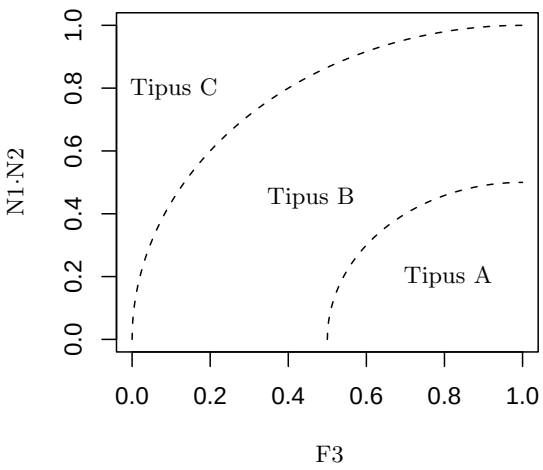

Figura 5.5: El mapa de complexitats està definit mitjançant les mètriques N1, N2 i F3. La seva combinació defineix 3 zones, on A és la zona de menor complexitat i C la de major complexitat.

De l'estudi de les diferents mètriques proposades per Ho & Basu (Ho i Basu, 2002) respecte el rendiment del SOMCBR (veugeu l'apèndix G) (Fornells et al., 2007b) s'identifiquen tres mètriques interessants:

- ***Feature efficiency* - F3**. Mesura el grau d'eficiència individual de les característiques per separar les classes.
- ***Length of the class boundary* - N1**. Mesura la distància entre instàncies de classes diferents properes.
- ***Intra/inter class nearest neighbor* - N2**. Mesura la dispersió de les classes.

Tanmateix, com diuen els seus autors, és la combinació d'aquestes mètriques i no el seu ús individual el que ens permet construir un espai fiable per separar comportaments.

La figura 5.5 representa l'espai de complexitats que combina les mètriques més correlacionades amb el SOMCBR. L'eix horitzontal F3 representa el poder discriminant dels atributs. A major valor, major el poder discriminant. L'eix vertical representa la separabilitat de classes mitjançant el producte de N1 i N2. S'ha fet servir el producte d'ambdues mètriques perquè les dues fan referència a la distància entre classes oposades i, per tant, el seu producte potencia el seu significat per determinar el grau de separabilitat. Tot i que un valor baix ens indica que hi ha una alta separabilitat de classes, un valor alt no ens indica necessàriament l'efecte invers.

El punt (1,0) es considera el punt de menor complexitat (mCP - *minimum complexity point*), i el punt (0,1) fa referència al punt de major complexitat (MCP - *maximum complexity point*). A partir d'aquests punts és possible definir tres tipus segons la zona:

- **Tipus** A: problemes amb una baixa complexitat (distància $<$ a 0.5 respecte el mCP).
- **Tipus** B: problemes amb una complexitat mitjana (distància entre 0.5 i 1 respecte el mCP).
- **Tipus** C: problemes amb una alta complexitat (distància $>$ 1 respecte el mCP).

5.3 Aplicació de la metodologia sobre el SOMCBR

Aquest punt avalua la metodologia proposada a l'apartat anterior pel cas del SOMCBR, el qual és un CBR amb la memòria de casos clusteritzada mitjançant SOM. Primer, s'introdueixen els jocs de dades que es fan servir, així com la complexitat associada a cadascun d'aquests. A continuació, es plateja el mapa d'estratègies i s'avalua per cadascun dels tipus de complexitats l'*scatter plot* que compara el percentatge de casos utilitzats respecte el rànquing mig que té l'estratègia. Finalment, s'analitzen i es discuteixen els resultats.

5.3.1 Experimentació

L'aplicació de la metodologia per l'estudi de les estratègies de recuperació es realitza a partir de diversos *datasets* amb característiques i dominis diferents. La taula 5.1 mostra els 56 *datasets* utilitzats, on els *datasets* miasbi, mias3c, ddsm, i μCa provenen de l'HRIMAC (vegeu l'apèndix B) i la resta de l'*UCI Repository* (Asuncion i Newman, 2007). Tots els *datasets* de J classes han estat convertits a J *datasets* de dues classes (cada classe contra la resta de classes) per tal d'incrementar el joc de dades. La taula indica el nom, el nombre d'atributs i instàncies, així com el tipus de complexitat per cada *dataset* assignat mitjançant el mapa de complexitats de la figura 5.6.

Taula 5.1: Descripció dels *datasets* utilitzats: nom, nombre d'atributs i d'instàncies, i tipus de complexitat. El sufix $2cX$ indica que el dataset classifica la classe X respecte la resta de classes.

Dataset	**Atributs**	**Instàncies**	**Tipus**	***Dataset***	**Atributs**	**Instàncies**	**Tipus**
segment2c2	19	2310	A	wav2c3	40	5000	B
iris2c2	4	150	A	wav2c1	40	5000	B
glass2c1	9	214	A	miasbi2c3	152	320	B
thy2c1	5	215	A	ddsm2c1	142	501	B
thy2c2	5	215	A	mias3c2c2	152	322	B
segment2c6	19	2310	A	thy2c3	5	215	B
segment2c7	19	2310	A	mias3c2c1	152	322	B
wine2c2	13	178	A	ddsm2c4	142	501	B
iris2c1	4	150	A	miasbi2c2	152	320	B
segment2c1	19	2310	A	wisconsin	9	699	B
wine2c1	13	178	A	wbcd	9	699	B
glass2c2	9	214	A	wav2c2	40	5000	B
miasbi2c4	152	320	A	sonar	60	208	B
glass2c4	9	214	A	wpbc	33	198	B
wine2c3	13	178	A	glass2c6	9	214	B
iris2c3	4	150	A	mias3c2c3	152	322	B
wdbc	30	569	A	biopsia	24	1027	B
segment2c3	19	2310	B	vehicle2c3	18	846	B
segment2c5	19	2310	B	vehicle2c2	18	846	B
glass2c3	9	214	B	bal2c3	4	625	C
vehicle2c1	18	846	B	bal2c2	4	625	C
segment2c4	19	2310	B	bal2c1	4	625	C
tao	2	1888	B	ddsm2c3	142	501	C
hepatitis	19	155	B	heartstatlog	13	270	C
glass2c5	9	214	B	μCa	21	216	C
ionosphere	34	351	B	ddsm2c2	142	501	C
vehicle2c4	18	846	B	pim	8	768	C
miasbi2c1	152	320	B	bpa	6	345	C

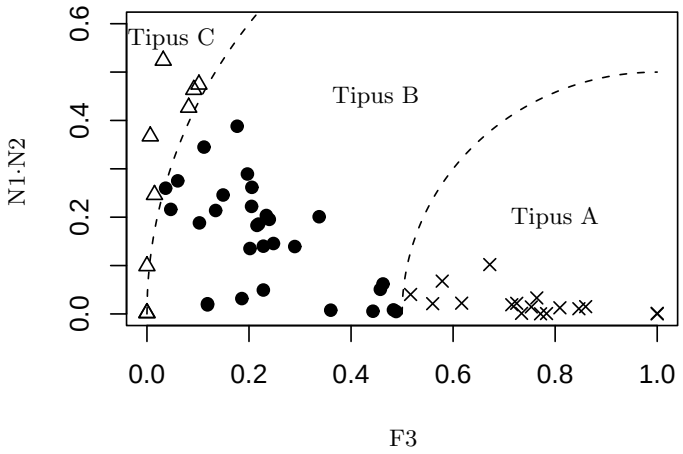

Figura 5.6: Mapa de complexitat dels 56 *datasets* analitzats.

5.3.2 Definició del mapa d'estratègies

El primer component que ens cal per aplicar la metodologia és definir un conjunt d'estratègies representatiu per analitzar el SOMCBR a través de la definició del mapa d'estratègies, el qual està esquematitzat a la figura 5.7. Pel que fa al nombre de clústers seleccionats, se n'han seleccionat 3 com a estat intermig entre seleccionar el millor clúster i tots, ja que sense ser un nombre elevat de clústers ens permet no perdre casos que puguin estar en un clúster veí dins del nostre espai de dues dimensions. D'altra banda, el factor del nombre de casos recuperats s'estudia des de 5 situacions, les quals fan referència des de recuperar un percentatge dels casos de cada clúster seleccionat segons la seva bondat (vegeu les equacions 5.2, 5.3, 5.4) fins a la situació on es recuperen tots els casos dels clústers seleccionats. S'han seleccionat els paràmetres $x_0 = 0.5$ i $x_0 = 0.8$ amb $\mu = 10$ perquè són dues configuracions que trenquen la linealitat de l'equació 5.2, i així pot avaluar-se una situació conservativa i una altra més restrictiva respectivament segons el valor de x_0. Hi ha quatre configuracions (marcades amb una creu) que s'ignoren ja que equivalen a la configuració *Tots_1Millor*. A partir d'aquestes combinacions representades al mapa de la figura 5.7 s'avaluen els 56 *datasets* de la taula 5.1. Els paràmetres de configuració comuns per a totes elles són els següents:

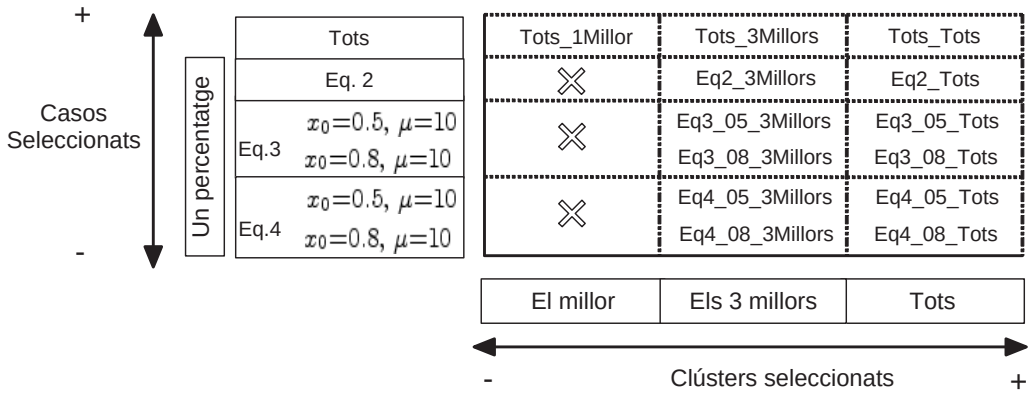

Figura 5.7: Mapa d'estratègies de les diferents maneres de recuperar casos de la memòria clusteritzada. Els quadres marcats amb una creu representen configuracions ignorades perquè són el mateix que *Tots_1Millor*.

- La funció de distància utilitzada tant per la construcció dels models, com per la comparació entre clústers i casos és la del complement de la funció Euclidiana (vegeu l'equació 5.1).
- La mida del mapa s'assigna de manera automàtica tal com s'explica al capítol 3, és a dir, es selecciona la mida que minimitza l'error. El rang de mides avaluades va de 2 a 6.
- Els models poden tenir diferent nombre de casos.
- La fase d'adaptació proposa la nova solució fent servir el cas recuperat més semblant.
- La fase d'emmagatzematge no guarda nous casos.
- Cada resultat s'obté d'aplicar un *10-fold stratified cross-validation*.
- Cada configuració és la mitja de 10 llavors per tal de compensar els efectes aleatoris de la construcció dels models.

5.3.3 Avaluació de les estratègies segons la complexitat de les dades

Tenint en compte les configuracions del mapa d'estratègies i la complexitat de cadascun dels *datasets* de la taula 5.1, el pas següent és analitzar el comportament del SOMCBR per cadascuna de les tipologies de dades mitjançant els *scatter plot* de les figures 5.8, 5.9 i 5.10. En tots els casos l'estratègia de referència és la que equivaldria a realitzar una cerca amb tots els elements de la memòria de casos, és a dir, recuperar tots els casos de tots els clústers (*Tots_Tots*) com si d'un CBR amb una memòria de casos lineal es tractés. Un cop fet una anàlisi individual, es discutirà sobre quines configuracions són les més rellevants.

La figura 5.8 representa l'estudi pels *datasets* de complexitat baixa (tipus A). La disposició dels cercles que representen les estratègies al llarg dels dos eixos ens dóna una idea de la relació lineal entre les dues components per aquest escenari. De fet, el càlcul de la correlació entre els valors de les estratègies que tenen una reducció significativa dels casos utilitzats és del 0.96 (per les estratègies que fan servir només del 0% al 25% de casos de la memòria). Per tant, l'efecte del SOM per problemes senzills és feble, fent que el percentatge d'encerts sigui proporcional a la reducció realitzada per moltes de les configuracions.

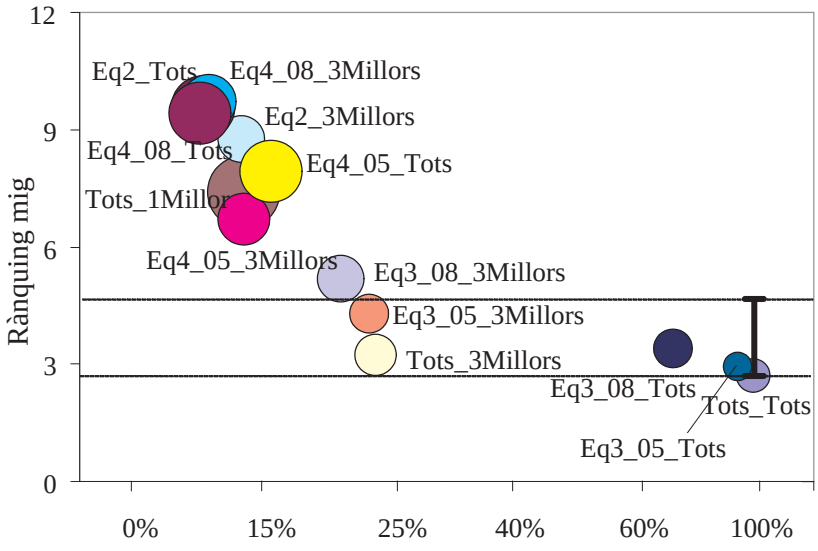

Figura 5.8: *Scatter plot* de les estratègies analitzades sobre els *datasets* del tipus A.

No obstant això, hi ha un conjunt d'estratègies que són interessants perquè estan dins l'àrea del CD:

- *Eq3_05_Tots*. Dóna gairebé els mateixos resultats que l'estratègia de referència ja que selecciona tots els clústers, i de cadascun selecciona un alt percentatge de casos.
- *Eq3_08_Tots*. Igual que l'estratègia anterior, però els efectes de moure el llindar x_0 possibiliten fer servir en mitjana només un 70% de la memòria de casos, és a dir, hi ha una reducció del 30% dels casos.
- *Tots_3Millors*. Aquesta estratègia accentua encara més els efectes de l'estratègia anterior ja que només selecciona 3 clústers, provocant una reducció mitjana del 73% dels casos.
- *Eq3_05_3Millors*. Manté una reducció important del nombre de casos a utilitzar, tot i que les capacitats resolutives es veuen lleugerament afectades per la reducció de casos al recuperar només un percentatge dels 3 clústers més semblants. En qualsevol cas, aquesta reducció no és significativa.

La figura 5.9 representa l'estudi pels *datasets* de complexitat mitjana (tipus B). L'increment de la complexitat té dos efectes positius a destacar respecte els resultats anteriors:

- El nombre de casos utilitzats es redueix en la majoria de les estratègies. Els dos grups anteriors (un ubicat al 12% i l'altre al 22% de casos utilitzats de la memòria) redueixen encara més els casos que fan servir (6% i 19 % respectivament). Això vol dir que els clústers que es creen són cada cop més específics, i poden representar millor la geometria de les dades perquè aquesta comença a ser complexa.
- La linealitat dels dos components dels eixos passa del 0.96 al 0.86. Aquest efecte apareix a l'haver més complexitat, ja que ara els clústers poden modelar millor les particularitats dels casos en patrons.

El comportament de les millors estratègies és en línies generals igual, tot i que ara hi ha més reducció. L'excepció apareix a l'estratègia *Eq3_05_3Millors*, la qual és per poc significativament diferent.

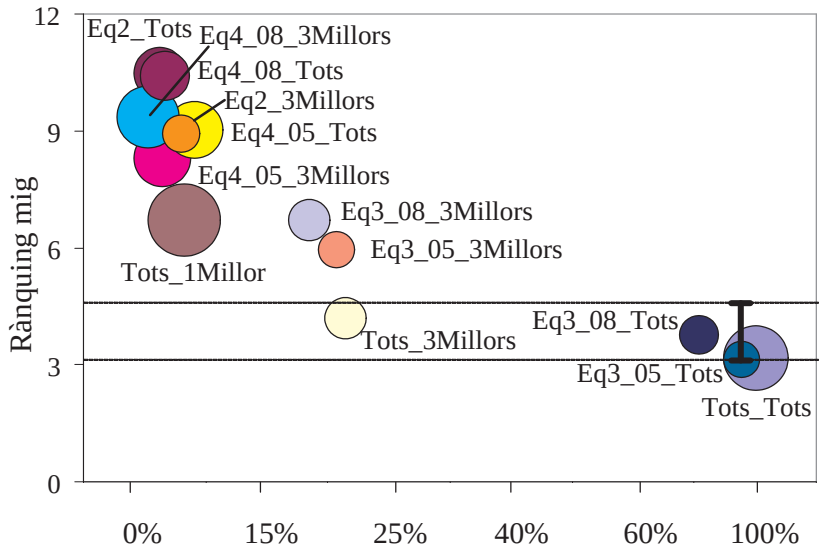

Figura 5.9: *Scatter plot* de les estratègies analitzades sobre els *datasets* del tipus B.

Finalment, la figura 5.10 representa l'estudi pels *datasets* de màxima complexitat (tipus C). L'increment de la complexitat continua afectant positivament al comportament del SOMCBR:

- Continua la reducció dels casos utilitzats de la memòria de casos, tot i que en aquest cas és més lleugera.
- La correlació entre les components dels dos eixos baixa fins al 0.76.

A més a més, en aquest escenari es posen totalment de manifest les capacitats *Soft-Computing* i de *Knowledge Discovery* del SOM ja que:

- L'estratègia *Eq3_05_Tots* redueix gairebé un 20% els casos utilitzats i proporciona millors resultats que l'estratègia de referència, i gairebé de manera significativa.
- L'estratègia *Eq3_08_Tots* accentua els efectes anteriors, reduint un 40% els casos utilitzats de mitjana per recuperar el cas més semblant, i proporciona resultats molt destacats respecte l'estratègia de referència.
- L'estratègia *Tots_3Millors* accentua encara més els efectes anteriors, provocant una reducció del 80% dels casos, tot oferint unes capacitats resolutives millors que l'estratègia de referència encara que no significativament.
- L'estratègia *Eq3_05_3Millors* manté una reducció semblant a l'anterior però amb les capacitats resolutives una mica reduïdes, tot i que no són estadísticament diferents.
- L'estratègia *Eq4_05_Tots* provoca una reducció encara més dràstica que les anteriors, al voltant del 90%, i proporciona resultats semblants als de l'estratègia de referència.

Per tant, aquest tercer escenari on SOM funciona millor ja que els clústers són capaços de representar d'una manera més precisa la geometria de les dades, aconseguint fins i tot, millorar el percentatge d'encerts obtingut al fer servir tota l'experiència ja que els casos sorollosos són descartats. L'efecte de la millora del percentatge d'error pot veure's millor a les gràfiques de la figura 5.11, on es relaciona l'error obtingut en cadascuna dels *datasets* de complexitat C per les mètriques de complexitat F3 i N1·N2. A més a més, pot observar-se l'alta correlació d'ambdues mètriques respecte la precisió del SOMCBR.

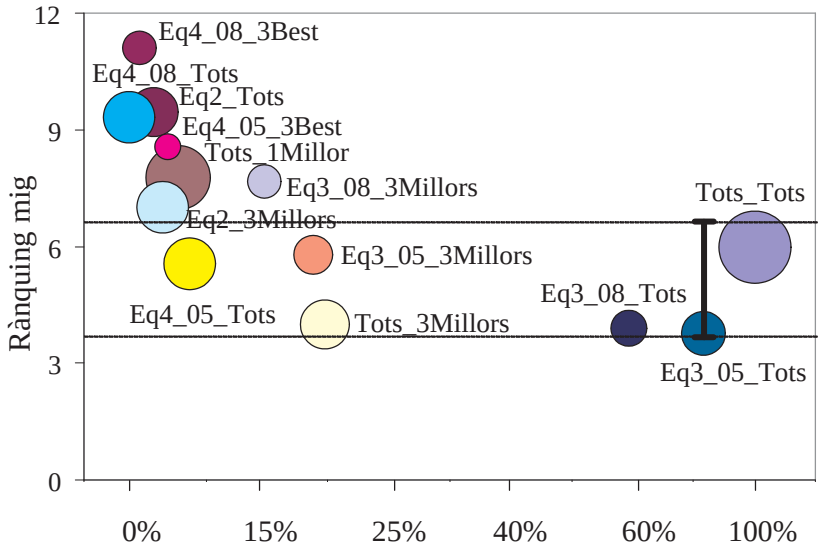

Figura 5.10: *Scatter plot* de les estratègies analitzades sobre els *datasets* del tipus C.

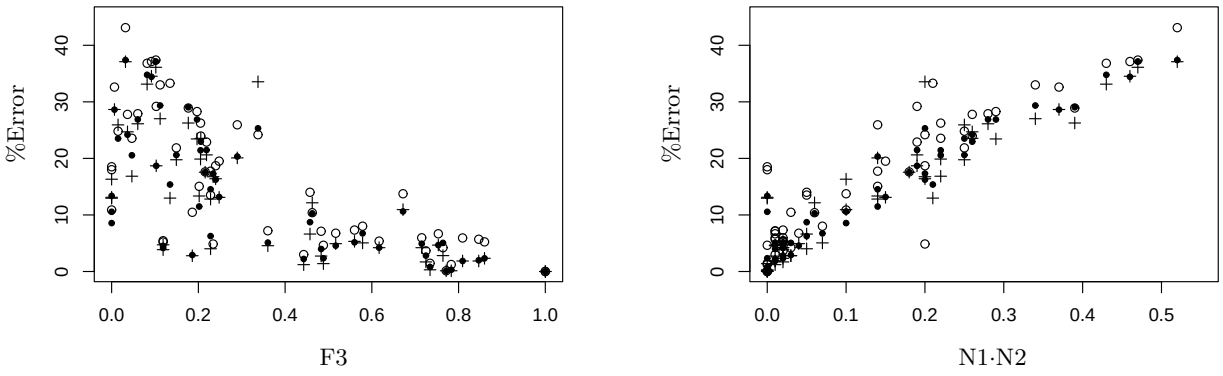

Figura 5.11: Relació entre el percentatge d'error del CBR (+), Eq2_3Millors (●) i Eq4_08_3Millors (○) respecte les mètriques N1·N2 i F3 pels 56 *datasets*.

5.4 Comparació del model de recuperació de dos nivells de SOM respecte d'altres

La recuperació de la memòria de casos organitzada amb SOM es basa en la definició de dos nivells: el nivell dels clústers seleccionats, i el nivell dels casos recuperats. Això permet discernir l'espai en segments (models), els quals després són explorats per dins per tal de veure l'element més interessant (casos). No obstant, hi ha aproximacions basades només en definir segments, on cada segment representa directament una classe. Aquest apartat té com a finalitat comparar aquestes dues aproximacions per analitzar els seus rendiments.

5.4.1 La plataforma ULIC

ULIC (*Unsupervised Learning in CBR*) (Vernet i Golobardes, 2003) és una plataforma CBR que organitza la memòria de casos mitjançant l'algorisme X-means (Pelleg i Moore, 2000). Aquesta plataforma, també desenvolupada dins el GRSI, planteja la recuperació dels casos d'una manera totalment diferent a la plantejada al SOMCBR. A continuació s'introdueix com organitza la memòria i, així com el procediment de recuperació.

L'organització de la memòria a ULIC requereix de dos passos tal com mostra la figura 5.12. El primer pas consisteix en definir tants grups de casos com L classes hi hagi. A continuació, s'aplica l'algorisme X-means a cadascun d'aquests grups. D'aquesta manera, s'obtenen un conjunt de subgrups representats cadascun d'ells per un centroide, el qual és la mitja dels casos del subgrup. La idea de X-means és definir el nombre K òptim de particions que es poden fer a l'espai amb l'algorisme K-means (Hartigan i Wong, 1979).

L'algorisme 5.1 mostra a alt nivell el cicle de funcionament d'ULIC. Ara, la fase de recuperació només està composada per un nivell de comparacions: el cas nou respecte els K centroides. Per tant, estem davant d'una manera molt diferent d'organitzar la informació, on a primera vista es poden identificar diversos punts febles:

- Pèrdua d'informació provocada per la no utilització dels casos dels subgrups.
- El fet de separar els casos segons la seva classe pot provocar que davant de casos propers, però de classes diferents, s'obtinguin patrons molt semblants amb classes totalment diferents. Per tant, el sistema es confondrà fàcilment

Algorisme 5.1: Funcionament de la plataforma ULIC.

Funció *ULIC* **és**
Sigui I el nou exemple a classificar
//Fase de recuperació
Seleccionar el centroide més similar respecte I aplicant una mètrica de distància
//Fase d'adaptació
La classe del problema nou és la del centroide
//Fase de revisió
Es valida la nova solució
//Fase d'emmagatzematge
Es refan els clústers

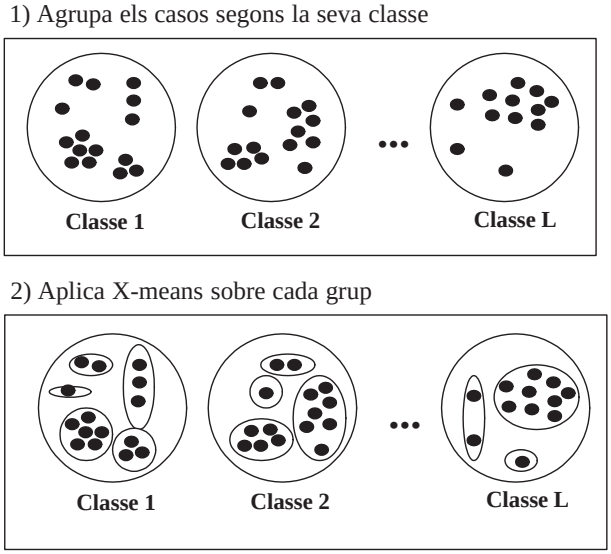

Figura 5.12: Organització de la memòria de casos a la plataforma ULIC.

El mètode té un gran avantatge fruït de la seva simplicitat: requereix molt poques operacions de comparació per tal de proposar una solució.

5.4.2 Experimentació

La comparativa de SOMCBR respecte ULIC es farà des del punt de vista del rendiment respecte un CBR sense organització de la memòria de casos: percentatge d'error i temps en recuperar l'element més similar. Els jocs de dades utilitzats es troben detallats a la taula 5.2, i provenen de l'*UCI Repository* (Asuncion i Newman, 2007) i del projecte HRIMAC (vegeu l'apèndix B). La taula conté − a part del nom del dataset, el nombre de casos, instàncies i classes − una columna que ens indica si el dataset presenta incertesa. Això serà interessant per veure si les capacitats *Soft-Computing* de SOM per tractar millor aquest tipus de coneixement proporcionen un tret diferencial.

Com que ULIC només fa servir la informació del clúster més semblant, l'estratègia de recuperació de SOMCBR es configurarà perquè només faci servir la informació del clúster més semblant i d'aquesta manera, que els dos sistemes estiguin en condicions similars. Altres paràmetres comuns:

- La funció de distància utilitzada tant per la construcció dels models, com per la comparació entre clústers i casos és la del complement de la funció Euclidiana (vegeu l'equació 5.1).
- La mida del mapa i de K s'assigna de manera automàtica seguint la condició que l'error dels

clústers sigui mínim. El rang de mides avaluades va de 2 a 12.

- Els models poden tenir diferent nombre de casos.
- La fase d'adaptació proposa la nova solució fent servir el cas recuperat més semblant (en el cas del SOMCBR) i del clúster recuperat (en el cas d'ULIC).
- La fase d'emmagatzematge no guarda nous casos.
- Cada resultat és el resultat d'un *10-fold stratified cross-validation*.
- Cada configuració és la mitja de 10 llavors per tal de compensar els efectes aleatoris de la construcció dels models.

5.4.3 Anàlisi i discussió dels resultats

La taula 5.3 resumeix els resultats d'executar el CBR fent servir (1) una memòria lineal, (2) una memòria organitzada amb SOM, i (3) una memòria organitzada amb X-means. Els paràmetres de rendiment estudiats són el percentatge d'encerts ($\%AR$) amb la seva desviació estàndard (σ), i el temps mig en milisegons que triga en executar la fase de recuperació sobre un P4-3Ghz amb 1 GRAM. S'ha fet servir el temps d'execució i no el nombre d'operacions realitzades perquè aquest és el paràmetre que dóna ULIC. A més a més, la taula remarca en negreta els millors resulats, i indica mitjançant els simbols $\uparrow$ i $\downarrow$ si el mètode de clusterització millora o no significativament els resultats respecte el model lineal al aplicar el *t-student* amb un nivell del 95% de confiança. D'altra banda, el símbol $\surd$ indica els casos on la proposta SOMCBR és estadísticament millor que la proposta d'ULIC.

Primer estudiarem els resultats referents al percentatge d'encerts. Els resultats de la taula mostren com tant l'estratègia d'organització del SOMCBR com la d'ULIC proporcionen percentatges d'encerts similars respecte als de l'organització lineal quan els *datasets* no presenten incertesa. En canvi, quan el *dataset* presenta incertesa, el rendiment oferit per ULIC en aquest aspecte es veu estadísticament reduït. Per tant, les capacitats *Soft-Computing* donen al SOMCBR un 'plus' gràcies al qual pot gestionar millor aquest tipus de coneixement. Aquest efecte lliga amb les reflexions que es van fer quan es van analitzar les estratègies de recuperació del SOMCBR mitjançant els mapes de complexitat. Això fa que SOMCBR proporcioni resultats significativament millors en SO, VE, MA, M3 i MB.

Taula 5.2: Descripció dels *datasets* utilitzats en l'avaluació de les plataformes SOMCBR i ULIC.

Codi	***Dataset***	**Atributs**	**Instàncies**	**Classes**	**Incertesa?**
BC	wisconsin	9	699	2	No
GL	glass	9	214	6	No
IO	ionosphere	34	351	2	No
IR	iris	4	150	3	No
SO	sonar	60	208	2	No
VE	vehicle	18	846	4	No
BI	biopsy	24	1027	2	Sí
MA	μCa	23	216	2	Sí
DD	ddsm	143	501	4	Sí
M3	mias-3c	153	320	3	Sí
MB	mias-bi	153	320	4	Sí

Taula 5.3: Mitja del percentatge d'encerts (%Encert), la desviació estàndard (σ), i el temps mig de recuperació d'un cas en milisegons sobre un CBR amb un model de memòria lineal, amb SOM i amb X-means.

Codi	**CBR Lineal**		**CBR amb SOM**			**CBR amb X-means**		
	%Encert (σ)	**Temps**	**%Encert (σ)**		**Temps**	**%Encert (σ)**		**Temps**
BC	96.14 (2.1)	1.8000	96.42 (2.6)		0.7000	**96.71 (1.9)**		1.0200
GL	69.16 (7.3)	0.6000	70.66 (7.8)		0.2100	**70.79 (8.7)**		0.5500
IO	**90.32 (4.2)**	0.3600	89.12 (4.8)		0.0800	90.31 (5.3)		0.0060
IR	96.32 (3.1)	0.3000	96.00 (3.2)		0.0150	**97.33 (3.2)**		0.0015
SO	**87.02 (6.9)**	0.3600	85.58 (7.2)	$\surd$	0.1400	82.93 (7.7)	$\downarrow$	0.1600
VE	69.05 (6.1)	0.4800	**69.15 (5.7)**	$\surd$	0.2200	65.60 (3.7)	$\downarrow$	0.0080
BI	**83.15 (3.5)**	0.7200	82.08 (3.7)		0.4300	81.40 (3.7)	$\downarrow$	0.3100
MA	62.50 (13.7)	0.1200	**68.06 (8.3)**	$\surd\uparrow$	0.0400	63.89 (9.8)		0.0900
DD	46.51 (5.4)	1.9800	**46.41 (4.1)**		1.2000	46.17 (5.2)		1.1000
M3	**70.81 (6.9)**	1.5000	69.57 (6.09)	$\surd$	0.7000	65.34 (6.2)	$\downarrow$	0.5400
MB	**70.31 (5.5)**	1.5000	**70.31 (5.4)**	$\surd$	0.7000	60.16 (9.2)	$\downarrow$	0.5400

D'altra banda, tant SOMCBR com ULIC milloren el temps mig en executar la fase de recuperació respecte el CBR amb organització lineal. No obstant, en aquest cas ULIC proporciona temps de resposta més ràpids fruït de l'estratègia de recuperació basada només en un nivell.

L'últim aspecte a comparar entre SOMCBR i ULIC és el nombre de clústers que generen. La taula 5.4 resumeix el nombre de clústers per cada problema i mètode. En els dos casos, el nombre ideal de clústers es calcula per aconseguir minimitzar l'error quadràtic. El sistema ULIC tendeix a construir més clústers que SOMCBR perquè ULIC es basa en definir patrons de comportament, mentre que SOMCBR es basa en indexar la informació segons la seva semblança per posteriorment cercar dins d'ell. Aquest comportament es modela amb les equacions 5.7, 5.8 i 5.9, les quals calculen el nombre de comparacions que realitza el CBR lineal, SOMCBR i ULIC. En funció del nombre de casos (I_{mc}), del nombre de clústers (M), i la distribució de les classes, el rendiment serà millor en SOMCBR o en ULIC.

$$time(Linear) = O(I_{mc}) \tag{5.7}$$

$$time(SOM) = O(M + \frac{I_{mc}}{M}) \tag{5.8}$$

$$time(SX - means) = O(M) \tag{5.9}$$

Taula 5.4: Resum del nombre de clústers de la memòria de casos per cada *dataset* i mètode. A més a més, pel cas del SOMCBR s'inclou la mida del mapa ($M \times M$), i en el cas d'ULIC el nombre de patrons per classe.

Codi	**Classes**	**Clústers en SOMCBR**	**Clústers en ULIC**
BC	2	30 (8×8)	42 (27-15)
GL	7	7 (6×6)	78 (20-15-10-0-20-3-10)
IO	2	44 (8×8)	30 (24-6)
IR	3	10 (6×6)	34 (20-4-10)
SO	2	37 (8×8)	52 (25-27)
VE	4	62 (10×10)	115 (25-20-35-35)
BI	2	4 (4×4)	44 (28-16)
MA	2	8 (16×16)	90 (50-40)
DD	4	3 (8×8)	10 (1-4-2-3)
M3	3	6 (10×10)	8 (2-3-3)
MB	4	6 (10×10)	8 (2-3-3)
NS1	-	3 (8×8)	3 (3)
NS2	-	8 (8×8)	8 (8)
NS3	-	8 (8×8)	8 (8)

Els resultats han mostrat com les propietats *Soft Computing* dels mapes de Kohonen permeten definir/trobar clústers robusts al soroll en situacions on és present coneixement imprecís i incert, ja que el percentatge d'encerts s'ha mantingut respecte el CBR lineal, a més de reduir dràsticament el temps computacional en recuperar un cas. En canvi, la proposta de l'ULIC es veu afectada per aquests dominis complexes ja que el seu percentatge d'encerts es redueix, tot i que els temps computacionals que s'obtenen són millors que els de la proposta SOMCBR.

Per tant, es pot concloure que l'estratègia per recuperar casos basada en dos nivells és una mecanisme més fiable per organitzar la informació que el fet de definir només patrons amb una classe associada.

5.5 Conclusions i línies futures

La finalitat d'aquest capítol ha estat presentar una metodologia que ajudi a l'expert a conèixer el rendiment de les diferents maneres de recuperar el casos segons la tipologia del problema. A partir d'això i dels seus requeriments, l'expert seleccionarà els factors pertinents fins assolir el nivell d'agressivitat desitjat. De l'estudi anterior es pot concloure:

- El mapa d'estratègies permet expressar d'una manera ràpida i fàcil d'entendre la taxonomia de les diferents maneres a través de les quals es poden recuperar els casos de la memòria clusteritzada.
- L'espai de geometries definit (A, B, C) representa bé les tres situacions en les quals SOM pot trobar-se.
- SOM és una tècnica de clustering que funciona millor davant de dominis complexes, ja que és aquí on les seves capacitats *Soft-Computing* i de *Knowledge Discovery* són realment utilitzades.
- L'*scatter plot* proposat per analitzar i comparar diferents estratègies sobre diferents *datasets* ha demostrat ser un mecanisme molt potent i fàcil d'entendre i construir.
- No té massa sentit fer clustering sobre *datasets* de geometries simples, ja que el sistema tendirà a fer clústers grans que no discerneixen patrons interns. El nivell de filtratge no tindrà la bondat desitjada.
- Les millors configuracions són aquelles que fan servir 3 clústers recuperant tots els casos, o bé, un percentatge d'aquestes. Aquestes configuracions resolen els problemes tan bé com si es fes servir tota l'expierència del sistema, però proporcionant reduccions dràstiques en el nombre de casos que es fan servir en mitjana a l'hora d'explorar la memòria de casos.
- L'usuari pot seleccionar estratègies agressives per problemes de complexitat mitjana-elevada, però per complexitats baixes ha de tendir a fer servir més casos per evitar perdre informació rellevant.

L'aplicació d'aquesta metodologia ens ha permet estudiar el comportament de les estratègies de recuperació de la memòria clusteritzada amb SOM. Per tant, la seva aplicació ha estat un èxit. Les contribucions d'aquest capítol es troben publicades als articles següents:

- A. Fornells, E. Golobardes, J.M. Martorell, J.M. Garrell, E. Bernadó i N. Macià. *A methodology for analyzing the case retrieval from a clustered case memor.* Al *7th International Conference on Case-Based Reasoning*, volum 4626 de LNAI, planes 122-136. Springer-Verlag, 2007. Nominat al millor article del congrés.

- A. Fornells, E. Golobardes, J.M. Martorell, J.M. Garrell, E. Bernadó i N. Macià. *Measuring the applicability of self-organizing maps in a case-based reasoning system.* Al *3rd Iberian Conference on Pattern Recognition and Image Analysis*, volum 4478 de LNCS, planes 532-539. Springer-Verlag, 2007.

- N. Macià, E. Bernadó, A. Fornells, E. Golobardes, J.M. Martorell i J. M. Garrell. *Revisión sobre métricas de complejidad en el modelado de clústers de un sistema CBR.* Al *V Taller nacional de minería de datos y aprendizaje*, 2007. En impremta.

- A. Fornells, E. Golobardes, D. Vernet i G. Corral. *Unsupervised case memory organization: Analysing computational time and soft computing capabilities.* Al *8th European Conference on Case-Based Reasoning*, volum 4106 de LNAI, planes 241-255. Springer-Verlag, 2006.

El treball futur s'emmarcaria en estudiar altres mètriques que s'ajustessin millor a SOM, o fins i tot, definir-ne de noves tenint en compte les peculiaritats de SOM. També seria interessant aplicar aquesta metodologia sobre altres estratègies de clustering.

Resum

La recuperació de casos d'una memòria organitzada amb clústers es diferencia d'una sense organització perquè permet fer una recuperació selectiva, on el nombre de casos explorats es redueix substancialment. Això permet millorar el rendiment del sistema en dues direccions. D'una banda, el temps d'exploració es redueix. D'altra banda, les capacitats resolutives es milloren perquè és més fàcil descartar casos incerts. El procés de recuperació es basa en dues etapes. Primer, es seleccionen els clústers que millor representen el cas d'entrada. Segon, s'exploren els clústers seleccionats per recuperar el conjunt de casos més similars. La dificultat en aquest procés està en definir el nombre de clústers a seleccionar i casos a recuperar, ja que no hi ha cap metodologia o principis que indiqui com fer-ho.

Aquest capítol ha presentat una metodologia per modelar el comportament de la recuperació dels casos d'una memòria clusteritzada a partir de dos factors. D'una banda, l'agressivitat desitjada per l'usuari, la qual es refereix a la reducció desitjada. D'altra banda, la complexitat de les dades, aspecte que fa referència a la capacitat que tenen els clústers per representar la geometria del problema. L'aplicació de la metodologia gira al voltant de tres elements: el mapa d'estratègies, l'*scatter plot* del rendiment i el mapa de complexitats. El mapa d'estratègies és una manera de representar les diferents maneres de recuperar la informació de la memòria de casos cluteritzada. L'*scatter plot* del rendiment és una proposta per comparar d'una manera ràpida i àgil un conjunt d'estratègies sobre un conjunt de *datasets*. El rendiment es mesura a partir de dues mesures: (1) quants cops va bé l'estratègia i (2) la reducció del nombre de casos emprats respecte fer servir tota la memòria de casos. Finalment, el mapa de complexitats és una eina per segmentar les dades segons la tipologia de la seva geometria.

L'avaluació de la metodologia s'ha fet a partir d'una proposta basada en organitzar la memòria mitjançant SOM. Els passos que s'han seguit han estat els següents: (1) generació de les configuracions a estudiar a partir del mapa d'estratègies; (2) segmentació dels *datasets* segons la seva complexitat mesurada amb el mapa de complexitats; (3) avaluació de les estratègies per cadascun dels tres conjunts de *datasets* mitjançant l'*scatter plot*. La conclusió de l'aplicació de la metodologia ha estat que el rendiment oferit per SOM és major a mesura que les dades són més complexes, ja que això permet potenciar al sistema amb el coneixement i relacions amagades dins de les dades. En canvi, la capacitat de SOM per segmentar les dades es veu minvat si els *datasets* són poc complexos. Aquest nivell de complexitat condiciona el nivell d'agressivitat amb el qual es con-

figura l'exploració de la memòria. En qualsevol dels casos, la proposta SOMCBR sempre millora dràsticament el nombre d'operacions requerides en explorar la memòria.

D'altra banda, el capítol també ha realitzat una comparativa entre SOMCBR i una altra plataforma que organitza la memòria de casos amb clústers. Aquesta plataforma, anomenada ULIC, només fa servir el nivell dels clústers ja que cada agrupació de casos té assignada la classe que representa. Tot i que aquesta propietat permet millorar el temps de resposta, els resultats han mostrat com aquesta reducció de la informació té un efecte negatiu sobre el percentatge d'encerts.

Capítol 6

Fase d'adaptació

La proposta de la solució al nou problema es realitza a partir de la informació obtinguda a la fase de recuperació. Concretament, i dins de l'àmbit dels problemes de classificació, aquesta proposta sovint es realitza a través d'esquemes de votació on la classe majoritària dels casos recuperats és directament la solució al nou problema. Aquest mateix procés pot traslladar-se directament al SOMCBR a partir dels K casos recuperats com a més similars del/s clúster/s seleccionat/s. En qualsevol cas, la qualitat de la solució dependrà directament de la fiabilitat d'aquests casos recuperats com a similars. Per tant, el seu grau d'incertesa pot afectar negativament a la capacitat resolutiva del sistema. Aquest capítol planteja com aprofitar la relació entre els casos i els patrons definits per SOM per establir graus de pertinença les classes modelades pel clúster i, d'aquesta manera, minimitzar l'impacte de la incertesa dels casos recuperats a l'hora de proposar la nova solució.

6.1 Motivació: fins a quin punt els veïns són de fiar?

La fase d'adaptació és l'encarregada de proposar una nova solució a partir del conjunt de casos recuperats com a més similars. El seu grau de complexitat és molt variable, i depèn principalment de la tipologia del problema i del tipus d'aplicació (Wilke et al., 1998; Wilke i Bergmann, 1998).

La construcció de la solució al nou problema pot realitzar-se principalment de dues maneres (Aamodt i Plaza, 1994). D'una banda, pot fer-se assignant un valor tipificat a un atribut, sovint anomenat 'classe', el qual representa un estat o una conclusió. Aquest enfocament, conegut com a *Copy Strategy*, s'aplica principalment a tasques de classificació com per exemple en el cas del diagnosi de càncer de mama. D'altra banda, la solució pot ser el resultat de la concatenació d'un conjunt de seqüències o premisses que defineixen les accions a realitzar per resoldre un problema. Aquest enfocament, conegut com a *Adapt Strategy*, s'aplica principalment a tasques de configuració, disseny o planificació com per exemple en definir com anar d'una ciutat a una altra a partir d'un conjunt de carreteres. En el nostre cas, i degut a la tipologia de les dades i dels problemes dels projectes amb els quals tractem, aquest capítol es centra en el primer enfocament.

L'assignació de la classe a partir dels K veïns més semblants (K-NN) al nou cas habitualment es basa en polítiques de votació, on la classe majoritària dels casos recuperats s'estableix com la més plausible (Cover i Hart, 1967; Dasarathy, 1991). Per exemple, si es recuperen els tres veïns $\{e_c, e_n, e_c\}$ on el subíndex representa la classe, la classe majoritària seria 'c'. Tanmateix, aquest esquema de decisió pot presentar moltes variants, com per exemple fer que els vots siguin ponderats segons la similitud del cas respecte els K casos recuperats. En qualsevol cas, és important destacar la rellevància del paper de K en la proposta de la solució perquè l'increment del seu valor permet minimitzar els efectes de seleccionar erròniament casos sorollosos. Això no vol dir que K hagi de

ser el més gran possible, ja que sinó pot introduir-se accidentalment soroll degut a fer servir casos que no són del tot semblants. Per tant, la definició de K és una arma de doble filament, la qual requereix establir valors de compromís (habitualment entre 3 i 5).

Tots aquests conceptes són extrapolables de manera directa al SOMCBR, de tal manera, que el procés de votació es realitza a partir dels K casos recuperats del/s clúster/s seleccionat/s com a més similar/s enlloc de considerar els K casos de tota la memòria. Sigui M_m el clúster m d'una memòria de casos clusteritzada en M clústers. Sigui $\vec{v_m}$ el vector director del clúster M_m. Sigui c_i el nou cas a classificar format per N atributs. Tal com va explicar-se al capítol anterior, el primer pas és determinar el clúster o el conjunt de clústers que millor representen el cas d'entrada. Si ens centrem en una estratègia agressiva, la selecció del clúster es fa a partir de trobar el clúster que minimitza la distància entre el seu vector director i el cas d'entrada tal com mostra l'equació 6.1. Un cop identificat el clúster, es seleccionen els K casos més similars a partir dels quals s'aplica l'esquema de votació esmentat anteriorment.

$$\forall M_m \in M : min \left(\frac{\sqrt{\sum_{n:0}^{N} v_m(n) - c_i(n)}}{N} \right) \tag{6.1}$$

Un dels principals beneficis del SOMCBR és que permet identificar patrons a les dades, aspecte que resulta molt útil per discernir entre comportaments diferents. Aquesta capacitat queda reflectida a l'exemple de la figura 6.1, la qual mostra la distribució a l'espai de les dades d'un problema de classificació en tres classes (A, B i C). Tot i que les capacitats de SOM defineixen tres zones amb gairebé tots els elements de la mateixa classe, les zones frontereres entre els clústers contenen casos amb un comportament ambigu. Aquesta incertesa generada pels elements pot afectar negativament al sistema fins al punt de confondre'l a l'hora de resoldre nous problemes. Aquest efecte es mostra clarament a la taula 6.1, la qual mostra el resultat d'aplicar diferents polítiques de votació (1-NN, 3-NN i 5-NN) sobre una memòria de casos sense i amb clústers. Com pot observar-se, l'èxit dels resultats està lligat a dos factors esmentats anteriorment. D'una banda, el fet d'incrementar K afavoreix a millorar la precisió en la classificació dels nous problemes X_A, X_B i X_C. D'altra banda, els clústers ajuden al sistema a discernir entre els patrons i, per tant, ajuden a rebutjar casos que pertanyen a altres patrons. No obstant, ambdós factors no són suficients.

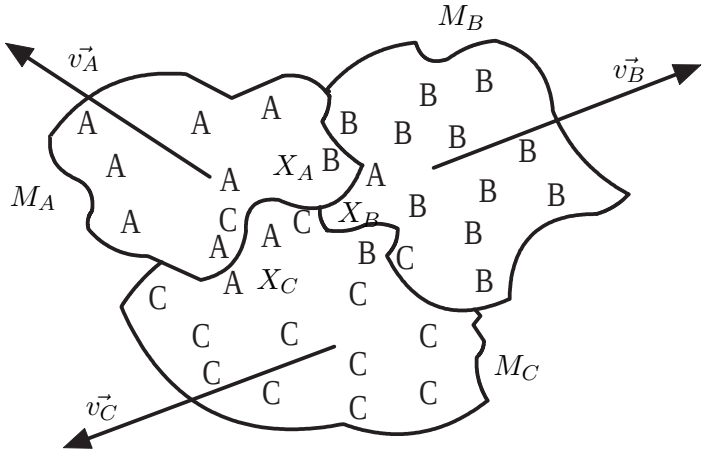

Figura 6.1: Distribució a l'espai de les dades d'un problema de classificació en tres classes (A, B i C), on l'aplicació de SOM permet identificar tres patrons de comportament (M_A, M_B i M_C). X_A, X_B i X_C representen els tres nous elements que s'han de classificar, on el subíndexs representen la classe real a la qual pertanyen.

Taula 6.1: Resultats d'aplicar diferents polítiques de votació sobre l'exemple de la figura 6.1. La part esquerra conté els resultats d'aplicar el K-NN sobre tota la memòria de casos, i la part dreta de fer-ho servir sobre el clúster més similar. Els elements ({...}) representen la classe dels elements retornats com a més semblants. Quan no és possible assignar un guanyador, es mostren els candidats.

Nou	**Tots els casos**			**Els casos del clúster més semblant**		
Cas	**1-NN**	**3-NN**	**5-NN**	**1-NN**	**3-NN**	**5-NN**
X_A	{B}: B	{B,B,A}: B	{B,C,A,A,B}: A/B	{B}: B	{B,A,A}: A	{B,A,A,A,C}: A
X_B	{A}: A	{A,B,C}: A	{A,B,C,C,B}: B/C	{A}: A	{A,B,C}: A	{A,B,C,B,B}: B
X_C	{A}: A	{A,A,A}: A	{A,A,A,C,C}: A	{A}: A	{A,A,C}: C	{A,A,C,C,B}: A/C

La principal mancança a l'hora d'abordar l'exemple anterior en els dos enfocaments ha estat el fet de considerar amb igual rellevància els K casos recuperats. Tot i que aquest efecte pot tenir-se en compte a partir d'introduir un pes que sigui proporcional a la similitud entre els K casos i el cas nou, això pot tornar-se en contra nostre si el cas que tenim més a prop és incert, ja que s'incrementaran els seus efectes negatius. Per tant, cal un mecanisme que mesuri més objectivament la robustesa dels K casos recuperats. És aquí on entra en joc el SOM.

Aquest capítol plateja un mètode basat en la definició de graus de pertinença dels K casos respecte el seu clúster per establir quina és la classe solució més plausible. El procés gestiona la incertesa dels casos de dues formes. D'una banda, SOM s'encarrega de separar els casos en patrons. D'altra banda, amb el grau de pertinença d'un cas a un clúster es podrà mesurar la seva robustesa. En el cas que es consideri que la proposta no és suficientment fiable, el sistema no classificarà l'exemple perquè es prefereix no classificar en cas de dubte. Aquest aspecte és molt important sobretot en entorns mèdics com el de diagnosi de càncer de mama, ja que el cost d'una equivocació és massa alt.

L'estructura del capítol és la següent. L'apartat 2 presenta l'esquema per millorar la fiabilitat de les solucions. L'apartat 3 avalua la proposta sobre un joc ampli de *datasets* de l'*Uci Repository* i del projecte HRIMAC (vegeu l'apèndix B). Finalment, l'apartat 4 conclou amb les conclusions i línies futures.

6.2 L'esquema de probabilitats

Una de les principals dificultats a l'hora de proposar una nova solució a partir d'un coneixement previ és conèixer fins a quin punt els casos del quals es disposa són robusts. Quan més robust sigui el coneixement, més fàcil serà descartar les possibles ambigüitats. Entenem que un cas és robust si es pot associar a un comportament amb una certa fiabilitat. En el cas del SOMCBR, aquesta robustesa es pot entendre com el grau de pertinença del cas al clúster que el conté.

Sigui c_i el nou cas a resoldre. Sigui $\vec{v_i}$ un vector que representa els N atributs de c_i. Sigui j una de les J possibles classes d'un problema. Sigui M_s el clúster més semblant al cas d'entrada. Sigui $\vec{v_s}$ el vector director de M_s. Sigui K el conjunt dels K-NN casos més semblants de X respecte el clúster M_s. Sigui K_j el conjunt dels casos de K que tenen la classe j. Sigui c_j un cas del conjunt K_j Sigui $d(\vec{v_1}, \vec{v_2})$ la distància Euclidiana entre dos vectors del mateix nombre d'elements.

Tenint en compte la definició de robustesa anterior, el grau de pertinença d'un cas a un clúster pot modelar-se quantitativament com l'invers de la distància normalitzada entre el cas i el vector director del model tal com mostra l'equació 6.2. D'aquesta manera, els elements més propers al vector director es consideren més fiables que els que estan prop de la frontera i, conseqüentment, han de considerar-se com a més rellevants perquè estan més correlacionats amb el clúster que modela el cas a resoldre. Valors propers a '1' reflecteixen alts nivells de pertinença del cas al clúster. En canvi, valors propers a '0' reflecteixen l'efecte invers.

$$robustesa(c_i, M_s) = 1 - d(\vec{v_i}, \vec{v_s}) \tag{6.2}$$

A partir del grau de robustesa del cas c_i respecte el clúster M_s es pot donar un pas més enllà encaminat a definir el grau de pertinença del nou cas a cadascuna de les J possibles classes modelades pels K casos recuperats tal com mostra l'equació 6.3.

$$\forall c_j \in K_j : G_j = \frac{\sum_{i:1,N} |1 - d(\vec{c_j}, \vec{v_s})|}{size(K_j)} \tag{6.3}$$

El valor G_j és alt si els casos recuperats associats a la classe j estan a prop del vector director. En cas contrari, el valor serà petit. Això permet introduir la robustesa dels casos dins del procés de decisió de la nova classe. Com a últim pas, els graus G_j es poden aprofitar per definir probabilitats de pertinença a cada classe tal com mostra l'equació 6.4, on $0 \leq P_j \leq 1$, i $\sum_{j:1..J} P_j = 1$. El valor més alt de P_j representa la classe més plausible. L'avantatge d'això és que d'una manera intuïtiva l'expert pot definir un nivell de confiança γ a partir del qual es considera fiable la solució ($P_j > \gamma$). En cas negatiu, el sistema no assigna cap classe perquè la informació de la qual es disposa no es pot considerar fiable.

$$P_j = \frac{G_j}{\sum_{j:1..J} G_j}(100\%) \tag{6.4}$$

Tot i que SOM és un mètode no supervisat d'aprenentatge, a la literatura hi ha treballs on s'aborda la possibilitat d'aprofitar SOM per construir distribucions de densitat a partir de les quals es generin probabilitats (Yin i Allinson, 2001; Verbeek et al., 2005). En el nostre cas l'enfocament és diferent, ja que la probabilitat de pertinença a una classe es defineix a partir dels K casos recuperats, i no sobre la distribució de les dades. A més a més, la distribució de les dades pot canviar a mesura que s'introdueix nou coneixement.

6.3 Avaluació de l'esquema de probabilitats per millorar la fiabilitat de les solucions

Aquest apartat avalua si la definició de l'esquema de probabilitat permet millorar la fiabilitat de les solucions proposades pels esquemes de votació. Entenem per fiabilitat a la capacitat que té el sistema de predir sense equivocar-se, és a dir, que prefereix no classificar a fer-ho erròniament. Primer es descriuen els experiments que es realitzaran i, a continuació, s'avaluen les dues propostes sobre un conjunt de datasets tenint en compte diferents paràmetres com la complexitat de les dades, el nombre de veïns i la mida dels mapes.

6.3.1 Experimentació

El rendiment dels dos enfocaments, tant el de votació i com el de probabilitats, està lligat al nombre de casos recuperats. Per aquest motiu, ambdós esquemes s'avaluen a partir dels valors més habituals de K (1, 3 i 5). Valors més grans que 5 es descarten perquè podrien introduir soroll al ser massa distants. Cal remarcar que l'esquema de probabilitats requereix d'almenys 2 casos i, conseqüentment, l'esquema 1-NN de probabilitat oferirà el mateix comportament que l'esquema 1-NN de votació.

A l'apartat anterior s'ha definit el valor γ com un valor a partir del qual es considera el mètode de probabilitats fiable. Si la probabilitat més alta és inferior a aquest valor, l'esquema no classifica ($P_j > \gamma$). Aquest mateix criteri pot definir-se de manera anàloga per l'esquema de votació, concretament, el sistema decideix no classificar si no hi cap cas que tingui una similitud més gran que aquest valor ($|1 - d(c_i, c_k| > \gamma)$). Des del punt de vista de l'esquema de classificació, la probabilitat mínima amb la qual es pot treballar depèn directament del nombre de classes (i.e. 0.5 amb dues classes i 0.33 amb tres classes). D'altra banda, a l'esquema de votació cal un valor suficientment alt per afirmar que dos elements són propers. Encara que els dos esquemes defineixin el mateix valor de γ, l'efecte sobre l'esquema de probabilitats és més restrictiu perquè totes les probabilitats han de sumar '1'. En canvi, al de votació això no succeeix perquè no intervé cap relació entre els casos a l'hora de validar el criteri de la distància. Això fa pensar que el valor de γ a l'esquema de votació hauria de ser lleugerament alt, sempre tenint en compte que és difícil trobar dos casos exactament iguals. Amb la finalitat d'estudiar la relació de γ en els dos enfocaments, el seu valor s'establirà a 0.7. Aquest valor pot considerar-se alt tant des del punt de vista de les probabilitats, com des del punt de vista de la mínima similitud per considerar dos casos com a similars.

D'altra banda, l'èxit dels dos mètodes també està molt vinculat a la capacitat dels clústers en representar les dades, la qual depèn principalment de la mida del mapa i de la complexitat dels datasets. Per aquest motiu, també s'estudia l'impacte dels dos mètodes sobre diferents mides de mapa i sobre *datasets* de diferents complexitats. Concretament, els *datasets* per avaluar els mètodes seran els estudiats al capítol anterior provinents de l'*UCI Repository* (Asuncion i Newman, 2007) i del projecte HRIMAC (vegeu l'apèndix B), els quals es troben novament resumits a la taula 6.2. La taula indica el nom, el nombre d'atributs i instàncies, així com el tipus de complexitat per cada *dataset* a partir del mapa de complexitat de la figura 5.6 presentat al capítol anterior.

Finalment, també s'inclou en l'experimentació els resultats d'executar un CBR amb una organització plana de la memòria de casos per disposar d'un altre punt de referència addicional. La resta de paràmetres de configuració són els següents:

- La funció de distància utilitzada tant per la construcció dels models, com per la comparació entre clústers i casos és la del complement de la funció Euclidiana (vegeu l'equació 5.1).
- S'avaluen diferents mides de mapa: 3×3, 4×4 i 5×5.
- Els models poden tenir diferent nombre de casos.
- La fase d'emmagatzematge no guarda nous casos.
- Cada resultat s'obté d'aplicar un *10-fold stratified cross-validation*.
- Cada configuració és la mitja de 10 llavors per tal de compensar els efectes aleatoris de la construcció dels models.

Alguns autors (Provost et al., 1998) han posat en dubte la utilitat del percentatge d'error a l'hora de mesurar la fiabilitat. Segons aquests, els paràmetres de la sensitivitat i l'especificitat són més adients, sobretot per dominis mèdics (Bradley, 1997). No obstant, això és cert sempre i quan hi hagi un cost associat a fer una classificació errònia com en el cas dels datasets μCa i *Biopsy*, sinó l'error és una mesura que mostra globalment millor ambdues magnituds. Per aquest motiu el següent apartat avalua de manera global tots els datasets anteriors, per després fer una anàlisi més concreta sobre els datasets específics del diagnosi de càncer de mama.

Taula 6.2: Descripció dels *datasets* utilitzats: nom, nombre d'atributs i d'instàncies, i tipus de complexitat. El sufix $2cX$ indica que el dataset classifica la classe X respecte la resta de classes.

Dataset	**Atributs**	**Instàncies**	**Tipus**	***Dataset***	**Atributs**	**Instàncies**	**Tipus**
segment2c2	19	2310	A	wav2c3	40	5000	B
iris2c2	4	150	A	wav2c1	40	5000	B
glass2c1	9	214	A	miasbi2c3	152	320	B
thy2c1	5	215	A	ddsm2c1	142	501	B
thy2c2	5	215	A	mias3c2c2	152	322	B
segment2c6	19	2310	A	thy2c3	5	215	B
segment2c7	19	2310	A	mias3c2c1	152	322	B
wine2c2	13	178	A	ddsm2c4	142	501	B
iris2c1	4	150	A	miasbi2c2	152	320	B
segment2c1	19	2310	A	wisconsin	9	699	B
wine2c1	13	178	A	wbcd	9	699	B
glass2c2	9	214	A	wav2c2	40	5000	B
miasbi2c4	152	320	A	sonar	60	208	B
glass2c4	9	214	A	wpbc	33	198	B
wine2c3	13	178	A	glass2c6	9	214	B
iris2c3	4	150	A	mias3c2c3	152	322	B
wdbc	30	569	A	biopsia	24	1027	B
segment2c3	19	2310	B	vehicle2c3	18	846	B
segment2c5	19	2310	B	vehicle2c2	18	846	B
glass2c3	9	214	B	bal2c3	4	625	C
vehicle2c1	18	846	B	bal2c2	4	625	C
segment2c4	19	2310	B	bal2c1	4	625	C
tao	2	1888	B	ddsm2c3	142	501	C
hepatitis	19	155	B	heartstatlog	13	270	C
glass2c5	9	214	B	μCa	21	216	C
ionosphere	34	351	B	ddsm2c2	142	501	C
vehicle2c4	18	846	B	pim	8	768	C
miasbi2c1	152	320	B	bpa	6	345	C

6.3.2 Anàlisi i discussió dels resultats

Les taules 6.3, 6.4 i 6.5 resumeixen les mitges dels percentatge d'error sobre els classificats (%Error) i el percentatge de casos no classificats (%No) amb les seves desviacions típiques (σ) corresponents per diferents valors de K, i sobre els esquemes de votació i probabilitat aplicats al SOMCBR per datasets de complexitat del tipus A[1], B[2] i C[3] respectivament. A més a més, s'ha afegit una columna addicional que fa referència al percentatge d'error respecte tots els classificats ($\%\text{Error}_{tots}$), la qual amplia proporcionalment el %Error respecte el 100% dels casos. D'aquesta manera, la comparació entre les estratègies que tenen un alt percentatge de no classificats pot fer-se qualitativament millor. D'altra banda, la taula també inclou les estadístiques del CBR aplicant diferents valors de K per l'esquema de votació.

A partir d'això, els resultats s'analitzen des de diferents perspectives i de manera transversal als tres tipus de complexitats. Primer, es fa una anàlisi horitzontal de les taules on s'avalua l'impacte de l'increment de K. A continuació, es realitza una anàlisi vertical per estudiar la relació de l'error i dels no classificats entre les diverses configuracions a les dues estratègies SOMCBR estudiades. Finalment, es conclou amb una reflexió sobre el paràmetre γ.

[1]Dominis poc complexos on SOM no pot explotar les seves capacitats.
[2]Dominis de complexitat mitjana on SOM comença a explotar les seves capacitats.
[3]Dominis de complexitat alta on SOM explota les seves capacitats.

Taula 6.3: Mitges del percentatge d'error sobre els classificats, del percentatge dels no classificats, i del percentatge d'error respecte tots els casos per diferents valors de K sobre els esquemes de votació (al CBR i al SOMCBR) i probabilitat (al SOMCBR) sobre els datasets de complexitat A. Els símbols '↑' i '↓' indiquen que l'estadística s'incrementa o es decrementa respecte l'estratègia 1-NN significativament a l'aplicar un t-test amb un 95% de confiança. En cas contrari es fa servir el símbol '-'.

Configuració		1-**NN**			3-**NN**			5-**NN**		
Sistema	**Mida**	**%Error** (σ)	**%No** (σ)	**%Error**$_{tots}$	**%Error** (σ)	**%No** (σ)	**%Error**$_{tots}$	**%Error** (σ)	**%No** (σ)	**%Error**$_{tots}$
CBR	-	2.8 (2.4)	0.0 (0.0)	2.8	2.6 (2.2) -	0.0 (0.0) -	2.6 -	2.7 (2.2) -	0.0 (0.0) -	2.7 -
SOMCBR-v	3×3	5.7 (0.1)	0.0 (0.0)	5.7	6.0 (0.1)↑	0.0 (0.0) -	6.0↑	7.1 (0.2)↑	0.0 (0.0) -	7.1↑
SOMCBR-v	4×4	6.5 (1.7)	3.9 (2.8)	6.8	6.8 (1.8) -	4.2 (2.1) -	7.2 -	7.1 (1.8) -	4.4 (2.3) -	7.5 -
SOMCBR-v	5×5	11.8 (1.5)	1.6 (1.2)	12.2	15.2 (1.8)↑	1.8 (1.1) -	15.6↑	18.0 (2.6)↑	2.1 (1.2) -	18.5↑
SOMCBR-p	3×3	5.7 (0.1)	0.0 (0.0)	5.7	2.6 (0.1)↓	10.3 (0.2)↑	3.1↓	1.6 (0.1)↓	19.7 (0.2)↑	2.4↓
SOMCBR-p	4×4	6.5 (1.7)	3.9 (2.8)	6.8	4.3 (1.7)↓	10.7 (3.0)↑	4.9↓	3.7 (1.6)↓	15.0 (3.1)↑	4.6↓
SOMCBR-p	5×5	11.8 (1.5)	1.6 (1.2)	12.2	7.4 (1.3)↓	21.9 (2.9)↑	10.9↓	6.2 (1.2)↓	34.0 (3.5)↑	12.3 -

Taula 6.4: Mitges del percentatge d'error sobre els classificats, del percentatge dels no classificats, i del percentatge d'error respecte tots els casos per diferents valors de K sobre els esquemes de votació (al CBR i al SOMCBR) i probabilitat (al SOMCBR) sobre els datasets de complexitat B. Els símbols '↑' i '↓' indiquen que l'estadística s'incrementa o es decrementa respecte l'estratègia 1-NN significativament a l'aplicar un t-test amb un 95% de confiança. En cas contrari es fa servir el símbol '-'.

Configuració		1-**NN**			3-**NN**			5-**NN**		
Sistema	**Mida**	**%Error** (σ)	**%No** (σ)	**%Error**$_{tots}$	**%Error** (σ)	**%No** (σ)	**%Error**$_{tots}$	**%Error** (σ)	**%No** (σ)	**%Error**$_{tots}$
CBR	-	16.0 (3.8)	0.0 (0)	16.0	14.6 (3.5) -	0.0 (0.0) -	14.6 -	14.3 (3.8) -	0.0 (0.0) -	14.3 -
SOMCBR-v	3×3	19.9 (0.1)	0.0 (0.0)	19.9	20.0 (0.1) -	0.0 (0.0) -	20.0 -	19.7 (0.1) -	0.0 (0.0) -	19.7 -
SOMCBR-v	4×4	21.0 (1.3)	3.3 (1.1)	21.4	20.7 (1.4) -	3.6 (1.2) -	21.1 -	20.5 (1.5) -	3.7 (1.1) -	21.2 -
SOMCBR-v	5×5	26.2 (1.2)	0.5 (0.3)	26.3	25.2 (1.3) -	1.2 (0.4) -	25.3 -	24.3 (1.3) -	1.1 (0.7) -	24.4 -
SOMCBR-p	3×3	19.9 (0.5)	0.0 (0.0)	19.9	5.2 (1.0)↓	35.7 (0.1)↑	10.3↓	2.7 (0.8)↓	50.1 (0.1)↑	8.0↓
SOMCBR-p	4×4	21.0 (1.3)	3.3 (1.1)	21.4	7.1 (1.2)↓	36.5 (2.2)↑	13.1↓	4.2 (1.2)↓	49.5 (2.4)↑	10.7↓
SOMCBR-p	5×5	24.3 (1.2)	0.5 (0.3)	24.4	7.7 (1.0)↓	41.3 (2.0)↑	15.3↓	4.4 (0.8)↓	57.9 (2.0)↑	12.9↓

Taula 6.5: Mitges del percentatge d'error sobre els classificats, del percentatge dels no classificats, i del percentatge d'error respecte tots els casos per diferents valors de K sobre els esquemes de votació (al CBR i al SOMCBR) i probabilitat (al SOMCBR) sobre els datasets de complexitat C. Els símbols '↑' i '↓' indiquen que l'estadística s'incrementa o es decrementa respecte l'estratègia 1-NN significativament a l'aplicar un t-test amb un 95% de confiança. En cas contrari es fa servir el símbol '-'.

Configuració		1-**NN**			3-**NN**			5-**NN**		
Sistema	**Mida**	**%Error** (σ)	**%No** (σ)	**%Error**$_{tots}$	**%Error** (σ)	**%No** (σ)	**%Error**$_{tots}$	**%Error** (σ)	**%No** (σ)	**%Error**$_{tots}$
CBR	-	26.4 (5.8)	0.0 (0.0)	26.4	24.3 (5.2) -	0.0 (0.0) -	24.3 -	24.2 (4.6) -	0.0 (0.0) -	24.2 -
SOMCBR-v	3×3	31.2 (0.4)	0.0 (0.0)	31.2	27.8 (0.3)↓	0.0 (0.0) -	27.8↓	27.0 (0.3)↓	0.0 (0.0) -	27.0↓
SOMCBR-v	4×4	27.7 (1.3)	4.3 (1.8)	28.6	26.7 (0.9) -	4.6 (2.1) -	27.7↓	25.3 (1.2)↓	4.8 (1.8) -	28.2 -
SOMCBR-v	5×5	33.0 (1.1)	0.5 (0.2)	33.1	32.3 (0.9) -	0.7 (0.2) -	32.5 -	32.7 (0.9) -	0.8 (0.4) -	32.9 -
SOMCBR-p	3×3	31.2 (0.4)	0.0 (0.0)	31.2	8.7 (0.2)↓	47.0 (0.6)↑	19.7↓	3.3 (0.1)↓	64.1 (0.4)↑	13.9↓
SOMCBR-p	4×4	27.6 (1.3)	4.3 (1.8)	28.6	10.4 (0.9)↑↓	44.8 (1.9)↑	20.6↓	6.1 (0.9)↓	61.6 (2.2)↑	18.8↓
SOMCBR-p	5×5	32.9 (1.1)	0.5 (0.2)	33.1	11.3 (1.0)↑↓	48.2 (2.9)↑	24.7↓	6.2 (0.8)↓	66.5 (1.9)↑	22.7↓

Tot i que l'increment de K hauria sempre de guiar al sistema cap a resultats més fiables, això no succeeix sempre a l'esquema SOMCBR de votació. Aquest efecte es veu clarament als *datasets* de complexitat de tipus A, on els resultats 3-NN i 5-NN no ofereixen resultats significativament millors si s'aplica un t-test amb un nivell de confiança del 95%. En canvi, aquest efecte s'inverteix a mesura que la complexitat augmenta tal com mostren les taules dels *datasets* de complexitat de tipus B i de tipus C. El motiu d'això està directament relacionat amb el que va estudiar-se al capítol anterior. A mesura que la complexitat augmenta, la potència de SOM comença a aparèixer degut a que pot aprendre del coneixement ocult entre les dades per construir patrons més fiables. Això fa que la situació s'estabilitizi als de complexitat de tipus B, i que els resultats comencin a ser significativament millors en els de complexitat de tipus C. Pel que fa a l'esquema de probabilitats, pot observar-se com l'increment de K sí que implica una reducció considerable de l'error a l'aplicar un t-test amb un nivell de confiança del 95%, tant pel que fa referència a l'error sobre els casos classificats, com l'error extrapolat al 100% dels casos. Al mateix temps i, com a conseqüència de la millora de la fiabilitat, es produeix un increment del percentatge de casos que el sistema no classifica perquè no pot garantir el grau de fiabilitat. Per tant, l'increment del nombre de veïns millora considerablement la fiabilitat del sistema en l'esquema de probabilitats respecte el de votació, però un valor massa alt pot arribar a crear incertesa tal com va comentar-se als inicis d'aquest capítol.

Un cop hem vist l'impacte del valor de K en els dos esquemes, passarem a analitzar l'impacte de la mida del mapa sobre les configuracions. Tal com va explicar-se al capítol 3, la mida del mapa es defineix de manera automàtica a l'entrenament: a partir de diferents mides construïdes, es selecciona aquella que tingui el sumatori de l'error mínim entre cada clúster i els seus casos. En aquest cas d'estudi s'ha forçat l'execució del sistema per diferents mides, i els resultats que s'han obtingut en totes les configuracions han estat els mateixos: l'increment de la mida del mapa implica un augment del error. Aquest efecte es produeix perquè el nombre de casos per clústers es redueix a l'incrementar el nombre de clústers i, conseqüentment, alguns casos poden ser ignorats si només es selecciona un clúster com en aquest cas. D'altra banda, l'efecte s'accentua lleugerament als *datasets* de complexitat de tipus A, on els clústers són menys precisos. Tot això lliga amb els resultats del capítol anterior, en el sentit que si els clústers tenen pocs casos (perquè hi ha molts clústers), és millor seleccionar més d'un clúster per evitar perdre coneixement que pot arribar a ser potencialment útil. Per tant, la mida del mapa afecta d'igual forma als dos esquemes, essent l'estratègia de selecció i la tipologia de les dades dos factors que afectaran més o menys al rendiment.

Finalment, analitzarem els percentatges d'error entre els esquemes de votació i els de probabilitats des d'un punt de vista global tenint en compte els escenaris anteriors. Respecte la configuració basada en CBR, l'esquema de probabilitats no ofereix una millora significativa de l'error (%Error) en el cas dels *datasets* de complexitat de tipus A. En canvi, en els *datasets* de complexitat de tipus B i C hi ha una reducció contundent de l'error al voltant del 9%-16% segons la configuració. Aquests mateixos efectes es fan patents també si es compara l'error respecte el 100% dels casos ($\%\text{Error}_{tots}$), però baixant lleugerament la millora 6%-11%. Tot i aquests bons resultats '*no és or tot el que rellueix*', és a dir, aquesta reducció implica haver de deixar un percentatge considerable de casos sense classificar (entre un 10% i un 60% segons la complexitat i la configuració).

La figura 6.2 mostra la correlació entre el percentatge d'error sobre els classificats i el percentatge de casos no classificats per les configuracions de l'esquema de probabilitats, tenint en compte les mides i complexitat dels *datasets*. Tot i que conceptualment la configuració 1-NN no té massa sentit perquè només intervé un cas en el càlcul de la probabilitat (sempre és 100%), els resultats s'han inclòs per tenir una referència més a l'hora de mostrar la relació entre els dos paràmetres. Les gràfiques reflecteixen els comentaris que s'han anat fent als paràgrafs anteriors respecte l'impacte del valor de K i la mida dels mapes segons la complexitat dels datasets: a major K i menor mida de mapa s'obtenen menys errors, però el nombre de casos no classificats augmenta.

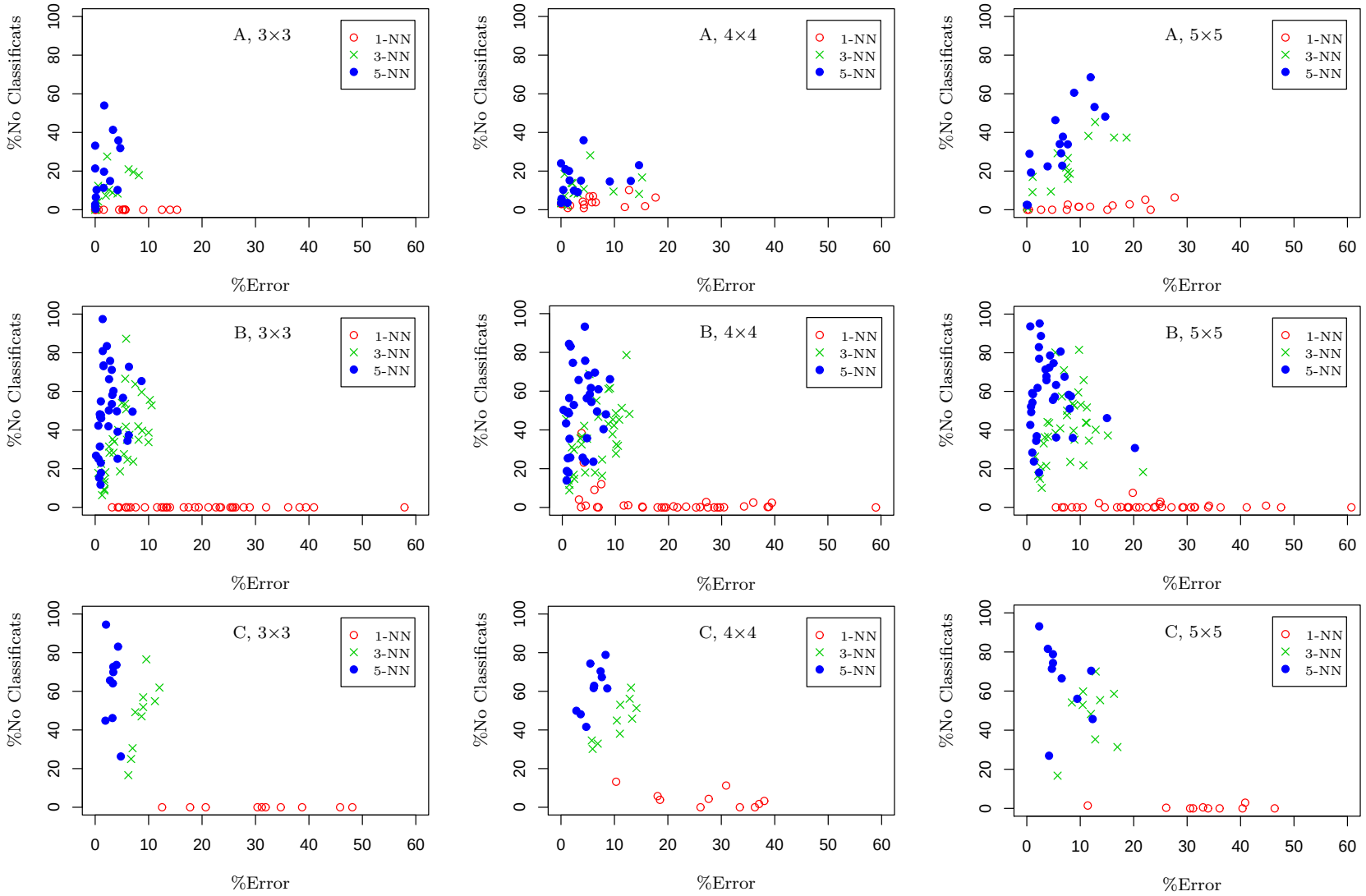

Figura 6.2: Les gràfiques mostren la relació entre el percentatge d'error sobre els classificats respecte el percentatge de casos no classificats de la configuració SOMCBR-p pels datasets estudiats tenint en compte la seva complexitat i diferentes mides de mapa.

Malgrat això, aquest 'preu' vers la capacitat de resolució és raonable en entorns tan crítics com els mèdics, on el preu d'equivocar-se és una vida humana. Si el nombre de no classificats es vol reduir, la solució passa per reduir el valor de γ. En el cas de l'esquema de votació, caldria augmentar el valor de σ per fer que els casos que intervinguin siguin el més semblant possibles a l'actual, tot i que això no ens garanteix que no pertanyin a un patró ambigu.

Per tant, com a reflexió final l'esquema de probabilitat millora dràsticament la fiabilitat respecte l'esquema de votació tenint en compte els resultats i els comentaris anteriors.

6.4 Conclusions i línies futures

La fiabilitat és un dels principals requisits de qualsevol sistema. No obstant, millorar la reducció de l'error sovint implica haver de reduir la capacitat resolutiva, en el sentit de deixar de classificar un conjunt de casos per considerar-los massa ambigus.

Aquest és precisament el que succeeix amb la proposta de l'esquema de probabilitats. A partir dels K casos recuperats com a més similars respecte el cas d'entrada, es calcula la probabilitat de pertànyer a cadascuna de les classes dels casos tenint en compte el grau de pertinença dels casos al clúster. Amb aquest criteri es considera que els casos que estan a prop de la part central del clúster són més robusts, i els que són més distants es consideren més ambigus, és a dir, que potser alguna part d'ells és característica d'altres patrons. La gran diferència d'aquest enfocament respecte d'altres és que s'intenta objectivitzar la detecció de casos ambigus a partir de les relacions que estableix SOM.

Per tal d'avaluar l'impacte sobre la fiabilitat d'aquesta proposta s'ha comparat aquest esquema respecte un esquema de votació, on la classe es calcula a partir de la classe majoritària dels K casos recuperats. L'anàlisi de la proposta ha girat entorn a diferents paràmetres: la complexitat de les dades, la mida del mapa i el nombre d'elements utilitzats per a proposar la solució. A partir de l'estudi sobre un joc ampli de *datasets* de tres tipus de complexitat, les conclusions a les que s'han arribat han estat les següents:

- L'increment de K permet millorar la fiabilitat dels resultats en ambdós esquemes. El fet de disposar de més informació permet minimitzar l'impacte d'un cas incert. Per contra, un increment excessiu pot fer que el sistema tendeixi a deixar masses casos sense classificar degut a l'ambigüitat d'introduir casos que siguin potser massa distants.

- La mida del mapa afecta de manera similar als dos esquemes. Si la memòria es clusteritza en masses casos, potser hi ha informació que no es té en compte a l'hora d'aplicar la fase de recuperació. Per tant, segons la relació nombre de casos i nombre de clústers, la fase de recuperació ha de considerar seleccionar més d'un clúster.

- La complexitat de les dades afecta directament al rendiment dels esquemes. A mesura que afecta la complexitat, l'aproximació SOMCBR és capaç d'oferir una millora de rendiment respecte fer servir un CBR amb una memòria organitzada de manera lineal.

- L'efecte del valor de γ per gestionar la fiabilitat és molt més important a l'esquema de probabilitats, ja que hi ha una relació directe entre totes les probabilitats (la seva suma ha de fer '1'). En canvi, a l'esquema de votació això no ha de perquè ser així.

Per tant, tenint en compte tot això pot afirmar-se que l'esquema de probabilitats és capaç d'oferir un percentatge d'error més reduït que l'esquema de votació, malgrat que el nombre de classificats es veurà sacrificat ja que es prefereix no errar a l'hora de classificar. La contribució d'aquest capítol, aplicada al context del diagnòstic de càncer de mama, es troba publicada a l'article següent:

- A. Fornells, E. Golobardes, J.M. Martorell, J.M. Garrell i X. Vilasís. *Management of relations between cases and patterns from SOM for helping experts in breast cancer diagnosis.* A *International Journal of Neural Systems*, 2007. En impremta.

Les línies futures d'aquest treball es poden dividir en dues parts. D'una banda, seria interessant estudiar en profunditat els criteris que haurien de donar-se per definir el valor llindar σ segons el nombre de classes i la complexitat del *dataset*. D'altra banda, seria interessant extrapol·lar l'esquema de probabilitats a més d'un clúster.

Resum

El capítol ha presentat una proposta per millorar la fiabilitat de les solucions proposades pel SOMCBR. La nova proposta es basa en definir probabilitats de pertinença a les classes dels casos retornats com els més similars. La probabilitat es defineix a partir de la robustesa del cas al clúster, és a dir, a la distància que té respecte el seu director vector. D'aquesta manera, els casos que són més afins al clúster tenen un pes més important en la decisió. El principal avantatge d'aquest enfocament respecte d'altres, és que el criteri de robustesa es fa des del punt de vista de la similitud de patrons entre els clústers.

Els resultats han mostrat que l'esquema de probabilitats permet reduir notablement l'error de classificació, tot i que això afecta a la capacitat del sistema per classificar: si no s'acompleix un cert nivell de confiança, el sistema no classifica. Lligant amb el capítol anterior, els efectes de l'esquema de probabilitats s'accentuen més a mesura que la complexitat dels datasets augmenta.

Capítol 7

Fase de revisió

La gran dificultat amb la qual ha d'enfrontar-se l'expert a la fase de revisió és que no pot afirmar categòricament que el resultat proposat pel sistema sigui del cert correcte degut a la complexitat, incertesa i coneixement parcial que es té en la majoria dels problemes reals. Aquest capítol presenta la caracterització dels clústers generats per SOM com a un mecanisme addicional per ajudar a l'expert a la resolució d'aquesta tasca. Per fer-ho es planteja la substitució del primer nivell de l'organització de la memòria de casos, composat pels vectors directors, per explicacions generades a partir de la generalització dels casos mitjançant el concepte d'antiunficació. Això permet descriure els clústers fent servir el mateix llenguatge que els casos, possibilitant a l'expert entendre millor perquè els casos s'han agrupat, i perquè un clúster és seleccionat.

7.1 Motivació: entendre el perquè de les coses

L'expert apareix a la fase de revisió amb la finalitat de validar la feina realitzada pel sistema a les fases de recuperació i adaptació. A més a més, el resultat de la seva validació serà clau per decidir si cal o no guardar el coneixement nou a la memòria de casos. Per tant, la seva responsabilitat és elevada en aquesta fase.

Les maneres més habituals de donar suport a l'expert a l'hora de validar els resultats en el CBR es realitzen tenint en compte les similituds respecte el cas original (McSherry, 2005), les seves diferències (McCarthy et al., 2004), o bé, combinant ambdós punts de vista (Doyle et al., 2003). No obstant, aquestes informacions sovint són massa matemàtiques per l'expert, aspecte que pot dificultar la seva comprensió.

Pel que fa al SOMCBR, la seva organització amb clústers aporta a l'expert informació addicional respecte els sistemes CBR que no la tenen. D'una banda, permet que l'expert conegui d'una manera ràpida i visual quins casos són semblants entre ells encara que siguin de classes diferents. D'altra banda, aquest meta-nivell definit pels vectors directors ajuda a conèixer perquè un conjunt de casos són recuperats. Tot i això, el meta-nivell dels vectors directors té la mateixa mancança que les aproximacions anteriorment citades: són explicacions matemàtiques de massa baix nivell.

La finalitat d'aquest capítol és aprofitar el coneixement descobert per SOM per caracteritzar cada clúster mitjançant explicacions simbòliques que descriguin les agrupacions dels casos d'una manera més comprensible que els vectors directors. El tret diferencial és que les explicacions fan servir el mateix llenguatge de representació que els casos i, conseqüentment, la seva comprensió és més fàcil per l'expert (vegeu la figura 7.1). A més a més, això permet aprofitar millor els avantatges del meta-nivell comentat anteriorment.

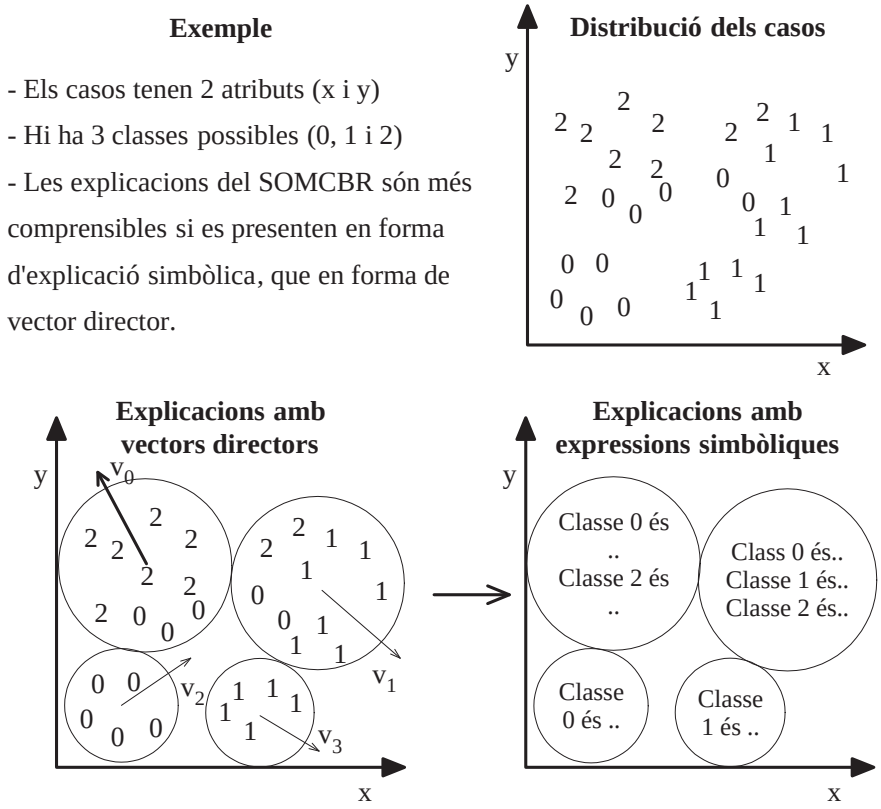

Figura 7.1: La capacitat per definir un nivell de jerarquia addicional a la memòria de casos permet potenciar el nivell de comprensió del resultat. Aquest aspecte és més important encara si es presenta en forma d'explicació.

Aquesta proposta, desenvolupada conjuntament amb la Dra. Eva Armengol[1], es basa en generalitzar el contingut dels clústers utilitzant el concepte d'antiunificació desenvolupat per ella mateixa i l'Enric Plaza (Armengol i Plaza, 2000). En un dels seus treballs més recents, proposa un enfocament basat en similituds on opcionalment l'usuari pot contribuir a detectar les diferències entre classes (Armengol i Plaza, 2006). En el nostre context de clustering, aquest enfocament pot ajudar a explicar les similituds entre classes a partir dels casos que estan al mateix clúster, així com també permetre comparar les diferències entre casos que, tot i tenir la mateixa classe, estan en clústers diferents.

La caracterització de clústers mitjançant generalitzacions no és un enfocament nou. Recentment Zenko (Zenko et al., 2005) va fer servir l'algorisme CN2 per induir regles que determinessin prototips pels clústers mitjançant vectors que representen freqüències. Lechevalier (Lechevallier et al., 2006) va fer servir una combinació de SOM i un algorisme de clústering dinàmic anomenat SCLUST per generar explicacions amb dades simbòliques. Tornant altre cop al camp del CBR, Malek and Amy (Malek i Amy, 2007) van realitzar una proposta supervisada basada en una organització dels casos en dos nivells per definir prototips sobre grups de casos. No obstant, la nostra proposta té dos trets diferencials com més endavant es veurà. D'una banda, les explicacions generades no són discriminants, és a dir, no identifiquen de manera unívoca un clúster. D'altra banda, les explicacions són el resultat d'un procés semi-supervisat on (1) les dades són agrupades sense tenir en compte la classe a partir de SOM, i (2) es generen tantes explicacions per clúster com classes diferents tinguin els casos del clúster.

[1]La Dra. Eva Armengol és Científica Titular del *Consejo Superior de Investigaciones Científicas* (CSIC), la qual actualmente treballa a l'Institut d'Investigació d'Intel·ligència Artificial (IIIA)

L'estructura d'aquest capítol es divideix en els punts següents. El capítol 2 introdueix el mecanisme per construir generalitzacions sobre les dades supervisades i no supervisades, així com la manera d'interpretar-les dins el context dels clústers i casos tant per part de l'expert com del sistema. L'apartat 3 descriu l'experimentació realitzada per avaluar la precisió de les explicacions per representar les dades des dos punts de vista, un qualitatiu i un altre quantitatiu. L'apartat 4 finalitza amb les conclusions i línies futures.

7.2 Mineria de clústers a través de descripcions simbòliques

SOMCBR justifica l'agrupació de casos a partir del vector director. Malgrat això, la seva representació no és fàcil de comprendre pels experts degut al seu nivell matemàtic.

Aquest apartat presenta com aplicar el concepte de generalització a partir de l'operador d'antiunificació per generar explicacions que millorin el grau de comprensió del meta-nivell definit pels vectors directors generats per SOM.

Primer es descriu com aplicar l'operador i, tot seguit, es detalla com generar aquestes explicacions per aprofitar les seves avantatges en els clústers.

7.2.1 Descripció simbòlica d'un clúster amb dades supervisades

A (Armengol i Plaza, 2006) va introduir-se un esquema d'explicacions simbòliques per justificar la recuperació d'un conjunt de casos. En el nostre cas, es vol seguir la mateixa línia però aplicat a explicar el perquè un conjunt de casos s'han agrupat. Aquest procés es realitza a través d'una generalització dels casos del clúster a partir del concepte d'antiunificació introduït a (Armengol i Plaza, 2000), però amb algunes diferències. L'antiunificació de dos objectes es defineix com la generalització més específica d'ells dos, on la descripció conté els atributs compartits d'ambdós objectes amb el seu valor més específic. Aquí s'aplica la mateixa idea però tenint en compte només els atributs comuns entre els dos objectes.

Sigui M_m un clúster, i $C_1, ..., C_n$ el conjunt de casos que hi pertanyen després d'aplicar SOM a la memòria de casos. Cada cas c_j està descrit per un conjunt d'atributs $\mathcal{A}$, els quals poden prendre valors continus (reals), enters, o simbòlics pertanyents a un conjunt $\mathcal{V}$. L'explicació D_m del perquè un conjunt de casos s'han agrupat en un mateix clúster M_m es construeix de la manera següent:

- L'explicació D_m conté tots els atributs que siguin comuns a tots els casos de M_m. Els atributs que tinguin valors desconeguts són descartats.
- Per a cada atribut seleccionat, es guarden tots els valors que apareguin als casos a l'explicació.
- Sigui a_k un atribut comú a tots els casos de M_m tal que a_k pot adoptar qualsevol dels valors simbòlics que pertanyen a $\mathcal{V}_k$. L'atribut es descarta quan la unió de tots els valors de a_k dels casos de M_m és el mateix que $\mathcal{V}_k$. Això vol dir que l'atribut és irrellevant per descriure M_m.

L'explicació resultant és una conjunció de premisses, on cada premissa representa el conjunt de valors que pot tenir un atribut. A continuació s'il·lustra el procediment amb un exemple. Sigui M_m un clúster composat per quatre casos tal com mostra la part superior de la figura 7.2(a). Seguint els passos, els atributs *steroid, spleen-palpable, spiders, fatigue, malaise, liver-big, protime*, i *ascites* es descarten perquè no apareixen a tots els casos (i.e., *steroid* no està a *obj-137*). D'altra banda, els atributs *sex, antiviral* i *histology* també es descarten perquè prenen tots els valors possibles (i.e., *male* en *obj-136* i *female* en *obj-137*). Per tant, D_m permet representar la generalització dels casos del clúster M_m. L'explicació generada a la figura 7.2(a) s'entendria com que el clúster conté casos on l'atribut *age* pot valer 33, 31, 78 o 34, l'atribut *varices* val no, i així per la resta.

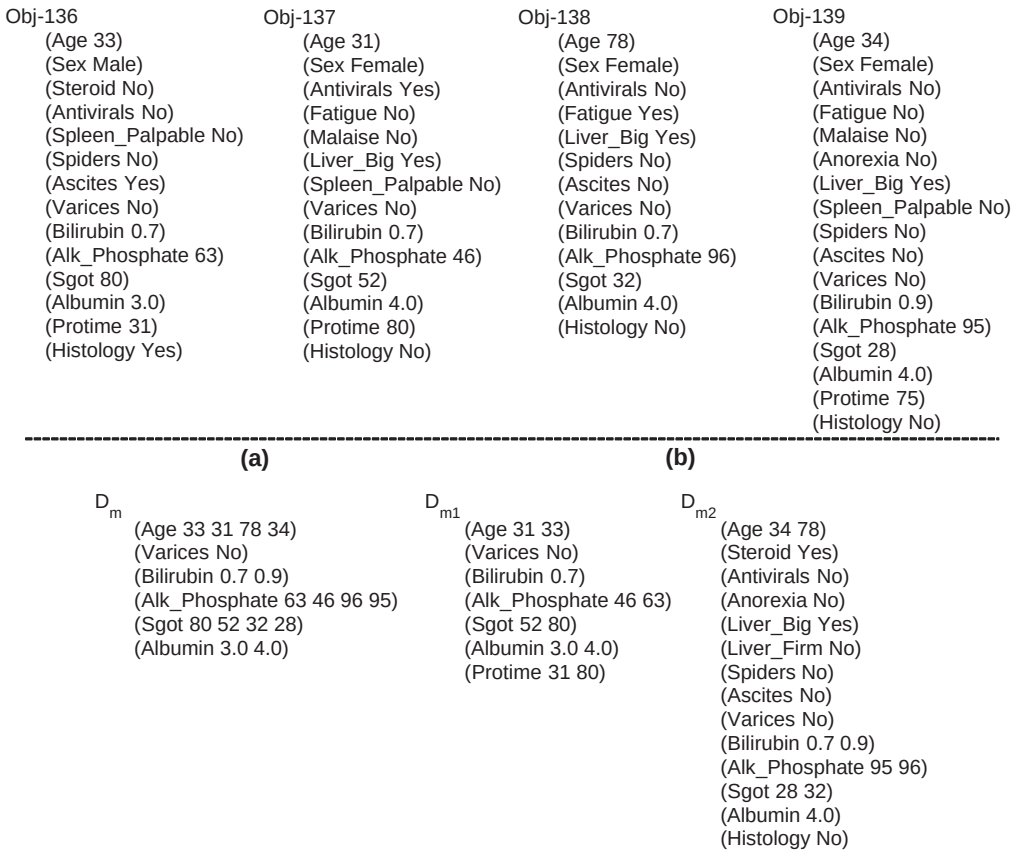

Figura 7.2: La part inferior esquerra mostra la generalització realitzada sobre els quatre casos del clúster M_m de la part superior. D'altra banda, la part inferior dreta mostra el mateix procés però aplicat de manera independent sobre els casos que tenen la mateixa classe. D_{m1} representa la classe 1 formada pels elements obj-136 i obj-137. D_{m2} representa la classe 2 formada pels elements obj-138 i obj-139.

Els clústers sovint contenen casos que pertanyen a diferents classes. El procediment anterior pot aplicar-se en aquest context per aconseguir disposar de J explicacions, on J representa el nombre de classes presents en el clúster, sobre perquè un conjunt de casos de la mateixa classe s'han agrupat. El procediment a seguir és idèntic a l'anterior amb la diferència que ara el procés s'aplica de manera independent sobre el conjunt de casos que tenen la mateixa classe. D'aquesta manera, un clúster M_m està descrit per una disjunció de descriptors $D_{m,j}$ que satisfan la descripció global D_m.

Suposem que els elements *obj-136* i *obj-137* de la figura 7.2 pertanyen a la classe 1, i els elements *obj-138* i *obj-139* a la classe 2. En aquest cas, el clúster M_m es descriu com una disjunció de dues explicacions D_{m1} i D_{m2} tal com mostra la part inferior de la figura.

7.2.2 Descripció simbòlica d'un clúster amb dades no supervisades

L'apartat anterior ha abordat la generalització dels casos d'un clúster per cadascuna de les diferents classes presents. No obstant, pot succeir que la memòria de casos disposi de casos que no tenen classe. Això per exemple pot succeir en dominis on l'objectiu no és classificar sinó fer una planificació o una configuració a partir d'un conjunt de restriccions i, per tant, la informació de la classe no és clau. En aquest context, el procediment anterior pot aplicar-se igual amb la diferència que ara es crearia una classe 'virtual' que donaria cabuda a tots els casos que no tenen classe.

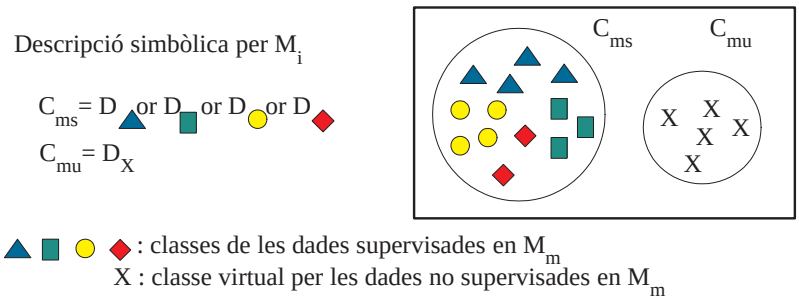

Figura 7.3: M_m és un clúster amb dades supervisades (C_{ms}) i no supervisades (C_{mu}). Els cinc símbols representen les quatre classes i la classe 'virtual'.

Sigui M_m un clúster, i C_m el conjunt de casos que hi pertanyen tal com mostra la figura 7.3. A partir d'això es pot definir C_{ms} com el conjunt de casos que tenen classe coneguda, i C_{mu} els casos que no tenen ($C_m = C_{ms} \cup C_{mu}$).

7.2.3 Interpretacions de les descripcions simbòliques

La introducció de les descripcions simbòliques permet representar les relacions que vinculen els casos amb un llenguatge de més alt nivell que l'emprat amb els vectors directors. Aquesta nova interpretació dels clústers permet als experts facilitar la tasca d'entendre el significat del clúster. A part d'aquest benefici, aquesta mateixa capacitat pot aprofitar-se també per dotar al sistema d'una altra manera d'interpretar com accedir als casos.

L'accés a la memòria de casos va definir-se al capítol 5 com un procediment de dues etapes basat en seleccionar els clústers que modelaven el cas d'entrada, per després recuperar els casos dels clústers seleccionats. En ambdós passos el procés es basa en l'aplicació d'una mètrica de similitud per avaluar la validesa dels clústers i els casos. La qüestió a abordar és la següent: com poden les explicacions aprofitar-se per validar això?

Les explicacions són una especificació de les relacions propietat-valor que tenen un conjunt de casos dins un clúster. Per tant, la condició perquè un cas pertanyi a un clúster o que sigui semblant a un cas resolt només implica que el cas en qüestió disposi de les mateixes relacions atribut-valor que el clúster i que el cas resolt. En el cas dels atributs numèrics, i en especial els continus, pot arribar a ser impossible que dos valors siguin exactament iguals. Per aquest motiu, es fa necessari definir un nivell de tolerància ϵ que ens permeti definir un rang dins el qual dos valors es consideren acceptables. L'algorisme 7.1 recull el procés de recuperació amb l'aplicació de les explicacions.

És important destacar que aquest nou enfocament no garanteix necessàriament que es seleccionin el mateixos clústers i casos que a l'enfocament basat en la mètrica de distancia. El motiu d'això és fruït dels criteris que es segueixen per seleccionar els elements. A l'enfocament de la mètrica la justificació de la selecció es basa en l'acompliment d'un grau de similitud global de tots els atributs. Per exemple, si de 100 atributs del vector director hi ha 3 que són molt diferents respecte el cas d'entrada, el sumatori normalitzat de les diferències es veurà poc afectat per aquests tres atributs ja que aquesta diferència queda amagada. D'altra banda, la justificació de les explicacions es basa en que totes les relacions atribut-valor s'acompleixen sense excepció. Només que hi hagi un atribut que no s'acompleixi, la similitud es considera 0. Per tant, la mètrica de similitud ens defineix un marge d'acceptació, i les explicacions una acceptació total o nul·la. Conseqüentment, els casos resultants poden diferir entre els dos mètodes, tot i que molts casos seran comuns ja que conceptualment les representacions es basen en el mateix: la similitud dels atributs. Per exemple, la figura 7.4 mostra un exemple on en aquest cas es selecciona el mateix clúster pels dos punts de vista.

Algorisme 7.1: Selecció dels casos més semblants a un cas nou c_i de la memòria de casos mitjançant les explicacions dels clústers.

```
Sigui c_i el cas nou composat per N atributs
Sigui M el conjunt de clústers
Sigui J el nombre de classes diferents
Sigui D_{m,j} l'explicació dels casos del clúster m per la classe j
Sigui CR el conjunt de casos recuperats
CR=∅
Per tot m ∈ M fer
    Per tot j ∈ J fer
        Si c_i satisfà l'explicació D_{m,j} llavors
            CR=CR ∪ {casos del clúster m associats a l'explicació D_{m,j} }
Funció satisfà l'explicació és
    input  : c_i és el cas nou; D_{m,j} és l'explicació de la classe j d'un clúster m
    output : si c_i satisfà D_{m,j}
    Sigui A el conjunt d'atributs que descriuen un cas
    Sigui a un atribut qualsevol de A
    Sigui V_a els possibles valors simbòlics d'un atribut a
    Sigui v_a un dels valors V_a d'un atribut a
    Sigui v_{a,i} un dels valors V_a d'un atribut a pel cas c_i
    Per tot a ∈ D_{m,j} fer
        Si a és enter o real llavors
            Si v_a − ε ≤ v_{a,i} ≤ v_a + ε llavors
                retorna cert
            Sinó
                retorna fals
        Si a és simbòlic llavors
            Si v_a=v_{a,i} llavors
                retorna cert
            Sinó
                retorna fals
```

A banda de les diferències de criteris, hi ha un aspecte que és crític: el nivell de precisió o especificitat de l'explicació. Si les explicacions són molt específiques, pot succeir que no es seleccioni cap cas. D'altra banda, si són generals i flexibles el nombre de clústers i casos serà massa elevat. En qualsevol cas, el problema no és que s'agafin molts casos, sinó que no hi ha cap informació addicional que relacioni el grau de similitud. És aquí on apareix la necessitat de potenciar aquest aspecte mitjançant la introducció de la mètrica de similitud en aquest enfocament. Això permet generar un nou mapa d'estratègies 3D tal com mostra la figura 7.5.

El nou mapa d'estratègies de la figura 7.5 és una taxonomia de les diferents maneres de recuperar els casos tenint en compte els criteris de la mètrica de similitud i de les explicacions. L'eix vertical representa les dues maneres diferents de representar el meta-nivell, l'eix horitzontal representa els dos nivells d'accés i l'eix de profunditat representa el nombre d'elements que intervenen a l'accés. En aquest enfocament les coordenades del mapa d'estratègies s'han reorganitzat respecte l'enfocament del capítol 5 per facilitar la visualització de les quatre branques principals que es poden distingir fruït de la combinació d'ambdós criteris:

1. Mapa d'estratègies basat en la mètrica de distància per seleccionar el/s clúster/s i el/s cas/os. Aquest és l'enfocament que es va analitzar al capítol 5.

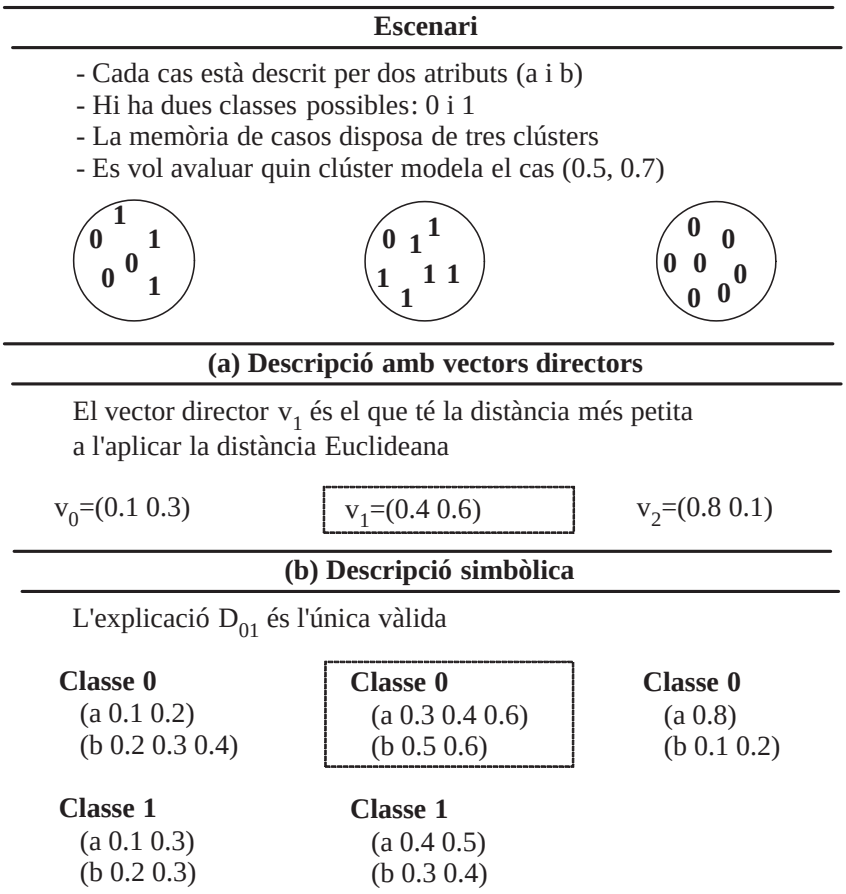

Figura 7.4: L'exemple mostra la selecció del clúster més adequat a partir de (a) vectors directors i (b) explicacions. En aquest cas, seleccionen el mateix clúster.

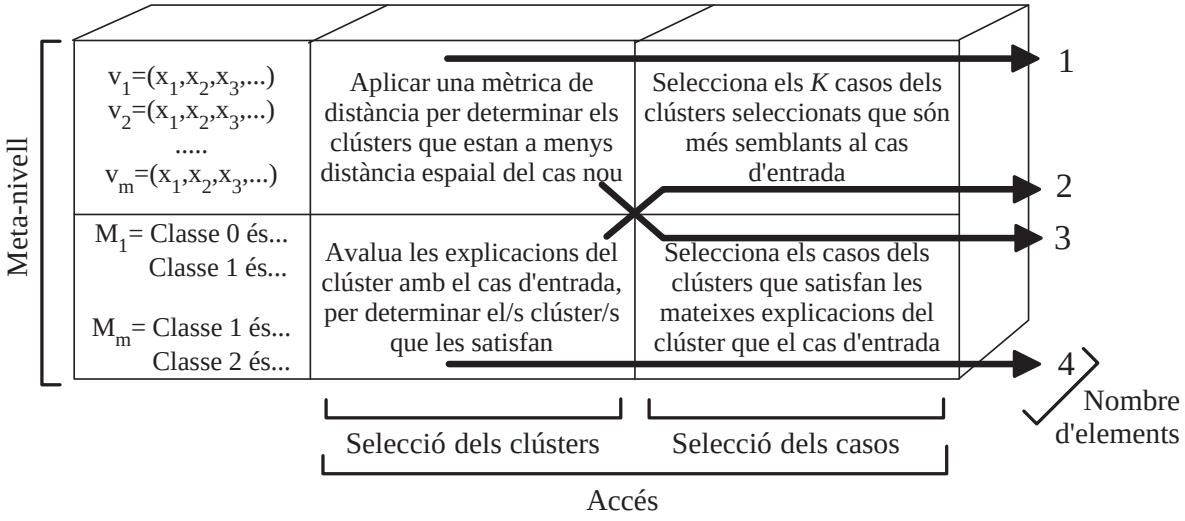

Figura 7.5: El mapa d'estratègies 2D proposat al capítol 5 es converteix en 3D amb la introducció de les explicacions com a mecanisme conjunt amb les mètriques de distància per organitzar la memòria de casos.

2. Mapa d'estratègies basat en les explicacions per seleccionar el/s clúster/s, i en la mètrica de distància per seleccionar el/s cas/os. L'enfocament respon a la necessitat de definir un criteri a través del qual es puguin establir graus de similitud entre els casos recuperats. El tret diferencial respecte el grup 1 d'estratègies és que permet seleccionar automàticament el nombre de clústers, i l'especificitat de les explicacions marcaran la reducció del nombre d'operacions.

3. Mapa d'estratègies basat en la mètrica de distància per seleccionar el/s clúster/s i, en les explicacions per seleccionar el/s cas/os. Aquest enfocament té la mancança que els casos recuperats no poden ser ordenats per similitud. Això fa que el mecanisme de proposta de solució a la fase d'adaptació es basi en la votació de la classe. A més a més, tot i que un clúster tingui una distància petita respecte el cas d'entrada, això no vol dir que els seus casos es facin servir. Aquest fet pot produir-se si l'explicació és massa específica. No obstant, això permet disposar d'un volum de casos recuperats reduït i ad hoc al cas nou.

4. Mapa d'estratègies basat en les explicacions per seleccionar el/s clúster/s i el/s cas/os. Aquest enfocament actua igual que l'anterior amb la diferència que el nombre de clústers es selecciona de manera automàtica. En aquest cas, tot clúster validat implicarà la contribució d'algun cas al menys.

Quin enfocament és millor? Com sempre la resposta és la mateixa: depèn. En dominis estructurats on es disposi d'una mètrica de similitud que aporti una fiabilitat bona, els esquemes 1 i 2 oferiran una degradació de la similitud que serà eficient per discernir quins casos són similars o no al cas d'entrada. L'esquema 2 a més a més serà capaç de seleccionar automàticament el nombre de clústers, tot i que no es podrà influir sobre la reducció d'operacions desitjada. D'altra banda, en dominis no estructurats, variables, o que tinguin una funció de similitud que no sigui molt fiable, els esquemes 3 i 4 podran gestionar millor les dades perquè entenen de relacions atribut-valor i no d'esquemes. No obstant, l'enfocament 3 tindrà la dificultat de l'aplicació de la mètrica si la funció no és fiable i el domini és variable. A part d'això, l'aplicació de la mètrica no garantirà que el clúster contribueixi amb casos.

En qualsevol d'aquests quatre enfocaments, cal tenir present que SOM requereix d'una mesura de similitud per tal de construir els clústers tal com va exposar-se al capítol 3. Per tant, si el domini no permet l'aplicació de mètriques fruït de l'estructura del problema, caldrà definir nous mecanismes per crear els clústers amb l'ajuda de les explicacions.

Resumint, en els enfocaments 1 i 4 és fàcil identificar els entorns òptims d'aplicació perquè depenen només d'un factor, i els enfocaments 2 i 3 requereixen la correcta definició de dos factors.

7.3 Avaluació de la contribució de les explicacions a la interpretació dels casos

Aquest capítol avalua les contribucions de les explicacions dins del SOMCBR des de dos punts de vista: un qualitatiu i un altre quantitatiu. D'una banda, s'avalua la seva contribució sobre els resultats oferts a l'expert en un domini no-supervisat com és l'ANALIA. D'altra banda, es compara la capacitat de les explicacions per descriure el contingut dels clústers respecte els vectors directors sobre un conjunt de *datasets* de l'*UCI Repository* aplicant les estratègies de recuperació basades en l'apartat anterior.

7.3.1 Avaluació qualitativa

L'apartat avalua de manera qualitativa la precisió de les explicacions. Primer es detalla l'experimentació a realitzar, i tot seguit s'analitzen els resultats dels experiments.

7.3.1.1 Experimentació

El projecte ANALIA té com a finalitat el desenvolupament d'una eina per ajudar a l'expert en la detecció de vulnerabilitats d'una xarxa. Dins d'aquest context, en treballs previs del GRSI s'ha demostrat l'èxit de l'aplicació de tècniques de clustering com K-means, X-means, *AutoClass* i més recentment SOM, per detectar patrons 'sospitosos' (Corral et al., 2005b; Corral et al., 2005c; Corral et al., 2005d; Corral et al., 2005a; Corral et al., 2006). El motiu de l'èxit d'aquest tipus de tècniques és que són capaces d'agrupar els dispositius segons certes propietats considerades per la comunitat dels experts com a rellevants per detectar vulnerabilitats (vegeu l'apèndix C). Aquestes agrupacions faciliten molt la tasca de l'analista, ja que li permeten estudiar grups similars de dispositius en lloc de tota la informació de cop.

Aquest capítol vol presentar les explicacions generades a partir de l'aplicació d'una variant de l'operador anti-unificació com un mecanisme que pot ajudar a l'expert a entendre l'organització de la memòria i, d'aquesta manera, comprendre perquè un conjunt de casos són recuperats. Seguint en aquesta idea, aquest apartat vol aplicar la mateixa idea sobre els clústers generats per SOM per validar la seva interpretabilitat de manera qualitativa, ja que és un domini no supervisat. Per aquest motiu, el CBR no jugarà cap paper en aquesta avaluació.

A detall d'exemple la taula 7.1 mostra l'explicació que es generaria sobre tres mostres aplicant la generalització descrita a l'apartat 7.2.1.

Els tests de seguretat per realitzar l'avaluació s'han executat sobre sobre la xarxa d'Enginyeria i Arquitectura La Salle. Concretament, s'han recopilat dades de 44 dispositius de (1) 21 dispositius de dos laboratoris d'alumnes (14 i 7 ordinadors respectivament), (2) 9 servidors públics, (3) 11 servidors interns i (4) 3 ordinadors de professors. El format de la representació de les dades és l'especificat a la taula C.1(a) de l'apèndix C, la qual és la representació que ha demostrat ser més efectiva en estudis previs. Sobre aquest conjunt de dades s'han aplicat diferents algorismes de clustering:

- Els algorismes K-means i X-means, fent servir diferents valors de K entre 3 i 8.
- L'algorisme SOM amb diferents mides de mapa 2×2, 3×3, 4× 4, 5×5 and 6×6.
- L'algorisme Auto-Class amb selecció automàtica del nombre.

A més a més, s'han fet servir 10 llavors diferents per compensar els efectes aleatoris de les inicialitzacions. Tenint en compte totes aquestes configuracions, s'han aplicat els criteris de intra/inter cohesió presentats en (Corral et al., 2006) per seleccionar la millor configuració, la qual és analitzada a l'apartat següent.

Taula 7.1: Aplicació de la variant de l'anti-unificació sobre tres exemples. És important remarcar que el port 25 no es té en compte perquè pren tots els possibles valors (0, 1 i 2).

	Ports						**W2000**		**XP**	**W2003**	**Notes de seguretat**
	21	**25**	**53**	**80**	**135**	**445**	**SP3**	**SP4**	**SP2**	**Server**	
e_1	1	0	0	0	1	1	41%	41%	41%	41%	3
e_2	1	2	1	0	1	1	41%	41%	41%	41%	6
e_3	1	1	0	1	1	1	41%	41%	41%	41%	6
D_i	1	-	-	-	1	1	41%	41%	41%	41%	(3,6)

7.3.1.2 Anàlisi i discussió dels resultats

La figura 7.6 mostra tres dels clústers generats que fan referència als laboratoris dels alumnes. La resta s'han omès per motius de seguretat al ser IP públiques. Cada laboratori té una configuració única i, per tant, la informació de clustering referent a aquestes alumnes hauria de ser dues agrupacions. No obstant, la figura mostra que hi ha un dispositiu, el 10.0.14.203, que té propietats diferents als altres dos. La interpretació d'aquest fet és ben senzilla: algun alumne ha modificat la configuració de la màquina, o bé, ha instal·lat algun *software* que compromet la seguretat. Per tant, estem davant d'un clar exemple on el clustering permet d'una manera ràpida i fàcil ajudar a l'expert a detectar vulnerabilitats.

D'altra banda, la figura 7.7 ens proporciona una altra perspectiva interessant basada en representar una estadística de quants cops un dispositiu ha estat clusteritzat amb els altres dispositius per totes les execucions indicades a l'apartat anterior. Curiosament, el dispositiu 28 amb IP 10.0.14.203 destaca de la resta perquè mai s'ha agrupat amb ningú. En canvi, els dispositius 1,

Clúster 1
10.0.14.206 - 10.0.14.207 - 10.0.14.208 - 10.0.14.209 - 10.0.14.201 - 10.0.14.202

Clúster 5
10.0.14.203

Clúster
10.0.14.130 - 10.0.14.131 - 10.0.14.132 - 10.0.14.133 - 10.0.14.134 - 10.0.14.135
10.0.14.136 - 10.0.14.137 - 10.0.14.138 - 10.0.14.139 - 10.0.14.140 - 10.0.14.141
10.0.14.142 - 10.0.14.443

Figura 7.6: L'anàlisi dels dispositius dels laboratoris dels alumnes mostra que hi ha tres tipologies de configuració, tot i que només hi haurien d'haver dues.

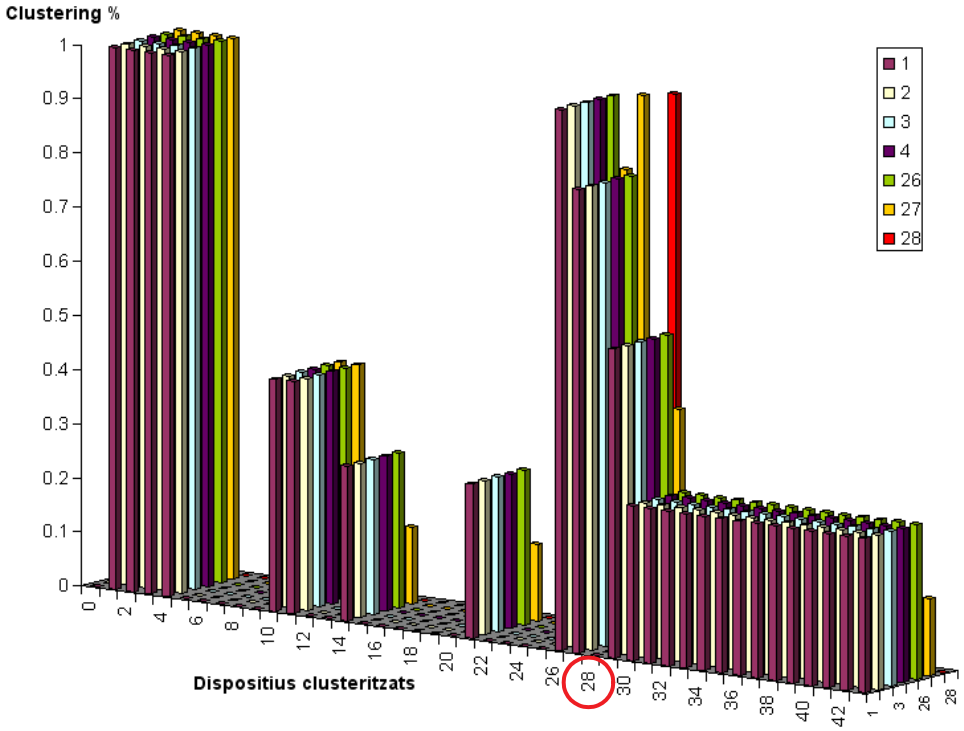

Figura 7.7: Resultats de la clusterització dels laboratoris.

Taula 7.2: Explicacions dels clúster 1 i 5 referents a un dels laboratoris on la seguretat està compromesa.

	Ports					W2000		XP		2003	
Cluster	**135**	**139**	**445**	**781-807**	**904-930**	**WS_SP4**	**S_SP2**	**SP1**	**SP2**	**St.**	**Ent.**
1	-	1	-	-	-	67%	67%	67%	67%	70%	70%
5	1	1	1	2	2	67%	67%	67%	67%	70%	70%

2, 3, 4 i 26 s'han agrupat sempre junts, i el 27 s'ha agrupat amb ells més d'un 80% dels cops. Aquests dispositius fan referència al clúster 1 indicat a la figura 7.6. Per tant, la trobada de la vulnerabilitat anterior no ha estat casualitat. Arribats a aquest punt l'analista ja sap que hi ha un problema, però com l'afronta? És aquí on l'aplicació de les explicacions sobre els clústers generats pot ajudar a conèixer perquè determinats dispositius s'han agrupat en un mateix clúster. La descripció simbòlica li permetrà disposar d'informació valuosa per detectar, aïllar i solventar l'esquerda de seguretat.

La taula 7.2 mostra les explicacions generades pels clústers 1 i 5. La descripció indica que el sistema operatiu del clúster 5 no ha estat modificat respecte el clúster 1. No obstant, el clúster 5 conté un alt nombre de ports filtrats (valor=2), és a dir, un port filtrat significa que hi ha un *firewall*, filtre, o algun obstacle a la xarxa que està bloquejant el port i evita que el sistema determini si està o no obert. A més a més, els ports 135 i 445 estan oberts, el que vol dir que els serveis MSRPC i Microsoft-DS [2] han estat alterats. Els ports haurien de bloquejar-se per evitar atacts a l'operatiu. Per tant, l'analista ha pogut conèixer d'una manera ràpida el perquè el dispositiu presentava vulnerabilitats per tal de poder prendre les mesures oportunes sense haver d'analitzar totes les vulnerabilitats dels dispositius del laboratori.

7.3.2 Avaluació quantitativa

L'apartat avalua de manera quantitativa la precisió de les explicacions. Primer es detalla l'experimentació a realitzar, i tot seguit s'analitzen els resultats dels experiments.

7.3.2.1 Experimentació

L'avaluació de la precisió de les explicacions per descriure el contingut dels clústers amb la finalitat de seleccionar els clústers i els casos es realitza sobre diferents *datasets* de l'*UCI Repository* descrits a la taula 7.3. Tot i que les explicacions funcionen sobre atributs numèrics i simbòlics, els *datasets* només tenen atributs numèrics ja que la mètrica de similitud que es farà servir serà la mateixa que als experiments dels capítols anteriors. D'altra banda, el volum dels *datasets* és més reduït respecte la resta dels estudis dels altres capítols perquè l'experimentació ha necessitat un procés manual de tres etapes: (1) generar els clústers, (2) aplicar el software de la Dra. Eva Armengol per generar les explicacions dels clústers i (3) executar el SOMCBR amb les explicacions generades com a nous descriptors. Per aquest motiu s'han seleccionat *datasets* majoritàriament de la complexitat B, ja que és el comportament intermig. Com es veurà a l'últim apartat, degut al bon funcionament de les explicacions en el SOMCBR es té previst realitzar una automatització del procés mitjançant un procés d'integració dels processos, amb el que seria possible estudis dels efectes dels enfocaments tenint en compte la complexitat del domini.

La precisió de les explicacions s'avalua com la capacitat que tenen les explicacions per seleccionar els clústers i casos més adequats. Per aquest motiu s'estudien i comparen els enfocaments 1, 2 i 4 descrits a l'apartat 7.2.3. L'enfocament 3 s'ha descartat ja que en el millor dels casos funcionarà igual de bé que l'enfocament 4. Els enfocaments 2 i 4 es realitzen a partir d'explicacions

[2]MSRPC: *Microsoft Remote Procedure Call*. Microsoft-DS: Port utilitzat per compartir arxius en Windows.

Taula 7.3: Descripció dels *datasets* avaluats (nom, nombre d'atributs, d'instàncies i de classes). Els *datasets* s'ordenen per nombre d'instàncies.

Codi	***Dataset***	**Atributs**	**Instàncies**	**Classes**	**Complexitat**
HE	hepatitis	19	155	2	B
GL	glass	9	214	6	A/B
TH	thyroids	5	215	3	A/B
HS	heart-statlog	13	270	2	C
IO	ionosphere	34	351	2	B
WD	wdbc	30	569	2	A
BA	bal	4	625	3	C
WB	wbcd	9	699	2	B
WI	wisconsin	9	699	2	B
VE	vehicle	18	846	4	B
TA	tao	2	1888	2	B
SE	segment	19	2310	7	A/B
WA	waveform	40	5000	3	B

generades amb la variant de l'operador d'anti-unificació descrita a l'apartat 7.2.1. La funció de distància utilitzada als enfocaments 1 i 2 és el complement de la distància euclidiana. A més a més, per tal de poder compensar el fet que les propostes dels enfocaments 2 i 4 puguin seleccionar de manera automàtica el nombre de clústers, l'enfocament 1 s'estudia fent servir la configuració òptima que va deduir-se al capítol 5 basada en fer servir 3 clústers. La resta de paràmetres comuns a l'experimentació són els següents:

- La funció de distància utilitzada tant per la construcció dels models, com per la comparació entre clústers i casos és la del complement de la distància Euclidiana (vegeu l'equació 5.1).
- La mida del mapa s'assigna de manera automàtica tal com s'explica al capítol 3, és a dir, es selecciona la mida que minimitza l'error. El rang de mides avaluades va de 2 a 6.
- Els models poden tenir diferent nombre de casos.
- La fase d'adaptació proposa la nova solució fent servir K-NN igual a 1, 3 i 5 en els enfocaments 1 i 2, i la classe majoritària de tots els casos recuperats a l'enfocament 4.
- La fase d'emmagatzematge no guarda nous casos.
- Cada resultat s'obté d'aplicar un *10-fold stratified cross-validation*.
- Cada configuració és la mitja de 10 llavors per tal de compensar els efectes aleatoris de la construcció dels models.

7.3.2.2 Anàlisi i discussió dels resultats

Les taules 7.4, 7.5, 7.6, 7.7 i 7.8 mostren el percentatge d'error, la desviació típica i el percentatge de reducció mig dels casos seleccionats a la fase de recuperació per totes les configuracions anteriors. A més a més, les taules indiquen els valors migs per aquestes estadístiques.

Les taules 7.4 i 7.5 contenen les estadístiques pel sistema SOMCBR fent servir les explicacions per seleccionar els clústers, i la mètrica de distància per seleccionar els casos. En ambdós casos es mostren els resultats fent servir diferents valors K-NN per tal de comparar posteriorment l'efecte de fer servir més casos a l'hora de proposar la solució. La diferència entre les dues taules és el

Taula 7.4: Estadístiques del percentatge d'error (%Error), la seva desviació típica (σ), i la percentatge mig de reducció dels casos recuperats a la fase de recuperació (%R) pel SOMCBR seguint l'enfocament 2 amb ϵ igual a 0.1.

Codi	1-**NN**		3-**NN**		5-**NN**	
	%Error (σ)	%R	**%Error** (σ)	%R	**%Error** (σ)	%R
BA	23.4 (3.2)	44.1	19 (2.9)	44.1	15.2 (2.6)	44.1
GL	40.5 (9.8)	69.8	37.1 (17.1)	45.8	39.2 (15.4)	45.8
HS	23.4 (6.4)	40.7	20.8 (5)	40.7	19.8 (5.7)	67.9
HE	20.5 (8)	69.1	16.2 (7.5)	69.1	15.4 (9.5)	69.1
IO	12.9 (6.4)	89.8	11.3 (6)	89.8	12.6 (5.8)	89.8
SE	3.2 (1.3)	55.1	4 (0.9)	55.1	5 (1.2)	55.1
TA	3.7 (1.6)	8.2	4.2 (2)	8.2	3.3 (1)	8.2
TH	3.3 (2.2)	17.6	7.1 (5.7)	17.6	7.1 (5.3)	17.6
VE	30.4 (4.9)	15	29.6 (4.5)	15	29.1 (6.2)	15
WA	26.8 (1.9)	14	22.7 (1.4)	21.7	20.8 (1.6)	14
WB	3.6 (3.6)	63.9	4 (3.6)	63.9	3.8 (2.4)	65.7
WD	5.1 (2.5)	59.8	3.7 (2.2)	59.8	3.4 (2)	34.2
WI	4.6 (2.7)	60.6	4.2 (1.7)	79.2	4.3 (2.8)	60.6
	14	52.4	11.6	61.7	9.8	52.4

Taula 7.5: Estadístiques del percentatge d'error (%Error), la seva desviació típica (σ), i la percentatge mig de reducció dels casos recuperats a la fase de recuperació (%R) pel SOMCBR seguint l'enfocament 2 amb ϵ igual a 0.2.

Codi	1-**NN**		3-**NN**		5-**NN**	
	%Error (σ)	%R	**%Error** (σ)	%R	**%Error** (σ)	%R
BA	23.4 (3.2)	44.1	19 (2.9)	44.1	15.2 (2.6)	44.1
GL	39.3 (7.4)	45.3	38.9 (17.6)	42.2	39.9 (17.9)	42.2
HS	25 (8)	54.7	21.2 (5.5)	33.7	18.2 (5.1)	33.7
HE	21.5 (8.1)	61.9	16.3 (7.7)	61.9	15.6 (7.5)	61.9
IO	14.5 (4)	34.3	16.2 (3.9)	34.3	18.2 (4)	34.3
SE	2.9 (1.2)	32.7	4 (0.8)	32.7	4.9 (1.1)	32.7
TA	3.8 (1.6)	1.4	4.3 (2)	1.4	3.4 (0.9)	1.4
TH	3.3 (3.1)	17.1	6.1 (4.8)	17.1	6.1 (4.8)	17.1
VE	29.9 (4.1)	3.8	30 (4.9)	3.8	29.3 (4.8)	3.8
WA	27 (1.9)	0.4	22.6 (1.7)	1	21.1 (1.6)	1
WB	4.7 (2.8)	24.5	3.4 (1.5)	24.5	2.9 (2.1)	24.5
WD	5.3 (2.5)	12.7	3.5 (1.6)	25.4	3.4 (2)	26.6
WI	3.9 (1.6)	22.6	3.4 (1.1)	22.6	2.9 (1.9)	22.6
	13.6	33.4	11.2	33.4	9	33.4

factor ϵ utilitzat per validar l'explicació amb valor 0.1 i 0.2 respectivament. L'increment del valor ϵ té un efecte immediat sobre les estadístiques tal com mostren les mitges dels resultats en els tres valors de K-NN. Al fer '*matching*' amb més casos perquè el rang del valor de l'atribut és major, el percentatge d'error es redueix perquè més casos es tenen en compte. Al mateix temps, això implica una reducció en el nombre mig de casos recuperats com pot veure's. Això no vol dir que les explicacions funcionin malament en el primer cas, només que amb un valor de 0.1 estan essent massa específiques i algun atribut pot donar-nos problemes i no validar l'explicació. Cal recordar que amb les explicacions només que un atribut sigui molt diferent, l'explicació no es satisfà.

La capacitat de les explicacions per seleccionar els clústers pot comprovar-se a partir de la comparació dels resultats oferts per l'enfocament 1 a la taula 7.6. Les estratègies de l'enfocament 2 ofereixen un percentatge d'error igual o millor (però no estadísticament diferent si s'aplica un t-test) que l'estratègia de l'enfocament 1. El motiu és que l'enfocament 2 tendeix a seleccionar més de 3 clústers i, conseqüentment, es té una visió més àmplia de la memòria. Aquest efecte

Taula 7.6: Estadístiques del percentatge d'error (%Error), la seva desviació típica (σ), i la percentatge mig de reducció dels casos recuperats a la fase de recuperació (%R) pel SOMCBR seguint l'enfocament 1.

Codi	1-**NN**		3-**NN**		5-**NN**	
	%Error (σ)	%R	**%Error** (σ)	%R	**%Error** (σ)	%R
BA	23.7 (3.9)	60	19.7 (2.8)	67.4	18.7 (4)	4
GL	33.6 (14.4)	47.9	31.3 (10.2)	21.4	33.2 (9.6)	9.6
HS	24.1 (8.8)	65.4	20.4 (5.8)	56	21.1 (3.7)	3.7
HE	19.4 (7.1)	67.6	13.6 (8.7)	56.1	16.8 (7.3)	7.3
IO	12.8 (4.5)	50.8	13.7 (4.9)	12.7	14.3 (4.8)	4.8
SE	4.7 (0.9)	56.5	5.7 (1.3)	56.5	6 (1.2)	1.2
TA	5.6 (1.7)	55.6	3.8 (1.8)	55.6	4.4 (2.1)	2.1
TH	2.8 (2.3)	3.1	7.9 (5.2)	52.8	5.1 (5)	5
VE	32 (4.6)	55.2	30.7 (6.7)	55.2	31.9 (5)	5
WA	27.2 (2.2)	76.5	23.8 (1.7)	54	21.8 (1.7)	1.7
WB	4.9 (1.5)	52.9	3.9 (2)	65.2	3.9 (1.9)	1.9
WD	4 (2.8)	44.3	3.3 (2)	44.3	4.2 (2.6)	2.6
WI	4 (1.2)	55.3	3.4 (1.7)	55.3	3.2 (1.2)	1.2
	13.9	57.7	11.6	61.4	11	2.6

resta amagat en l'enfocament 2 amb valor ϵ=0.1 perquè molts casos són filtrats degut a la poca flexibilitat de l'explicació. No obstant, la compensació desapareix si la flexibilitat de l'explicació s'augmenta tal com passa amb el valor ϵ de 0.2. Per tant, les explicacions són un mecanisme de selecció automàtic de clúster que permet ajustar l'exploració dels casos segons l'especificitat. Això sens dubte és un valor afegit molt important, i restaria veure en futures anàlisis amb més *datasets* l'impacte que té sobre la complexitat, així com sobre la naturalesa de les dades.

D'altra banda, les taules 7.7 i 7.8 contenen les estadístiques per les dues estratègies basades en l'enfocament 4 amb valors de 0.1 i 0.2 de ϵ respectivament. Els resultats en aquest cas són clarament més dolents que els altres perquè tant el percentatge d'error com la reducció del nombre de casos recuperats empitjora molt. El motiu d'aquest efecte es produeix perquè les explicacions no han sigut suficient específiques per trobar un conjunt de casos reduït que representi bé el nou problema. També és important destacar el paper invers del valor ϵ en aquest enfocament. Abans interessava que el valor permetés una certa flexibilitat per afavorir la selecció de clústers, ja que després un altre procés les filtraria (la mètrica de similitud). En canvi, ara interessa que l'explicació sigui el més restrictiva possible perquè no hi ha cap més procés de filtratge. Per analitzar l'impacte de la flexibilitat de les explicacions, les estratègies d'aquest enfocament s'han avaluat per diferents mides de mapes de les quals es mostren els resultats per les mides 3×3, 4×4 i 5×5. Quan el mapa sigui més gran, hi haurà més clústers amb menys clústers i, per tant, les generalitzacions seran més específiques. Aquest efecte pot comprovar-se mirant com a mesura que augmenta el mapa, les mitges d'error i reducció de casos es redueixen. No obstant, aquesta millora no s'acosta als resultats oferts pels enfocaments anteriors. Les explicacions no han estat suficientment específiques. A més a més, les configuracions K-NN en aquest cas no tenen sentit perquè la proposta de classe és fa sempre amb tots els casos seleccionats, ja que no hi ha K casos més semblants: o són o no són iguals.

Per tant, els resultats mostren com les explicacions són un mecanisme automàtic per seleccionar el nombre de clúster gràcies al qual es pot arribar a simplificar la configuració del SOMCBR. D'altra banda, les explicacions aïllades com a mecanisme de selecció de casos no han ofert bons resultats perquè no són capaces de discernir entre els graus de similitud. A més a més, els resultats serien molt diferents si els dominis haguessin estat no estructurats i variables. Per tant, l'ampliació del mapa d'estratègies amb les explicacions ens obre un ampli camp de recerca.

Taula 7.7: Estadístiques del percentatge d'error (%Error), la seva desviació típica (σ), i la percentatge mig de reducció dels casos recuperats a la fase de recuperació (%R) pel SOMCBR seguint l'enfocament 4 amb ϵ igual a 0.1.

Codi	**3×3**		**4×4**		**5×5**	
	%Error (σ)	%R	**%Error (σ)**	%R	**%Error (σ)**	%R
BA	39.8 (4.4)	44.1	42.2 (9.5)	60.1	38.9 (6)	87
GL	57.6 (11.4)	69.8	56.7 (10.4)	45.8	61.7 (12.1)	84.9
HS	33.1 (11.5)	40.7	37.7 (10.4)	67.9	37.6 (10.6)	79.4
HE	17.1 (6.4)	69.1	20.8 (9)	87.1	30.8 (17.2)	93.5
IO	29.8 (5.4)	69.8	35.3 (15.9)	66.3	14.5 (6.4)	89.8
SE	23.5 (3)	55.1	52.1 (8.5)	91.4	18.6 (3.8)	71
TA	36.4 (2)	1.9	33.7 (14.3)	59.3	22.5 (3.1)	8.2
TH	29.4 (4.3)	36.3	29.1 (4.2)	17.6	26.1 (5.7)	62.2
VE	60.9 (3.7)	15	57.8 (8.5)	43.9	58.4 (7.8)	57.2
WA	59.3 (1)	2.9	40.2 (1.3)	14	35.9 (3.1)	21.7
WB	18.1 (13.9)	63.9	22.2 (12.5)	65.7	10.9 (4.7)	83.9
WD	32.3 (6.3)	34.2	26 (10.7)	51.6	11.2 (7.5)	59.8
WI	13.2 (14.3)	60.6	11.6 (11.4)	76.5	4.7 (1.8)	79.2
	26.5	52.4	26.9	68.3	21.8	83.1

Taula 7.8: Estadístiques del percentatge d'error (%Error), la seva desviació típica (σ), i la percentatge mig de reducció dels casos recuperats a la fase de recuperació (%R) pel SOMCBR seguint l'enfocament 4 amb ϵ igual a 0.2.

Codi	**3×3**		**4×4**		**5×5**	
	%Error (σ)	%R	**%Error (σ)**	%R	**%Error (σ)**	%R
BA	39.8 (4.4)	44.1	42.2 (9.5)	60.1	38.9 (6)	87
GL	61.1 (16.1)	45.3	59.6 (11.6)	42.2	65.7 (15)	57.8
HS	34.6 (10.6)	33.7	39.6 (6.5)	54.7	34.7 (9.1)	68.3
HE	18.5 (5.8)	61.9	19.7 (10.5)	83.5	24.1 (12.5)	87.8
IO	34.2 (5.7)	34.3	33.2 (14.4)	61	33.1 (5)	73.3
SE	46.8 (1.9)	32.7	60.3 (6.4)	88.2	44.7 (7.8)	42.9
TA	44.9 (2)	0.2	43.1 (7.6)	51.4	31 (7.3)	1.4
TH	29.8 (3.8)	17.1	29.4 (4.3)	11.4	26.5 (4.8)	32.6
VE	71 (3.5)	3.8	67.9 (5.5)	18.5	61.9 (2.9)	22.9
WA	66.1 (0.2)	0	65.6 (0.4)	0.4	64.9 (0.7)	1
WB	23.9 (12.2)	24.5	22.2 (15.1)	44.2	20.7 (13.1)	45.9
WD	35.2 (3.2)	12.7	31 (8.9)	26.6	32.4 (3.9)	25.4
WI	16.3 (12.9)	22.6	14.9 (12.8)	54.8	9.9 (9.5)	55.8
	28.1	33.4	28.6	57.5	24.4	71.4

7.4 Conclusions i línies futures

L'organització de la memòria de casos amb SOM permet definir grups de casos semblants a través d'un procés no supervisat. Això permet identificar relacions entre les dades, les quals són molt útils per potenciar les fases del CBR gràcies a la informació que proporcionen els vectors directors. A més a més, això aporta un altre valor afegit al sistema, el qual és que permet a l'expert entendre les relacions entre els casos i determinar perquè determinats casos són recuperats. No obstant, la comprensió dels vectors directors pot ser complexe per l'expert degut al seu baix nivell de representació.

Aquest capítol ha presentat les explicacions com un mecanisme per dotar als clústers d'una representació que faci servir el mateix llenguatge que es fa servir per representar els casos i, d'aquesta manera, millorar el grau de comprensió per part de l'expert. A banda de millorar la interpretació de l'expert perquè processa la informació de manera diferent, les explicacions

generades poden aprofitar-se per definir nous mecanismes de selecció de clústers i casos basats en fer servir les explicacions en lloc dels vectors directors. Aquest nou enfocament ha permès la definició d'un mapa d'estratègies en 3D on la combinació dels criteris de selecció basats en mètriques de similitud i explicacions obren un ampli ventall de possibilitats. Concretament, aquestes es poden resumir en quatre tipus: (1) selecció de clústers i casos amb mètriques de similitud, (2) selecció de clústers amb explicacions i casos amb mètriques de similitud, (3) selecció de clústers amb mètriques de distància i casos amb explicacions i (4) selecció de clústers i casos amb mètriques. L'èxit de les estratègies estarà directament relacionat amb la tipologia de les dades, on la viabilitat per definir funcions de similitud i explicacions específiques seran la clau per recuperar la informació més adient.

L'avaluació de les explicacions s'ha fet des de dos punts de vista. D'una banda, tenint en compte la seva contribució a la fase de revisió per ajudar a un expert a entendre millor el perquè de les coses. Aquest enfocament al ser totalment qualitatiu perquè depèn de l'opinió d'un expert, s'ha fet a partir de la problemàtica del projecte ANALIA basada en el desenvolupament d'una eina per la detecció de vulnerabilitats de la xarxa. L'eina es basa en aplicar tècniques de clustering per agrupar els dispositius segons certs criteris considerats pels experts com a rellevants per determinar esquerdes de seguretat, ja que és un domini no supervisat on hi ha una definició de classes. Tot i que el CBR no ha tingut cap paper, les explicacions generades per SOM sobre la memòria de casos formada pels casos han demostrat ser útils per ajudar a l'expert a desenvolupar la seva tasca.

D'altra banda, s'ha avaluat d'una manera quantitativa la precisió de les explicacions per representar el contingut de la memòria, concretament, per seleccionar els clústers i casos. Tot i que el joc de dades no era tot l'ampli que ens hagués agradat fruït del procés d'experimentació, els resultats són molt prometedors i han mostrat com les explicacions són un mecanisme que de manera automàtica pot seleccionar el nombre de clústers d'una manera òptima. D'altra banda, la seva limitada capacitat per discernir entre graus de similitud ha fet que els resultats on la selecció dels casos es feia amb explicacions fossin dolents.

Addicionalment, el seu desenvolupament ha conclòs amb la definició d'un mapa d'estratègies en 3D on els criteris de selecció dels clústers i dels casos han permès ampliar significativament el ventall de possibilitats. Les contribucions d'aquest capítol es troben publicades als articles següents:

- G. Corral, A. Fornells, E. Armengol i E. Golobardes. *Data security analysis using unsupervised learning and explanations.* Al llibre *Innovations in Hybrid Intelligent Systems*, volum 44. Editors: E. Corchado, J.M. Corchado, i A. Abraham. Springer-Verlag, 2007. En impremta.

- A. Fornells, E. Armengol, i E. Golobardes. *Explanation of a clustered case memory organization.* Al llibre *Artificial Intelligence Research and Development*, volum 163, pàgines 153-160. IOS Press, 2007.

Finalment, les línies futures poden dividir-se en els punts següents:

- Treballar amb explicacions discriminatòries, les quals facin servir exemples d'altres casos com a negatius.

- Automatitzar el procés d'experimentació per ampliar els jocs de dades per tal d'analitzar el rendiment tenint en compte la complexitat del domini segons la geometria de les dades, així com sobre l'estructuració i variabilitat dels casos.

- Definir un nou procés de clustering que tingui en compte les propietats dels dominis no estructurats.

Resum

L'expert té la missió de validar la proposta del sistema a la fase de revisió. La dificultat que apareix és que sovint no pot assegurar categòricament la solució proposada, perquè si la conegués no necessitaria disposar de cap sistema que li digués.

L'organització de la memòria de casos aporta molts beneficis tal com s'ha vist als capítols anteriors. A banda d'això, l'expert també pot beneficiar-se d'aquesta organització ja que pot ajudar-lo a entendre les relacions entre els casos, així com els motius pels quals determinats casos són o no recuperats. No obstant, l'expert es troba amb una nova dificultat, la qual és la complexitat d'entendre el baix nivell amb el qual els vectors directors són representats. És per aquest motiu que aquest capítol ha introduït les explicacions simbòliques com un mecanisme per potenciar la descripció de l'organització de la memòria. Una explicació simbòlica pot definir-se com un conjunt de premisses que mostren d'una manera esquemàtica una certa informació. En aquest cas, s'ha fet servir una variant de l'operador anti-unificació per generar la generalització més específica dels atributs comuns entre un conjunt d'elements. D'aquesta manera, les relacions entre els casos s'expressen amb el mateix llenguatge dels casos i, conseqüentment, l'expert pot comprendre-les millor.

La introducció de les explicacions simbòliques en el SOMCBR es basa en descriure el contingut de cadascun dels clústers. Això permet a l'expert disposar d'explicacions més entenedores. D'altra banda, aquesta capacitat per descriure els clústers pot ser interessant per determinar la selecció dels clústers i dels casos. Aquesta reflexió ens ha portat a definir un nou mapa d'estratègies en 3D, on es combinen els criteris de similitud presentats al capítol 5 amb els criteris de les explicacions simbòlics. En línies generals, la combinació dels criteris anteriors permet identificar quatre enfocaments principals: (1) selecció dels clústers i dels casos amb les mètriques de similitud, (2) selecció dels clústers amb les explicacions i els casos amb les mètriques de similitud, (3) selecció dels clústers amb les mètriques de distància i els casos amb les explicacions i (4) la selecció dels clústers i els casos amb les mètriques. En qualsevol cas, la precisió de la mètrica i/o de l'explicació determinarà l'èxit del sistema.

La validació del comportament de les explicacions simbòliques com a mecanisme per representar la memòria de casos s'ha avaluat satisfactòriament en un domini no supervisat de maner qualitativa, i en dominis supervisat de manera quantitativa. Finalment, s'ha vist com les explicacions poden jugar un paper molt important en la definició de polítiques de selecció automàtica dels clústers.

Capítol 8

Fase d'emmagatzematge

El CBR és un paradigma que aprofita la seva experiència per resoldre els problemes nous i, per tant, el manteniment d'aquest coneixement és crític per garantir el seu funcionament. Aquest és precisament l'últim aspecte amb el qual falta enfrontar-se per tancar el cicle del SOMCBR. El gran inconvenient que aquí ens apareix és que SOM és un mètode no supervisat i, a més a més, no és un mètode incremental. Aquest capítol presenta una proposta d'estratègia de manteniment de la memòria de casos per tal d'introduir, actualitzar i esborrar coneixement de la memòria de manera incremental. A més a més, aquesta estratègia és semi-supervisada perquè d'una banda es fa servir el *feedback* de l'expert respecte si el sistema ha resolt bé el cas nou, i d'altra banda els casos s'autoorganitzen de manera no supervisada.

8.1 Motivació: el repte d'aprendre

Com s'ha avaluat al llarg dels anteriors capítols, l'organització de la memòria de casos amb SOM permet aprofitar el coneixement ocult de les dades per autoorganitzar-les mitjançant la definició de patrons en forma de vectors directors. D'aquesta manera, és possible discernir entre els casos que són potencialment útils dels que no ho són, amb la conseqüent millora de rendiment.

A més a més de resoldre problemes tenint en compte la seva experiència, els sistemes CBR han d'aprendre dels problemes nous que resolen, així com dels errors que comenten. Aquesta tasca contínua d'aprenentatge els permet disposar d'una representació més fidel de la realitat i, conseqüentment, disposar d'informació més fiable per resoldre els problemes nous. Aquest tipus de tasques es coneixen sota el nom de polítiques de manteniment de la memòria de casos (*Case Base Maintenance*), i han d'establir que fer davant de:

- Coneixement nou del qual no es tenia constància.
- Coneixement nou que invalida d'altre que es tenia anteriorment.
- Coneixement nou que refina i ajuda a discernir situacions ambigües.

La definició d'aquestes polítiques en el nostre cas té un grau addicional de complexitat, fruit de la dimensió introduïda mitjançant els vectors directors per organitzar el coneixement. A banda d'establir les condicions sota les quals s'han de guardar o esborrar els casos, ara cal establir com actualitzar l'indexació definida inicialment per SOM. D'aquesta manera, si el coneixement de la memòria canvia, l'organització definida pels vectors directors també ha d'actualitzar-se per garantir la seva consistència. El problema és que SOM no permet l'actualització del coneixement que modela de manera incremental i, per tant, qualsevol canvi que es vulgui fer implica haver de reentrenar des de zero el mapa. Aquest aspecte pot tenir un impacte molt negatiu en entorns molt dinàmics i de temps real, degut a la important despesa computacional que això suposa.

Aquesta limitació ha fet que molts autors hagin abordat l'aprenentatge incremental de SOM, però sempre des d'un punt de vista no supervisat. Això és perquè la finalitat de SOM és agrupar les dades segons la seva rellevància i distribució topològica de l'espai original, i no la de classificar o predir un resultat.

Les estratègies que aborden l'aprenentatge incremental estan caracteritzades per, a partir d'un estat inicial de la xarxa, introduir o esborrar neurones a partir d'una funció d'error. Dins d'aquest ventall d'algorismes, les diferències entre elles es basen en (1) l'estructura de les neurones, (2) els criteris per actualitzar els vectors directors i les relacions entre les neurones, (3) i la definició de la funció d'error.

GCS (*Growing Cell Structure*) (Fritzke, 1994) i GNG (*Growing Neural Gas*) (Fritzke, 1996) defineixen una estructura basada en hipertetraedres d'una dimensionalitat prèviament seleccionada. Un hipertetraedre és el poliedre més simple d'una dimensió D. Per exemple, l'hipertetraedre per $D \in \{1,2,3\}$ seria la línia, el triangle i el tetraedre. En ambdós casos el procés d'aprenentatge consisteix en: (1) s'inicialitza el model amb un hipertetraedre, on els cantons uneixen les neurones; (2) donat un nou exemple, es busca la neurona q amb el màxim error acumulat; (3) l'arc més llarg que uneixi la neurona q amb una altra p és parteix en dos segments; (4) s'introdueix una nova neurona que uneixi l'arc anteriorment dividit. En aquests dos algorismes la diferència es basa en la connexió entre les neurones. GCS obliga que la nova neurona es connecti amb tots els veïns de p i q per mantenir l'estructura d'hipertetraedre. En canvi, GNC aplica l'aprenentatge Hebbià (White, 1992) per definir les unions addicionals. Per tant, amb GCS s'aconsegueix una estructura regular que facilita la visualització, i amb GNC s'aconsegueix una representació més fidel a la realitat però més complexa de visualitzar a conseqüència de la no uniformitat de les dimensions.

D'altra banda, GG (*Growing Grid*) (Fritzke, 1995) i GSOM (*Incremental Self-Organizing Network*) (Bauer i Villmann, 1997) es basen en estructures d'hiperrectangles. GC actua com GCS però l'introducció de neurones es basa en afegir hiperfiles o hipercolumnes per garantir l'estructura d'hiperrectangle en el graf que representa la xarxa. En canvi, GSOM introdueix una nova estructura hiperrectangular en la neurona que tingui el màxim error acumulat.

A més a més, d'aquest tipus d'estructures amb N dimensions, altres aproximacions es basen en mantenir la clàssica estructura 2-D de SOM com l'algorisme ILM (Benabdeslem, 2006). Aquest algorisme defineix una graella 2D, on l'actualització de la topologia es basa en expandir les neurones ubicades al perímetre de la reixa quan el nombre d'elements de la neurona és major que la mitja d'elements per neurona. Tot seguit, optimitza el nombre final de neurones.

La finalitat d'aquest capítol és la de presentar una estratègia que, independentment de si s'ha d'integrar coneixement nou, esborrar-ne, o refinar-ne l'existent, sigui capaç de reajustar els vectors directors definits per SOM de manera incremental sense que això repercuteixi en el rendiment del SOMCBR. A més a més, a diferència de les estratègies incrementals anteriors, aquesta estratègia ha de tenir en compte el *feedback* de l'usuari. Per tant, l'estratègia serà semi-supervisada.

8.2 Estratègia incremental i semisupervisada pel manteniment de la memòria de casos organitzada amb SOM

Les accions a realitzar dins de l'estratègia de manteniment de la memòria de casos clusteritzada han de definir-se a dos nivells. D'una banda, decidir què fer amb els nous casos resolts. D'altra banda, actualitzar els vectors directors que organitzen la memòria. A més a més, aquestes accions han de realitzar-se de manera incremental i semisupervisada. Incremental perquè es vol evitar reentrenar el mapa des de zero. Semisupervisada perquè al marge d'aprofitar el *feedback* de l'expert respecte si s'ha classificat bé el cas (punt de vista supervisat), es volen aprofitar les capacitats d'autoorganització de SOM (punt de vista no supervisat).

El resultat de la fase de revisió pot separar-se en tres escenaris: (1) classificació correcte, (2) classificació incorrecta, o bé, (3) sense classificació. En qualsevol cas, el grau d'èxit depèn directament de l'èxit del sistema en trobar els casos més adients. Suposant que la mètrica de comparació (funció de similitud) entre casos és correcte, els motius pels quals els casos més rellevants no s'han recuperat es poden dividir en:

- No hi ha casos similars a la memòria de casos.
- Hi ha casos incerts i/o amb soroll que distorsionen l'organització.
- El/s clúster/s que tenen els cas/os similars no ha/n estat seleccionat/s.

Els punts següents descriuen com detectar cadascuna d'aquestes situacions, així com les accions a emprendre en els dos nivells per tal d'abordar la problemàtica i permetre una futura correcte classificació. A més a més, els raonaments utilitzen els K-NN (*K-Nearest Neighbour*) casos trobats com a més similars com un nivell de confiança que indica l'èxit o el fracàs en trobar els casos més semblants respecte el cas nou. En els tres escenaris se suposa que la funció de similitud està correctament definida. Finalment, l'estratègia de manteniment final es resumeix a l'algorisme 8.1.

Introducció de coneixement nou. Pot succeir que el sistema només conegui una part del domini del problema i, conseqüentment, al preguntar per la zona desconeguda no sigui capaç de retornar un mínim número de K casos semblants. Per tant, cal afegir aquest coneixement nou i reajustar el sistema d'indexació perquè tingui en compte aquest element nou.

Tal i com es va explicar al capítol 3, SOM projecta l'espai original de les dades a un altre més reduït on les distàncies topològiques de l'espai original es mantenen en el nou espai. Aquesta propietat fa que si es realitza algun canvi en un clúster, els veïns immediats hagin de veure's també afectats per continuar garantint aquesta propietat. A més a més, aquest reajustament dels clústers pot provocar que els casos s'autoorganitzin automàticament i es moguin a un altre clúster que els representa millor.

Per tant, les accions a realitzar en aquest cas són les següents:

1. Incorporar el coneixement nou a la base de casos.
2. Associar el cas nou al clúster s que el sistema havia detectat com a més similar.
3. Actualitzar els vectors directors tenint en compte el coneixement nou, tant del clúster s com dels seus veïns [1], per tal de reajustar la indexació.
4. Autoorganitzar els casos dels clústers afectats, és a dir, analitzar si els casos es canvien a un altre clúster que els representi d'una manera més precisa.

Detecció i eliminació de casos sorollosos. Algunes vegades el sistema retorna casos que haurien de ser potencialment similars al cas nou, però resulten ser diferents. Continuant amb la suposició que la mètrica de similitud és correcte, el problema és una conseqüència del soroll del cas. Per tant, el que cal fer és esborrar els casos que confonen al sistema mitjançant un protocol que els detecti i, que al mateix temps, vetlli per no esborrar-ne accidentalment els que no ho són. Una manera de fer-ho és a través de protocols basats en donar oportunitats:

1. Per cada cas es defineix un comptador inicialitzat a 0.
2. Cada cop que un cas de la memòria es fa servir per classificar un problema nou (tant de manera correcta com incorrecta), s'incrementa el seu comptador si té la classe diferent respecte el problema nou, o bé, es torna a posar a zero si era de la mateixa classe.

[1] Es consideren 8 veïns perquè es treballa amb un mapa de topologia quadricular (vegeu la figura 3.2).

3. Si algun cas té un comptador igual que un cert valor llindar γ, aquest s'esborra.

Refinament dels vectors directors amb el coneixement existent. Tot i que els casos similars al nou problema estiguin a la base de casos, pot succeir que no siguin recuperats. Això és fruit d'una imprecisa indexació dels casos per part dels vectors directors, produït per exemple per la imprecisió o la complexitat de les dades. Per tant, en aquestes situacions no cal introduir ni esborrar coneixement, només cal refinar els vectors directors per corregir la manca de precisió.

Sigui c_i el cas nou. Sigui $\{c_1, ...c_K\}$ el conjunt dels K casos recuperats més semblants respecte c_i explorant tota la memòria de casos. Sigui M_m un dels $M \times M$ clústers del mapa 2D de SOM. Sigui $\overline{\theta_j}$ un vector de tantes posicions com clústers, on cadascuna d'elles fa referència a la pertinença entre el cas j i cada clúster M_m. Aquesta mesura pot calcular-se mitjançant el complement de la distància Euclidiana que va introduir-se al capítol 5, on N representa el nombre d'atributs.

$$similitud(c_i, m) = |1 - distància(c_i, M_m)| = \left| 1 - \sqrt{\frac{\sum_{n:1}^{N}(c(n) - M_m(n))^2}{N}} \right| \tag{8.1}$$

Si c_i és similar als K casos recuperats, la seva relació $\overline{\theta_i}$ respecte els clústers hauria de ser semblant a la relació $\overline{\theta_k}$ dels K casos respecte els clústers. Aquesta informació és important perquè pot ajudar-nos a conèixer quin clúster l'hauria d'haver indexat. El valor d'aquesta relació pot estimar-se com la mitja ponderada de $\overline{\theta_k}$ pels K casos recuperats. La mitja ha de ser ponderada perquè la relació entre c_i i els K casos és diferent i, per tant, la seva influència en l'estimació també ha de ser-ho: a major similitud entre casos, major pes. Aquest pes es calcula a partir del complement de la distància Euclidiana (vegeu l'equació 8.1). A més a més, el fet d'introduir el pes en la mitja, fa que calgui dividir pel sumatori dels pesos per normalitzar l'efecte. L'equació 8.2 mostra el càlcul de l'estimació, on el valor $\theta_i(m)$ més gran indica el clúster m que hauria de representar a c_i.

$$\theta_i(m) = \frac{\sum_{c_k \in K-NN} [similitud(c_i, c_k) \cdot \theta_k(m)]}{\sum_{c_k \in K-NN} similitud(c_i, c_k)} \ , \ \forall m \in M \times M \tag{8.2}$$

L'equació 8.3 il·lustra un exemple sobre com es calcula θ_i utilitzant 3 i 4 clústers. El vector final representa la pertinença estimada del cas c_i a cadascun dels clústers, i que el clúster 3 és qui l'hauria de representar perquè té el valor més alt ($\theta_i(3) = 0.73$).

$$\frac{(0.5\ 0.6\ 0.7) \cdot \begin{pmatrix} 0 & 0.2 & 0.4 & 0.8 \\ 0.2 & 0.3 & 0.7 & 0.5 \\ 0.3 & 0.3 & 0.3 & 0.9 \end{pmatrix}}{0.5 + 0.6 + 0.7} = (0.18\ 0.27\ 0.46\ \mathbf{0.73}) \tag{8.3}$$

Per tant, els passos a realitzar per refinar l'organització de la memòria de casos són els següents:

- Estimar el clúster s que hauria de representar el cas c_i.
- Actualitzar els vectors directors tenint en compte el coneixement nou, tant del clúster s com dels seus veïns, per tal de reajustar la indexació.
- Autoorganitzar els casos dels clústers afectats, és a dir, analitzar si els casos es canvien a un altre clúster que els representi d'una manera més precisa.

Taula 8.1: Resum de les situacions a tenir en compte a l'estratègia de manteniment de la memòria de casos, així com les accions a realitzar.

Situació	**Detecció**	**Accions a nivell de casos**	**Accions a nivell de clústers**
Coneixement Nou	No hi ha K casos similars	Afegir el cas nou	Actualitzar vectors directors i autoorganitzar casos
Casos sorollosos	Comptador del cas és igual a γ	Esborrar el cas conflictiu	–
Refinar vectors	Hi ha K casos similars	–	Actualitzar vectors directors i autoorganitzar

Algorisme 8.1: Estratègia incremental i semi-supervisada pel manteniment de la memòria de casos clusteritzada.

Sigui c_i el nou cas resolt
Sigui M_i el clúster que millor representa c_i
Sigui $\alpha(t)$ el factor d'aprenentatge de l'entrenament de SOM
Sigui $\vec{v_j}$ el vector director d'un clúster j
Per tot $c_k \in$ *el conjunt dels K-NN casos recuperats* **fer**
  Si c_k *té la mateixa classe que* c_i **llavors**
    Posa el comptador de c_k a zero
  Sinó
    Incrementa en una unitat el comptador de c_k
    Si *el comptador de* c_k *és igual que* γ **llavors**
      Esborra c_k de la memòria de casos
Si c_i *no estava classificat, o bé ho estava malament* **llavors**
  Cerca els K casos més semblants de la memòria respecte c_i que tinguin la seva mateixa classe
  Si *no hi ha K-NN casos* **llavors**
    Afegeix c_i a la memòria de casos
    Associa c_i al clúster M_i
  Sinó
    Estima quin clúster M_i hauria de representar millor c_i a través del valor més alt estimat de θ_i
  Per tot $\overline{v_j} \in$ *dels vuit veïns immediats de* M_i **fer**
    $\vec{v_j}(t+1) = \vec{v_j}(t) - \alpha(t) \cdot similitud(\vec{v_i}, \vec{v_j}) \cdot (\vec{c_i} - \vec{v_j}(t))$
  $\vec{v_s}(t+1) = \vec{v_s}(t) + \alpha(t) \cdot (\vec{c_i} - \vec{v_s}(t))$
  Per tot $c_r \in$ *casos de* M_i *i els seus vuit veïns immediats* **fer**
    Si c_r *s'ajusta millor en algun altre model* **llavors**
      Canvia l'associació de c_r al nou model

Totes les reflexions anteriors queden agrupades a la taula 8.1, on per cadascuna de les situacions on cal aplicar manteniment es descriu com detectar la situació, i com actuar en cadascun dels dos nivells. Finalment, l'algorisme 8.1 representa l'estratègia incremental i semi-supervisada definida pel manteniment de la memòria de casos clusteritzada.

8.3 Avaluació del rendiment de l'estratègia de manteniment de la memòria de casos

Aquest apartat avalua la introducció, l'eliminació i el reajustament del coneixement de manera incremental i semi-supervisada mitjançant l'estratègia de manteniment de la memòria de casos clusteritzada proposada. A més a més, s'avalua l'impacte sobre les capacitats del SOMCBR.

8.3.1 Experimentació

La finalitat de l'experimentació és mesurar l'impacte de l'estratègia de manteniment sobre el rendiment del SOMCBR, és a dir, analitzar la capacitat que té SOM per organitzar les dades de manera dinàmica a mesura que resol casos nous. Aquest dinamisme es simula a l'experimentació mitjançant un conjunt d'experiments on s'avalua el rendiment del SOMCBR des d'una situació on hi ha pocs casos i cal introduir molt coneixement, fins a una situació on hi ha molts casos i s'ha d'aprendre poc. De manera esquemàtica el procediment a seguir és el següent:

1. Sigui P un problema que té X casos.
2. S'inicialitza la memòria de casos del SOMCBR amb un percentatge dels casos (inicialment del 10%).
3. S'avalua la resta de casos i, al llarg d'aquest procés, el sistema aplica l'algorisme 8.1 per mantenir la memòria de casos.
4. Es repeteixen els passos 2 i 3 amb diferents percentatges inicials (10%, 33.3%, 50%, 66.6 % i 90%) i els corresponents percentatges de testeig (90%, 66.6%, 50%, 33.3%, 10%).

A diferència de la resta de capítols, el rendiment en aquest cas tindrà en compte la mida de la memòria de casos a més a més del percentatge d'errors i el nombre mig de casos recuperats per resoldre el problema. D'aquesta manera, es tindrà una visió completa de l'evolució del rendiment. A banda d'això, aquests paràmetres de rendiment es comparen respecte un CBR que no fa servir cap sistema d'indexació, per tal d'analitzar si el manteniment incremental i semi-supervisat manté les capacitats del SOMCBR. La configuració dels dos sistemes és la següent:

- La funció de distància utilitzada tant per la construcció dels models, com per la comparació entre clústers i casos és la del complement de la funció Euclidiana (vegeu l'equació 5.1).
- La mida del mapa és fixa a 3×3 per facilitar l'anàlisi de la seva evolució.
- Els models poden tenir diferent nombre de casos.
- La fase de recuperació es basa en seleccionar tots els casos del clúster més semblant. Es realitza aquesta configuració perquè suposem que el sistema està indexat correctament.
- La fase de recuperació retorna els 5 casos més similars (5-NN).
- La fase d'adaptació proposa la nova solució fent servir el cas recuperat més semblant.
- La fase d'emmagatzematge aplica l'algorisme 8.1.
- Cada resultat s'obté d'aplicar un *10-fold stratified cross-validation*.
- Cada configuració és la mitja de 10 llavors per tal de compensar els efectes aleatoris de la construcció dels models.
- L'agressivitat de la detecció i eliminació de casos sorollosos s'avalua per dos valors de γ: 2 (agressiu) i 5 (conservatiu).
- Totes les proves es realitzen tenint en compte diferents percentatges de casos per la fase de *train* i *test*: 10%-90%, 33.3$-66.6%, 50%-50%, 66.6%-33.3% i 90%-10%).

Finalment, totes les proves es realitzen amb *datasets* de diferents dominis i característiques provinents de l'*UCI Repository* (Asuncion i Newman, 2007) tal com descriu la taula 8.2.

Taula 8.2: Descripció dels *datasets* utilitzats per l'avaluació de l'estratègia de manteniment de la memòria de casos: Nom i codi del dataset, nombre d'instàncies, atributs i classes. Es presenten ordenats segons el nombre d'instàncies.

Codi	***Dataset***	**Inst.**	**Atrib.**	**Classes**	**Codi**	***Dataset***	**Inst.**	**Atrib.**	**Classes**
IR	iris	150	4	3	WD	wdbc	569	30	2
HE	hepatitis	155	19	2	BA	bal	625	4	3
IO	iosnoshpere	155	19	2	WB	wbcd	699	9	2
WN	wine	178	13	3	WI	wisconsin	699	9	2
WP	wpbc	198	33	2	PI	pim	768	8	2
SO	sonar	208	60	2	VE	vehicle	846	18	4
GL	glass	214	9	6	TA	tao	1888	2	2
HS	heart-statlog	270	13	2	SE	segment	2310	19	7
BP	bpa	345	6	2	WA	waveform	5000	40	3

8.3.2 Anàlisi i discussió dels resultats

Les taules 8.3 i 8.4 mostren les estadístiques de rendiment del CBR i del SOMCBR pels diferents *datasets* analitzats, i sobre les diferents configuracions de *train* i *test*. Les estadístiques inclouen: el percentatge d'error mig del CBR i del SOMCBR (en endavant $\%Err_{CBR}$ i $\%Err_{SOMCBR}$), la reducció mitja del nombre de casos utilitzats en el SOMCBR respecte el CBR (%R) i la reducció mitja de la mida de la memòria del SOMCBR respecte el CBR (%CM). A més a més, l'estratègia de manteniment s'aplica des d'una vessant agressiva (γ=2) i d'una altra més conservativa (γ=5). Finalment, l'última fila de cada taula conté el valor mig per cadascuna d'aquestes estadístiques.

Donant un cop de vista a les taules es pot veure que, de manera general, a mesura que hi ha més casos a l'inici els percentatges d'errors disminueixen i, de manera inversa, les reduccions del nombre d'operacions i la mida de la memòria de casos disminueixen perquè el sistema té més dades. A més a més, el fet d'establir una γ agressiva fa que, comparant les mitges d'ambdues taules, les reduccions d'operacions i de mida de la memòria de casos siguin millors, tot oferint percentatges d'error similars als de la política conservadora. Per tal d'extraure més detalls dels resultats, s'analitzaran les dues taules anteriors des de dos punts de vista:

Anàlisi horitzontal. En aquesta anàlisi es vol mesurar, de manera global, l'impacte del manteniment de casos per cada dataset. Per fer-ho, s'han definit les taules 8.5 i 8.6, les quals mostren per cada estratègia: (1) la mitja del $\%Err_{CBR}$, (2) la mitja del $\%Err_{SOMCBR}$, (3) si els percentatges d'error anteriors són significativament diferents amb l'aplicació d'un *t-test* amb un nivell del 95% de significància, (4) la mitja de %R, (5) la valoració de la reducció dels casos explorats de la memòria a la recuperació, (6) la mitja de %CM i (7) la valoració de la reducció de la mida de la memòria de casos. Les valoracions es presenten mitjançant el símbol •, el qual té una llegenda diferent segons es tracti de %R i %CM:

- •: Reducció baixa (x≤25% en %R − x≤10% en %CM).
- ••: Reducció mitja (25<x≤50% en %R − 10<x≤20% en %CM).
- •••: Reducció alta (50<x≤75% en %R − 20<x≤50% en %CM).
- ••••: Reducció molt alta (x>75% %R − x>50% en %CM).

Pel que a la significància dels percentatges d'error, en ambdues estratègies es defineixen dos grups de *datasets*:

- Els *datasets* BA, BP, HS, HP, IO, IR, PI, TA, WB, WD, WN, WI formarien el grup on el percentatge d'error es manté similar i hi ha una alta reducció del nombre de casos de la memòria explorats.

Taula 8.3: Resum dels percentatges d'error del CBR i del SOMCBR per les diferents configuracions de *train* i *test* aplicant γ=2. A més a més, s'inclou el percentatge de reducció del nombre d'operacions per recuperar el cas més semblant, així com la diferència de la mida de la memòria de casos del SOMCBR respecte el CBR.

	10% *train* - 90% *test*				**33.3% *train* - 66.6% *test***				**50% *train* - 50% *test***				**66.6% *train* - 33.3% *test***				**90% *train* - 10% *test***			
Codi	**CBR**	**SOMCBR**	**%R**	**%CM**	**CBR**	**SOMCBR**	**%R**	**%CM**	**CBR**	**SOMCBR**	**%R**	**%CM**	**CBR**	**SOMCBR**	**%R**	**%CM**	**CBR**	**SOMCBR**	**%R**	**%CM**
BA	25.36	27.07	85.29	69.44	21.35	18.68	83.57	28.23	21.80	16.25	82.91	11.08	21.42	16.10	82.25	5.26	19.90	16.83	81.14	0.18
BP	43.88	45.39	82.61	35.71	38.66	43.81	83.64	66.36	38.52	41.30	82.66	25.00	36.72	42.34	81.90	8.62	38.18	40.00	81.99	0.64
GL	43.37	54.72	69.23	32.14	35.20	41.42	75.00	39.44	33.77	38.39	75.47	20.00	33.05	38.12	75.35	10.56	35.58	40.38	76.04	0.52
HS	22.60	20.75	65.22	19.05	24.97	21.03	78.16	20.00	22.71	19.96	77.61	9.02	23.48	19.93	78.33	3.87	24.43	20.38	78.60	0.00
HE	24.52	20.23	62.50	6.25	21.83	18.64	70.59	8.00	22.01	20.00	73.33	8.00	20.23	19.93	75.49	3.92	20.53	16.00	76.98	0.00
IO	17.50	20.25	64.52	17.24	14.29	19.26	78.76	10.71	14.70	15.61	80.00	4.00	14.16	15.02	84.26	2.12	11.83	14.90	85.44	0.32
IR	6.28	11.10	46.67	12.50	6.16	9.12	68.00	6.00	5.82	8.39	70.27	4.05	4.51	7.12	72.45	2.02	3.42	4.14	75.37	0.00
PI	30.51	30.39	84.38	56.60	30.88	28.68	86.11	31.02	30.29	27.44	86.01	13.84	30.66	28.09	85.88	5.89	32.72	28.63	85.53	0.29
SE	8.60	28.66	88.07	68.10	4.84	8.80	80.39	8.72	4.47	6.95	81.22	3.04	3.56	6.66	82.68	0.98	2.75	6.06	85.31	0.10
SO	33.33	42.71	64.71	28.57	21.94	37.86	77.94	40.30	16.25	31.10	79.41	15.69	17.10	30.87	81.16	6.52	15.35	24.26	82.35	0.00
TA	6.50	13.00	86.41	39.11	4.84	5.15	86.71	5.39	4.44	3.91	86.70	1.91	4.04	3.58	86.83	0.48	3.70	3.12	86.75	0.00
VE	41.09	63.04	96.92	67.44	34.06	43.02	86.28	65.67	32.06	37.30	87.03	22.09	31.79	36.34	83.60	9.27	29.94	35.50	83.03	0.53
WA	26.69	21.64	88.99	63.95	26.86	21.35	80.33	21.26	26.81	21.63	84.79	7.58	26.22	22.08	83.66	4.07	26.23	21.24	83.46	0.47
WB	4.27	3.43	69.70	1.52	3.77	3.50	79.22	3.48	4.53	4.19	80.79	1.98	4.57	3.79	81.33	1.07	3.64	3.93	80.92	0.16
WD	5.40	6.34	75.93	12.96	5.34	5.80	82.26	6.99	4.85	5.32	83.79	2.76	4.93	4.18	83.99	1.05	4.10	4.63	84.57	0.00
WN	6.53	6.03	50.00	6.25	6.39	5.48	70.69	6.90	5.70	5.00	75.00	3.41	5.76	5.77	76.27	2.52	5.81	5.84	77.50	0.00
WI	4.76	2.87	67.16	3.03	4.78	3.67	79.13	3.48	3.95	3.46	80.23	1.69	3.92	3.28	80.47	0.43	5.39	4.37	81.40	0.00
WP	30.70	30.49	57.89	16.67	29.62	26.15	76.56	22.22	32.43	25.23	79.17	12.63	32.42	24.24	80.77	6.87	26.56	20.31	83.15	0.00
Mitja	21.22	24.90	72.57	30.92	18.65	20.08	79.07	21.90	18.06	18.41	80.36	9.32	17.70	18.19	80.93	4.20	17.23	17.25	81.64	0.18

Taula 8.4: Resum dels percentatges d'error del CBR i del SOMCBR per les diferents configuracions de *train* i *test* aplicant γ=5. A més a més, s'inclou el percentatge de reducció del nombre d'operacions per recuperar el cas més semblant, així com la diferència de la mida de la memòria de casos del SOMCBR respecte el CBR.

	10% train - 90% *test*				**33.3% train - 66.6% *test***				**50% *train* - 50% *test***				**66.6% *train* - 33.3% *test***				**90% *train* - 10% *test***			
Codi	**CBR**	**SOMCBR**	**%R**	**%CM**	**CBR**	**SOMCBR**	**%R**	**%CM**	**CBR**	**SOMCBR**	**%R**	**%CM**	**CBR**	**SOMCBR**	**%R**	**%CM**	**CBR**	**SOMCBR**	**%R**	**%CM**
BA	26.33	28.07	79.45	42.86	21.71	17.79	81.90	5.14	21.83	16.80	82.33	2.20	21.47	16.68	81.77	0.24	19.90	16.83	81.14	0.00
BP	44.33	45.33	71.88	48.39	39.48	41.62	78.38	6.25	38.47	41.33	80.92	2.87	36.72	41.44	81.47	0.00	38.18	39.70	81.99	0.00
GL	44.46	52.44	62.07	15.63	34.92	41.91	69.86	6.76	33.86	37.58	73.58	2.83	33.33	38.26	74.65	0.70	35.58	40.38	76.04	0.00
HS	24.82	21.07	61.54	8.00	25.69	21.99	76.14	4.55	22.94	19.53	77.61	1.49	23.48	20.16	77.78	0.00	24.43	20.38	78.60	0.00
HP	27.02	21.68	62.50	0.00	22.81	19.86	69.23	1.92	22.65	21.01	73.68	0.00	20.42	20.98	75.49	0.98	20.53	16.00	76.98	0.00
IO	18.92	20.84	63.64	3.13	14.41	18.68	78.07	1.75	15.04	15.73	80.11	1.14	14.16	15.52	83.83	0.85	11.83	14.90	85.44	0.00
IR	6.50	12.30	46.67	0.00	6.16	8.36	66.00	1.96	5.82	8.39	70.27	0.00	4.51	6.52	72.45	0.00	3.42	4.14	75.37	0.00
PI	33.01	30.76	74.32	16.90	31.51	29.71	84.38	5.08	30.34	28.00	85.27	1.55	30.74	28.28	85.49	0.39	32.72	28.76	85.53	0.00
SE	8.83	15.96	77.23	12.56	4.88	8.06	79.64	1.17	4.52	6.97	80.89	0.35	3.57	6.56	82.62	0.00	2.75	6.06	85.31	0.00
SI	34.19	42.85	52.63	15.79	22.16	36.99	73.53	4.35	16.25	31.55	78.64	0.97	17.10	29.86	80.43	0.72	15.35	24.26	82.35	0.00
TA	7.05	9.75	83.51	5.85	4.88	5.27	86.41	0.47	4.44	4.07	86.61	0.00	4.04	3.63	86.75	0.00	3.70	3.12	86.75	0.00
VE	40.97	53.43	79.27	64.56	34.36	39.37	82.69	9.57	32.18	37.19	85.88	2.58	31.83	36.55	83.24	0.36	29.94	35.50	83.03	0.00
WA	28.81	22.69	80.04	9.16	27.46	22.83	77.49	2.34	26.98	22.52	81.26	0.96	26.26	22.64	83.08	0.36	26.21	21.24	83.42	0.00
WB	4.70	4.06	69.12	0.00	3.94	3.85	79.22	1.30	4.53	4.07	80.79	0.56	4.57	3.66	81.33	0.43	3.64	3.93	80.92	0.00
WD	5.73	7.28	74.55	7.27	5.36	5.55	82.26	1.08	4.85	5.07	83.45	0.34	4.93	4.07	83.99	0.26	4.10	4.63	84.57	0.00
WN	6.97	7.75	52.94	11.11	6.39	6.71	70.69	3.39	5.70	5.79	75.00	1.14	5.76	5.77	76.27	0.84	5.81	5.84	77.50	0.00
WI	5.25	3.40	67.65	1.49	4.95	3.94	79.22	1.72	4.09	3.43	80.23	0.56	3.92	3.32	80.47	0.21	5.39	4.37	81.40	0.00
WP	32.55	32.00	52.63	0.00	30.00	25.06	75.38	6.15	33.23	25.63	78.35	3.06	32.58	24.55	80.92	0.76	26.56	20.31	82.58	0.00
Mitja	22.25	23.98	67.31	14.59	18.95	19.86	77.25	3.61	18.21	18.59	79.72	1.26	17.74	18.25	80.67	0.39	17.22	17.24	81.61	0.00

Taula 8.5: Taula resum de les mitges de les cinc configuracions per l'estratègia agressiva ($\gamma = 2$) de la taula 8.3. La taula inclou una valoració de la significància dels resultats amb el *t-test*, i del grau de reducció en els casos de la memòria explorats i la mida de la memòria.

Codi	$\overline{\%Err_{CBR}}$	$\overline{\%Err_{SOMCBR}}$	**t-test**	$\overline{\%R}$	**Valoració**	$\overline{\%CM}$	**Valoració**
BA	21.97	18.99	-	83.03	●●●●	22.84	●●●
BP	39.19	42.57	-	82.56	●●●●	27.27	●●●
GL	36.19	42.61	↓	74.22	●●●	20.53	●●●
HS	23.64	20.41	-	75.58	●●●●	10.39	●●
HE	21.82	18.96	-	71.78	●●●	5.23	●
IO	14.50	17.01	-	78.60	●●●●	6.88	●
IR	5.24	7.97	-	66.55	●●●	4.91	●
PI	31.01	28.65	-	85.58	●●●●	21.53	●●●
SE	4.84	11.43	↓	83.53	●●●●	16.19	●●
SO	20.79	33.36	↓	77.11	●●●●	18.22	●●
TA	4.70	5.75	-	86.68	●●●●	9.38	●●
VE	33.79	43.04	↓	87.37	●●●●	33.00	●●●
WA	26.56	21.59	↓	84.25	●●●●	19.47	●●
WB	4.16	3.77	-	78.39	●●●●	1.64	●
WD	4.92	5.25	-	82.11	●●●●	4.75	●
WN	6.04	5.62	-	69.89	●●●●	3.82	●
WI	4.56	3.53	-	77.68	●●●●	1.73	●
WP	30.35	25.28	↓	75.51	●●●●	11.68	●●
Mitja	18.57	19.77		78.91	●●●	13.30	●●

Taula 8.6: Taula resum de les mitges de les cinc configuracions per l'estratègia agressiva ($\gamma = 5$) de la taula 8.4. La taula inclou una valoració de la significància dels resultats amb el *t-test*, i del grau de reducció en els casos de la memòria explorats i la mida de la memòria.

Codi	$\overline{\%Err_{CBR}}$	$\overline{\%Err_{SOMCBR}}$	**t-test**	$\overline{\%R}$	**Valoració**	$\overline{\%CM}$	**Valoració**
BA	22.25	19.23	-	81.32	●●●●	10.09	●●
BP	39.44	41.88	-	78.93	●●●●	11.50	●●
GL	36.43	42.11	↓	71.24	●●●	5.18	●
HS	24.27	20.63	-	74.33	●●●	2.81	●
HE	22.69	19.91	-	71.58	●●●	0.58	●
IO	14.87	17.13	-	78.22	●●●●	1.37	●
IR	5.28	7.94	-	66.15	●●●	0.39	●
PI	31.66	29.10	-	83.00	●●●●	4.78	●
SE	4.91	8.72	↓	81.14	●●●●	2.81	●
SO	21.01	33.10	↓	73.52	●●●	4.37	●
TA	4.82	5.17	-	86.01	●●●●	1.26	●
VE	33.86	40.41	↓	82.82	●●●●	15.41	●●
WA	27.14	22.38	↓	81.06	●●●●	2.56	●
WB	4.28	3.91	-	78.28	●●●●	0.46	●
WD	4.99	5.32	-	81.76	●●●●	1.79	●
WN	6.13	6.37	-	70.48	●●●	3.30	●
WI	4.72	3.69	-	77.79	●●●●	0.80	●
WP	30.98	25.51	↓	73.97	●●●	1.99	●
Mitja	18.87	19.59		77.31	●●●	3.97	●

- Els *datasets* GL, SE, SO, VE, WA i WP són *datasets* on el SOMCBR obté un error més alt, tot i que hi ha una alta reducció del nombre de casos.

Hi ha alguna relació entre la formació d'aquests grups i el seu comportament? Sí. Tal com es va analitzar al capítol 4, el rendiment del SOMCBR està vinculat a la naturalesa de les dades, concretament, a la capacitat dels clústers per representar la seva topologia. En aquest cas, el primer grup de *datasets* pertanyen a la zona de l'espai de complexitat B i C, i els del segon grup a la zona de l'espai A i B. Per tant, les capacitats del SOMCBR es mantenen vigents segons la topologia dels *datasets* amb el manteniment de la memòria aplicant l'estratègia.

Fins aquí, amb les dues estratègies s'està obtenint el mateix rendiment. La diferència d'aplicar-les es pot observar a les columnes 7 i 8, les quals fan referència a la mida de la memòria de casos. En aquest cas, pot veure's que l'estratègia agressiva permet disposar d'una memòria més reduïda ja que els casos que introdueixen soroll o incertesa són eliminats.

Per tant, l'estratègia de manteniment de la memòria de casos funciona correctament des d'aquest punt de vista ja que continua mantenint les capacitats del SOMCBR. A més a més, l'estratègia agressiva ens ofereix un rendiment similar, tot reduint la mida de la memòria.

Anàlisi vertical. Aquest punt de vista avalua com evolucionen en mitja els cinc estats inicials a partir dels quals s'aplica l'estratègia de manteniment. Per fer-ho es construeixen dues gràfiques on es representa l'evolució de la mitja dels paràmetres del rendiment indicats a les taules 8.3 i 8.4 respectivament: $\overline{\%Err_{CBR}}$ (+), $\overline{\%Err_{SOMCBR}}$ (●), $\overline{\%R}$ (□) i $\overline{\%CM}$ (△).

Tal com mostren les gràfiques (a) i (b) de la figura 8.1, el $\%Err_{CBR}$ i el $\%Err_{SOMCBR}$ evolucionen pràcticament de la mateixa manera a les dues estratègies. L'únic instant on hi ha una petita diferència és a l'estratègia agressiva quan hi ha una inicialització de la memòria amb només un 10% dels casos. En aquesta configuració, hi ha més errors a l'haver-hi menys coneixement base i, per tant, l'agressivitat del mètode pot fer que alguns casos interessants siguin esborrats accidentalment perquè la indexació no ha estat la correcta al no haver-hi un mínim de casos representatius. No obstant, els errors no són significativament diferents. (21.22 i 24.90 respectivament).

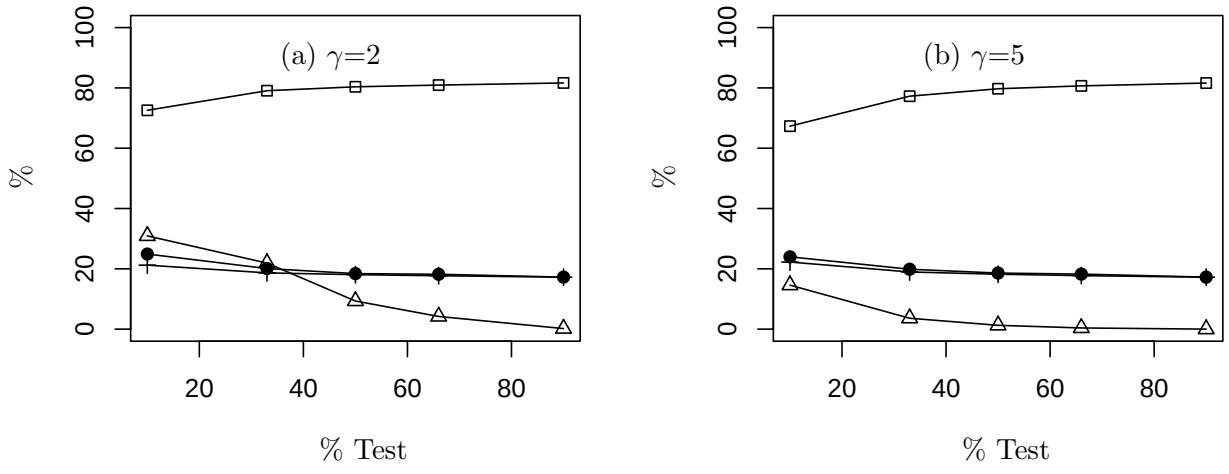

Figura 8.1: Les gràfiques mostren l'evolució del rendiment de les dues estratègies de manteniment segons la seva agressivitat de manera global pels *datasets* (γ igual a 2 i 5): (1) $\overline{\%Err_{CBR}}$ (+), (2) $\overline{\%Err_{SOMCBR}}$ (●), (3) $\overline{\%R}$ (□) i (4) $\overline{\%CM}$ (△).

Les gràfiques també mostren la relació entre la mida de la memòria ($\triangle$) i els casos que s'exploren de la memòria a la fase de recuperació ($\square$) a mesura que hi ha més dades. En aquest cas, la relació no és tan constant com abans ja que la inicialització de la memòria condiciona els dos paràmetres. Lligant amb el punt de vista anterior, l'estratègia agressiva tendeix a reduir més la memòria a les configuracions amb poques dades, ja que és més fàcil cometre errors perquè SOM potser no està organitzat de manera òptima si les dades no són prou representatives. En canvi, a partir de la situació on s'inicialitza la memòria amb el 50% dels casos, les dues estratègies tenen un comportament pràcticament igual.

Per tant, l'estratègia de manteniment de la memòria de casos funciona correctament també des d'aquest punt de vista.

8.4 Conclusions i línies futures

El capítol ha presentat l'estratègia incremental i semi-supervisada per realitzar el manteniment del coneixement emmagatzemat a la memòria de casos clusteritzada amb SOM. Aquest objectiu aborda dos fronts a resoldre a partir del *feedback* de l'expert. D'una banda, introduir o esborrar casos. D'altra banda, reorganitzar dinàmicament SOM. L'aspecte més conflictiu d'aquests dos era el segon perquè SOM no permet la gestió dinàmica del seu coneixement. Tot i que hi ha variants que ho resolen, aquestes sempre ho fan des d'un punt de vista no supervisat. Per aquest motiu, s'ha presentat una proposta d'estratègia que actua de manera conjunta sobre els dos fronts a partir dels conceptes de les estratègies de manteniment convencionals. L'estratègia aprofita el *feedback* de l'expert (punt de vista supervisat) i la capacitat dels casos per autoorganitzar-se (punt de vista no supervisat).

La validació de l'estratègia s'ha fet a partir de l'estudi de l'evolució de la memòria de casos clusteritzada a mesura que parteix d'un nombre diferent de casos sobre diferents *datasets*. A més a més, aquesta evolució s'ha contrastat respecte l'evolució d'una memòria de casos sense indexació amb la finalitat d'avaluar el grau de precisió amb el qual els índexs són reajustats.

L'estudi dels resultats s'ha fet des de dos punts vista. D'una banda, s'ha analitzat l'evolució del rendiment de manera individual sobre els *datasets*. D'altra banda, s'ha analitzat l'evolució dels estats de la memòria de manera global per tots els *datasets*. Els resultats han estat satisfactoris perquè l'estratègia de manteniment no altera les propietats de SOMCBR. A més a més, s'han reafirmat les conclusions extretes al capítol 4 respecte les situacions on l'organització amb SOM és exitosa. La contribució d'aquest capítol es troba publicada a l'article següent:

- A. Fornells i E. Golobardes. *Case-base maintenance in an associative memory organized by a Self-Organizing Map*. Al llibre *Innovations in Hybrid Intelligent Systems*, volum 44. Editors: E. Corchado, J.M. Corchado, i A. Abraham. Springer-Verlag, 2007. En impremta.

Les línies futures es poden dividir en dues línies. La primera fa referència al canvi dinàmic de l'arquitectura de la xarxa per tal d'adaptar-se millor a les dades. La segona es refereix a introduir més conceptes de les estratègies de manteniment de casos per potenciar més el seu funcionament.

Resum

El manteniment del coneixement en el CBR és un aspecte molt important perquè condiciona el seu rendiment futur. Això fa que el manteniment del seu coneixement sigui vital. El problema que apareix en el nostre cas és que SOM no permet la gestió dinàmica del seu coneixement. D'aquesta manera, l'aplicació de qualsevol de les estratègies convencionals de manteniment implicaria sempre haver de reentrenar el mapa des de zero, amb la conseqüent despesa computacional.

L'estratègia proposada defineix una actuació simultània sobre els dos nivells de dades que hi ha, casos i vectors directors dels clústers, per tal de permetre: (1) introduir coneixement nou, (2) detectar i esborrar casos sorollosos, i (3) reajustar el sistema d'indexació. Així doncs, s'obté un sistema incremental i semi-supervisat. D'una banda, s'aprofita el *feedback* de l'expert (punt de vista supervisat) i, d'altra banda, s'aprofita la capacitat per autoorgantizar (punt de vista no supervisat).

L'avaluació de l'estratègia ha constat de dues parts. A la primera s'ha estudiat l'evolució del rendiment del SOMCBR a partir de diferents configuracions inicials de la memòria de casos. A la segona, s'ha comparat el rendiment anterior respecte un sistema CBR que no fa servir cap tipus d'organització. Els resultats han demostrat que l'aplicació de l'estratègia proposada manté les virtuts del SOMCBR.

Capítol 9

Plataforma SOMCBR

Aquest capítol descriu breument els components de la plataforma SOMCBR desenvolupada al llarg de la tesi. La plataforma està caracteritzada per introduir les capacitats *Soft-Computing* dels Mapes autooganitzatius dins el CBR. Concretament, la memòria de casos s'organitza amb els clústers definits pels mapes, i les fases estan adaptades per treure el màxim profit dels clústers. Això permet reduir el temps computacional, tot mantenint les capacitats de resolució.

9.1 Disseny

Tots els aspectes tractats del capítol 5 fins al capítol 8 estan integrats sota la plataforma SOMCBR (*Self-Organizing Map in a Case-Based Reasoning system*).

Aquest apartat descriu a alt nivell els diferents mòduls que composen l'aplicació (vegeu la figura 9.1), així com les eines que s'han fet servir per desenvolupar-la. En canvi, s'ha omès la presentació dels diagrames de flux perquè a l'haver-hi tantes configuracions i paràmetres, el capítol creixeria innecissàriament sense aportar res al lector.

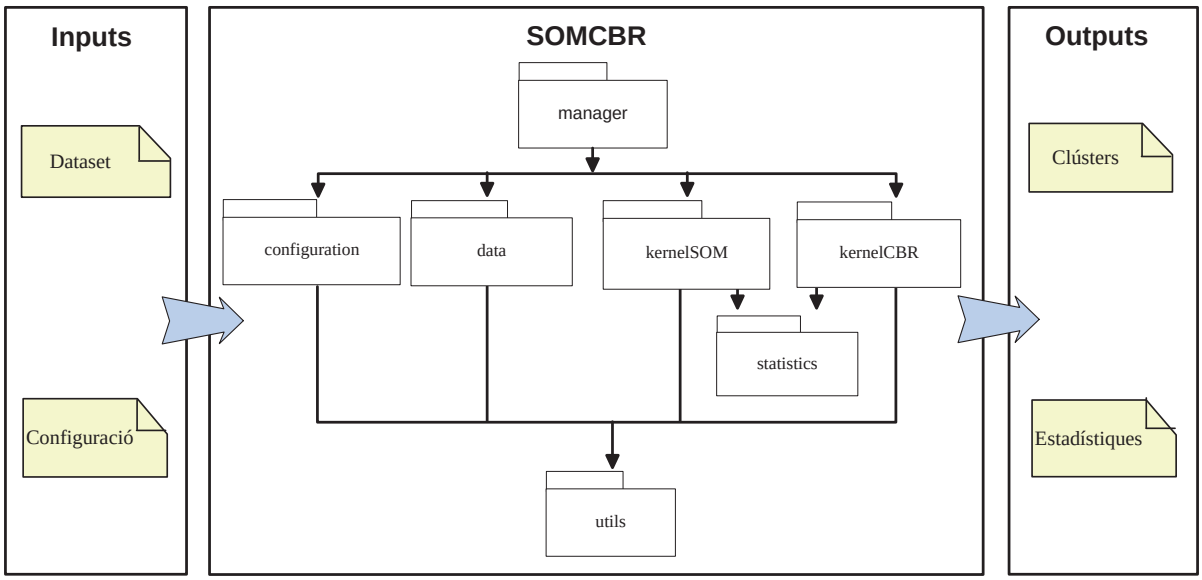

Figura 9.1: Diagrama de blocs general de la plataforma SOMCBR.

9.1.1 Elements d'*Input* i *Output*

A la figura 9.1 es distingeixen dos tipus diferents de fitxers d'*inputs*:

- **Dataset.** És l'especificació del problema que es vol resoldre.
- **Configuració.** Defineix el comportament de cadascuna de les fases del SOMCBR, així com el format que cal donar als resultats.

L'execució del sistema proporciona dos *outputs* en format de fitxer:

- **Clústers.** Conté la descripció dels models generats per SOM. La descripció consisteix en la definició del vector director del model, així com els casos que té associats.
- **Estadístiques de classificació.** Representa el % d'encerts, el % d'error, el % de no classificats, el % de sensitivitat, i el % d'especificitat. Si s'executa en mode *Cross-Validation* també es generen les mitges i desviacions estàndards associades. A més, calcula el temps mig de cadascuna de les fases del CBR.

9.1.2 Especificació dels mòduls

En aquest apartat es defineixen les finalitats i les funcionalitats de cadascun dels mòduls que integren la plataforma SOMCBR.

9.1.2.1 Mòdul *manager*

És el mòdul principal del sistema, i s'encarrega de coordinar la resta de mòduls. En funció dels paràmetres de configuració, ja sigui per fitxer o per línia de comandes, instancia els elements necessaris per activar l'organització de la memòria de casos amb SOM, així com de configurar quina de les variants de les fases del CBR proposades al llarg de la tesi s'ha de fer servir en cada pas.

9.1.2.2 Mòdul *configuration*

La finalitat d'aquest mòdul és gestionar la configuració de tots els paràmetres del sistema. Les seves principals tasques són:

- Carregar/guardar la informació de configuració emmagatzemada en un fitxer en format XML (*eXtensible Markup Language*) (WebXML, 2007).
- Modificar la configuració de fitxer per línia de comandes.
- Proporcionar a la resta de mòduls els paràmetres de la seva configuració.

9.1.2.3 Mòdul *data*

El mòdul de dades gestiona les dades que representen el problema del domini sobre el qual es treballa. Les seves principals tasques són:

- Carregar les dades del problema emmagatzemades en el format estàndard ARFF (Witten i Frank, 2000).
- Aplicar sobre les dades algunes de les operacions de preprocessament següents:
  (a) Normalització per rang, diferència i escalat decimal.

(b) Ponderació d'atributs mitjançant PCA o Correlació Mostral.

(c) Selecció de característiques.

(d) Gestió de valors desconeguts.

(e) Correcció d'atributs erronis fora de rang.

9.1.2.4 Mòdul *kernelCBR*

El mòdul kernelCBR té l'objectiu de simular el cicle de vida del CBR mitjançant la coordinació de les diferents fases. Les funcionalitats que implementa per cada element i fases són:

- Accés i manipulació de la memòria de casos del CBR:

- Recuperació dels K casos més similars a partir d'un cert llindar (paràmetre) de confiança, i d'una funció de similitud (Minkowski (r=1, 2, 3), Manhattan, Clark, Cosinus, funcions heterognenies, etc.)

- Proposta d'una solució a partir de la votació dels K casos recuperats.

- Acceptació de la solució proposta tenint en compte criteris de mínima semblança.

- Emmagatzematge dels nous casos resolts segons una de les polítiques següents:

  (a) No guardar el nou cas.

  (b) Guardar el nou cas si s'ha fallat.

  (c) Guardar el nou cas si s'ha resolt correctament, però és molt diferent als que hi ha guardats.

9.1.2.5 Mòdul *kernelSOM*

El mòdul kernelSOM conté les aportacions realitzades a la tesi a partir de SOM. Les funcionalitats que implementa són:

- Gestió del mapa:

  (a) Definició automàtica del nombre de clústers.

  (b) Definició de mètriques per comparar els elements respecte els clústers.

  (c) Definició dels factors d'aprenentatge i veïnatge.

  (d) Còmput de l'error del mapa.

- Recuperació dels clústers basada en dos nivells:

  (a) Selecció del nombre de clústers.

  (b) Selecció del nombre de casos a recuperar dels clústers prèviament seleccionats.

- Proposta de solució a partir de probabilitats computades a partir dels clústers.

- Revisió dels resultats fent servir tècniques de *Relevance Feedback*.

- Integració de nou coneixement en els clústers.

9.1.2.6 Mòdul *statistics*

És el mòdul encarregat de recopilar la informació de les execucions per tal de generar els informes de resultats. Les seves tasques són:

- Recopilar informació sobre l'execució actual:
  (a) Càlcul del % d'encerts, % d'errors i % dels no classificats.
  (b) Càlcul del % de sensitivitat i % d'especificitat si és possible.
  (c) Temps parcials de cada fase i temps total.
- Generació d'informes per pantalla o a fitxer.

9.1.2.7 Mòdul *utils*

El mòdul *utils* es composa per un conjunt de llibreries de propòsit general que implementen funcionalitats habituals realitzades per la resta de mòduls. Les llibreries que engloba són:

- Accés de fitxers.
- Control de temps.
- Gestió de nombres aleatoris.
- Conversions entre tipus de dades.
- Llibreria per tractar amb vectors i matrius.

9.2 Implementació i eines de desenvolupament emprades

El disseny de l'apartat anterior permet separar clarament les funcionalitats de cadascuna de les diferents parts que componen la plataforma. Per tal de permetre una fàcil integració dels mòduls i la incorporació de futures funcionalitats, totes les parts del sistema es relacionen mitjançant interfícies. Això permet separar de manera independent les funcionalitats que ha d'acomplir el mòdul, respecte de com s'han implementat.

La plataforma SOMCBR s'ha desenvolupat usant el llenguatge de programació C++ per dos motius. D'una banda, per fer servir els conceptes de modularitat implícits de l'orientació a objecte. D'altra banda, perquè C++, al ser un llenguatge compilat, permet una ràpida execució respecte altres llenguatges com Java. La plataforma està composada per 75 classes.
Les eines que s'han fet servir són:

- Anàlisi i disseny
  - Borland Together 6.1. Eina que permet realitzar el procés d'anàlisi i disseny mitjançant la metodologia UML (Together, 2007).
- Implementació de la plataforma SOMCBR
  - Lex/Yacc. Llibreries per la construcció de parsers mitjançant C/C++ (LexYacc, 2007).
  - Eclipse 3.2 amb plug-in C/C++. Entorn multiplataforma de desenvolupament (Eclipse, 2007).
  - Cygwin. És un entorn semblant a linux per Windows que inclou un compilador de C++, i un debugger (Cygwin, 2007).

Resum

El capítol ha presentat a alt nivell els mòduls de la plataforma SOMCBR desenvolupada en aquesta tesi. La seva principal virtut és que permet introduir les capacitats de *Soft-Computing* i de *Knowledge Discovery* de SOM dins el CBR, seguint les estratègies explicades als capítols anteriors. La plataforma ha estat implementada en C++ a partir d'eines de desenvolupament de lliure distribució i de llicències universitàries.

Com s'ha anat veient, la plataforma ha permès abordar amb èxit els diferents reptes plantejats als capítols anteriors.

Capítol 10

Aplicacions de la recerca

Part de les aportacions de la tesi s'han aplicat a dominis específics fruït dels projectes on ha estat emmarcada. Aquestes capítol presenta la utilització de SOMCBR com a motor de cerca per permetre la interacció dels experts en temps reals al projecte HRIMAC, i l'aplicació de SOM com a algorisme de clustering per estudiar les relacions entre les dades al projecte ANALIA.

10.1 Introducció

Aquest capítol presenta dues línies de treball dutes a terme dins els projectes HRIMAC i ANALIA, les quals han esdevingut una aplicació dels coneixements adquirits amb SOM.

Exploració de la memòria segons la subjectivitat de l'expert. El CBR recupera la informació més semblant en base a un conjunt de criteris determinats per un expert. Tot i que la seva definició ha de ser objectiva, sovint aquesta està condicionada per l'experiència de l'expert, la qual li proporciona una percepció pròpia de la realitat. A més a més, si el domini és complex i incert les possibilitats que dos experts vegin coses diferents a partir del mateix exemple s'incrementen notablement. Per aquest motiu es fa necessari disposar de mecanismes que permetin traslladar la subjectivitat de l'usuari al procés de cerca, per poder dirigir-la segons els seus criteris.

Aquest punt presenta l'aplicació d'estratègies basades en el *Relevance Feedback* (vegeu l'apèndix H) al projecte HRIMAC per permetre als experts explorar l'espai de mamografies en base a les seves preferències i percepcions de la realitat. Aquesta interacció entre l'expert i el sistema ha de ser en temps real. És aquí on la capacitat per indexar la informació de SOM es fa necessària per explorar la memòria de casos en base als criteris de l'usuari. A més a més, com que el domini de càncer de mama és complex i incert, les capacitats *Soft Computing* seran molt útils per tractar aquest tipus de coneixement.

Detecció de les vulnerabilitats d'una xarxa telemàtica. La seguretat telemàtica ha esdevingut un tema de preocupació en les xarxes corporatives. Dins d'aquest context, els testejos de seguretat s'han convertit en una eina fonamental per detectar vulnerabilitats que puguin comprometre la seguertat de la xarxa. La gran dificultat en aquest camp és la gran quantitat d'informació que cal analitzar, així com a la manca d'un estàndard que indiqui la informació sobre la que centrar-se.

Dins d'aquest marc de treball, SOM s'ha utilitzat com una tècnica de clustering per analitzar les dades del projecte ANALIA (vegeu l'apèndix C), i intentar definir agrupacions de dispositius amb vulnerabilitats similars. A més a més, el seu ús ha contribuït al desenvolupament de mesures per avaluar la cohesió dels dispositius en els clústers, és a dir, analitzar si tenen les mateixes vulnerabilitats.

10.2 Integració del *Relevance Feedback* a l'HRIMAC

La finalitat del projecte HRIMAC era implementar una eina per a la recuperació d'imatges a través de l'anàlisi del contingut. Tal com es comenta a l'apèndix B, les imatges mamogràfiques que es fan servir en el projecte HRIMAC procedeixen d'un conjunt de pacients de l'Hospital Universitari Dr. Josep Trueta de Girona i, d'altra banda, de bases públiques de mamografies. Un cop les dades són preprocessades i digitalitzades pel departament de Visió per Computador i Robòtica de la Universitat de Girona, són integrades posteriorment en un nucli CBR desenvolupat pel GRSI.

Un dels grans problemes de realitzar cerques sobre aquestes dades tan complexes és la facilitat amb la qual el sistema pot desviar-se del que realment ha de buscar. En el cas de l'HRIMAC, els experts realitzen dos tipus de cerques:

1. Cerques per detectar anormalitats o lesions espiculars en forma de microcalcificació, per analitzar la seva naturalesa benigne o maligne.

2. Cerques per determinar la densitat del teixit, per millorar la interpretació mamogràfica.

Les tècniques de *Relevance Feedback* (vegeu l'apèndix H) permeten incorporar la percepció dels resultats de l'expert en el procés de recuperació, en altres paraules, redirigir la cerca. En el nostre cas, l'estratègia de *Relevance Feedback* ha de tenir les següents propietats: treballar amb característiques de baix nivell (imatges processades), *feedbacks* positius i negatius, cerca categòrica, i consulta per exemples. A més, com que les percepcions dels experts són molt subjectives, el sistema no ha de modificar el seu comportament quan els experts interactuin amb ell. El procés s'integraria de la següent manera:

1. L'expert realitza una consulta sobre la base de dades de mamografies.

2. El nucli CBR cerca en la memòria de casos els més similars en base als criteris de cerca.

3. L'expert marca les imatges que creu que són més rellevants segons la seva experiència.

4. El sistema a partir del *feedback* retornat, reajusta la consulta i repeteix la cerca.

És molt important tenir present que tot aquest procés es fa en temps real i, per tant, ha de ser el més ràpid possible per no fer esperar a l'expert, i provocar que deixi de fer servir l'eina. El temps de resposta del sistema amb els nous resultats depèn directament del temps de la fase de recuperació del CBR, ja que cal comparar amb totes les instàncies de la memòria de casos. Reduir el temps de resposta equival a reduir el temps de la fase de recuperació. És necessari aplicar alguna estratègia per agrupar o indexar la informació i d'aquesta manera reduir el nombre d'operacions de comparació realitzades per trobar la informació. És en aquest context on l'aplicació del SOMCBR apareix per millorar el temps de resposta del CBR. A més a més, les seves capacitats *Soft-Computing* i de *Knowledge Discovery* han demostrat als capítols anteriors que permeten gestionar bé aquest domini.

No és el primer cop que SOM es fa servir dins del context de les tècniques de *Relevance Feedback*. Els treballs de Zhang (Zhang i Zhong, 1995) són els primers en utilitzar-la com eina d'indexació d'imatges, on SOM s'utilitza per filtrar imatges segons el color i la textura. Més endavant, Han i Myaeng (Han i Myaeng, 1996) el va utilitzar per delimitar objectes. També s'ha utilitzat pel desenvolupament d'eines Web per la cerca d'imatges (PicSOM (Laaksonen et al., 1999)) o documents (WEBSOM (Kaski et al., 1998a)), o per delimitar zones o regions en les imatges (ASPECT (Csillaghy et al., 2000)).

Proposta d'estratègia

La introducció del *Relevance Feedback* dins de l'HRIMAC consisteix en permetre a l'expert llençar consultes de manera iterativa contra el SOMCBR a partir d'un conjunt de casos, els quals representen les peticions a partir de les quals vol fer l'exploració. Aquest procés es reflecteix a la figura 10.1 i a l'algorisme 10.1. Donada la imatge inicial del pacient a estudiar, el sistema retorna el conjunt de casos més semblants. A partir d'això, l'expert marca les imatges que considera rellevants i irrellevants. El sistema retorna el conjunt d'imatges més semblants respecte cada imatge d'entrada, tot evitant el retorn d'imatges descartes per l'expert com irrellevants. El procés finalitza quan l'expert determina que la informació que ha trobat és la que buscava.

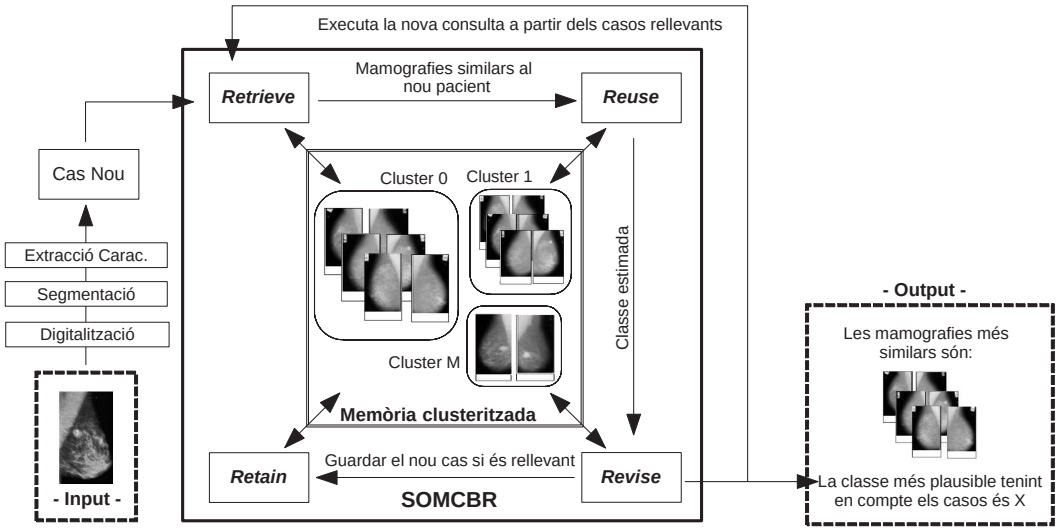

Figura 10.1: Representació gràfica de la interacció entre l'expert i el sistema.

Algorisme 10.1: Estratègia de *Relevance Feedback* basada en el SOMCBR per l'HRIMAC.

Sigui R el conjunt de casos recuperats del SOMCBR
Sigui $M_{i,j}$ un model
Sigui D^+ el conjunt d'elements rellevants i D^- el conjunt d'elements no rellevants
Sigui E_0 l'element d'avaluació inicial
Sigui E un element d'avaluació
Sigui L una llista que manté els elements ordenats creixement
Sigui X el nombre de models a tenir en compte
$D^-=\emptyset$; $D^+=\{< E_0 >\}$; L=$\emptyset$
Per tot *E de* D^+ **fer**
  Per tot $M_{i,j}$ *de M* **fer**
    Càlcul de la similitud d'E respecte $N_{i,j}$
    Guarda en L la relació $< M_{i,j}, similitud >$
  Per tot *E continguts en els X primers nodes de L* **fer**
    Si $E \notin D^-$ **llavors**
      Guarda E en R
  Tots els elements de R es mostren l'usuari
  Si *usuari troba el que busca* **llavors**
    Finalitza execució
  Sinó
    L'usuari marca els elements
    D^+=<Elements rellevants>
    $D^-=D^- \cup$ <Elements no rellevants>

Avaluació de la proposta

És difícil avaluar de manera quantitativa la millora d'integrar un mòdul de *Relevance Feedback* a l'HRIMAC. No existeix un *benchmarking* o criteri estàndard perquè l'avaluació està complementament relacionada amb el domini, la seva complexitat, i els punts de vista de l'expert. Tanmateix, la introducció de l'expert en el procés de recuperació permet dirigir la cerca segons les seves percepcions i, per tant, la seva integració pot considerar-se qualitativament com a positiva. En canvi, si es pot avaluar de manera quantitativa l'impacte de fer servir SOMCBR enlloc del CBR en termes de nombre d'operacions.

Per modelar de manera teòrica el nombre d'operacions que es realitzen durant una consulta partirem de les següents suposicions:

1. Es disposa de CM exemples.
2. La memòria de casos del SOMCBR disposa d'un mapa de $M \times M$.
3. Els models tenen una quantitat similar d'elements mapejats.
4. L'usuari fa sempre I interaccions.
5. En cadascuna de les iteracions marca D elements com rellevants.
6. Parteix sempre d'un única imatge inicialment.

Les equacions 10.1 i 10.2 modelen el nombre d'operacions a realitzar en la fase de recuperació sense i amb l'estratègia explicada a l'apartat anterior. Amb una organització plana, per cada interacció I el sistema ha de buscar en tota la memòria de casos (CM elements) els D elements que l'usuari ha seleccionat com a rellevants cada cop. En canvi, amb SOM el sistema només recupera els casos dels models en els que l'element a buscar es correspon en cada interacció, que són els X models més similars. El paràmetre X gradua l'agressivitat del mètode, on el cas extrem de fer servir tots els models equival a l'esquema de no fer servir el mapa.

$$\#operacions = (1 + D \cdot (I - 1)) \cdot CM \tag{10.1}$$

$$\#operacions = (1 + D \cdot (I - 1) \cdot (M^2 + X \cdot \frac{CM}{M^2})) \tag{10.2}$$

Les gràfiques de la figura 10.2 mostren l'evolució del nombre d'operacions a partir de les equacions 10.1 i 10.2 i suposant els valors per I (5) i D (3 i 5). A més a més, s'han realitzat els càlculs per dues mesures del mapa. Un mapa petit (2×2) i un altre més gran (8×8). Si de per si SOMCBR redueix dràsticament el nombre d'operacions, en aquest cas això s'accentua més perquè es realitzen moltes consultes al llarg de l'interacció.

D'altra banda, està l'aspecte que fa referència a la precisió de SOMCBR en trobar els casos més semblants. Aquest aspecte no cal que es torni a tractar, ja que s'ha avaluat de manera satisfactòria als capítols de la fase recuperació i revisió mitjançant els *datasets* del projecte HRIMAC. Per tant, els avantatges que SOMCBR ens aporta són evidents: Més precisió i més rapidesa.

10.3 Detecció de les vulnerabilitats d'una xarxa telemàtica

Quan es realitza un test de seguretat en una xarxa a priori no se sap quins resultats s'obtindran. Fins i tot, un cop obtinguts els resultats les conclusions extretes poden ser diferents depenent de la situació de cada xarxa. Això ha fet que hagi sorgit la necessitat de desenvolupar eines que donin suport als experts de seguretat a prendre les decisions.

L'objectiu del projecte ANALIA és aplicar tècniques de la intel·ligència artificial i de la mineria de dades als resultats recopilats pel sistema CONSENSUS per tal de millorar-ne la seva anàlisi. El sistema CONSENSUS és una eina desenvolupada entre la URL i l'empresa ISECOM que té com a finalitat la de recopilar informació d'una xarxa aplicant un conjunt de testejos de seguretat seguint la metodologia OSSTMM (*Open Source Security Testing Methodology Manual*).

La motivació per aplicar l'aprenentatge artificial és fruït del gran volum de dades que CONSENSUS recull, i que es veu agreujat per la gran diversitat de factors que hi són presents en aquest tipus d'entorns (dades malicioses, tràfic no autoritzat, patrons d'intrusió, etc.). Un altre problema afegit és que el resultat dels tests pot contenir dades incertes, incompletes o poden presentar soroll. Per exemple, quan es testeja una màquina per a detectar quins serveis té disponibles, certes eines de testeig poden detectar un port obert, mentre que unes altres eines de testeig pot ser que no detectin aquest port obert. Per tant, l'aplicació de tècniques de *data mining* per clusteritzar la informació pot ajudar a l'expert a agilitzar les seves anàlisis (Corral et al., 2005b).

Dins del ventall de tècniques de clustering, ANALIA es centra en aquelles que provenen d'una vessant no supervisada perquè els dispositius de la xarxa no tenen cap etiqueta que indiqui cap classe. Concretament, s'ha estudiat l'aplicació de K-means (Hartigan i Wong, 1979), X-means (Pelleg i Moore, 2000), Auto-Class (Cheeseman i Stutz, 1996) i SOM (Kohonen, 1984) entre d'altres sobre diferents *datasets* definits pels experts telemàtics de la URL (vegeu l'apèndix C).

En el context d'aquesta tesi, SOM s'ha aplicat satisfactòriament sobre els jocs de dades definits al projecte ANALIA (vegeu l'apèndix C). L'anàlisi dels clústers definits per SOM a través d'un expert ha demostrat la seva capacitat per agrupar els dispositius segons la similitud entre les seves vulnerabilitats. A més a més, l'aplicació de SOM ha ajudat a l'expert a la definició de criteris de cohesió específics a la telemàtica entre els clústers i dins del propi clúster (Corral et al., 2006).

Per tant, aquests resultats tan positius ens animen a continuar la investigació seguint aquesta línia, així com a avaluar l'aplicació d'altres paradigmes, com per exemple el CBR, que puguin ajudar a l'analista de seguretat a la detecció de vulnerabilitats a la xarxa.

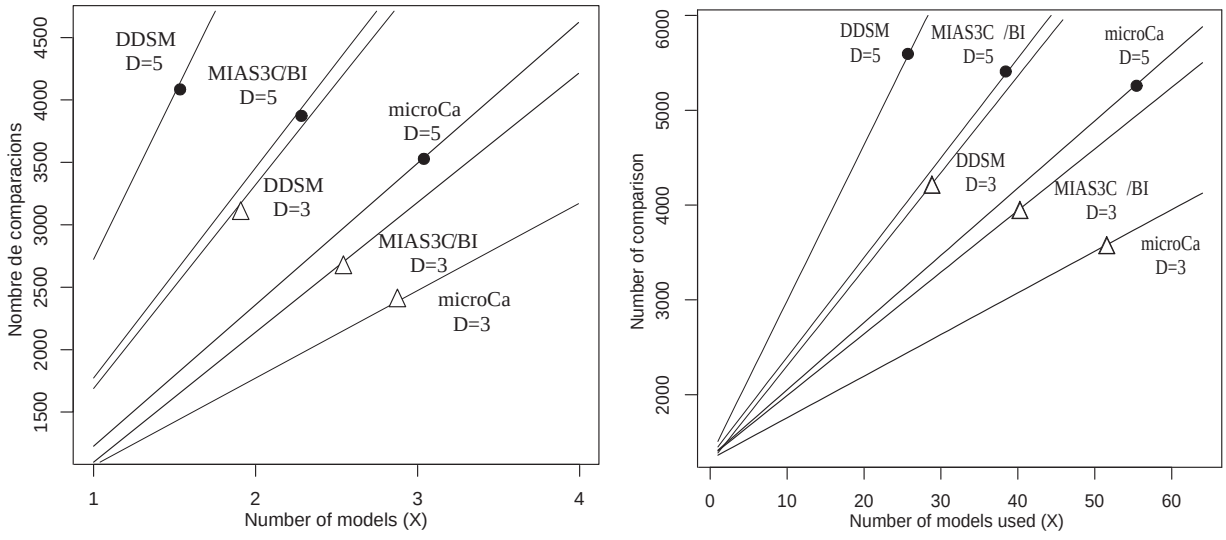

Figura 10.2: Evolució del nombre d'operacions que calen per aplicar el *Relevance Feedback* en un sistema CBR i en un altre SOMCBR. Els símbols 'Δ' and '$\bullet$' representen les configuracions associades als valors 3 i 5 de D respectivament.

10.4 Conclusions y línies futures

Aquest capítol ha presentat dues aplicacions de la recerca realitzades a la tesi dins dels projectes HRIMAC i ANALIA.

Al projecte HRIMAC s'ha abordat la incorporació de l'expert en el procés de cerca de les mamografies per reduir la distància entre el que el sistema busca, i el que l'expert vol buscar. Per fer això viable, s'ha reemplaçat el sistema CBR per un sistema SOMCBR, el qual és capaç de trobar la informació d'una manera més ràpida i precisa.

Com a línia futura seria interessant migrar el sistema basat en característiques de baix nivell, a un altre basat en conceptes. Per fer-ho, caldria etiquetar la informació de les imatges amb les dades de l'historial del pacient. Al mateix temps, això faria necessari desenvolupar mètriques de comparació més complexes per comparar eficaçment els casos. Aquesta última línia es podria afrontar des del punt de vista del disseny de funcions de similitud específiques per un domini, tema que es tracta a la part següent de la tesi.

D'altra banda, al projecte ANALIA s'ha contribuït en el desenvolupament de l'eina a partir de la integració de SOM dins del procés de processament de les dades. A més a més, s'ha col·laborat en el desenvolupament dels factors d'intra/inter cohesió per tal de mesurar la qualitat de l'agrupació dels dispositius, ja que al ser un domini no supervisat és complex determinar-ho 'a simple vista'. La línia futura de treball en aquest domini es basa en continuar investigant altres mètriques de compactació.

Les contribucions de les aplicacions presentades en aquest capítol es troben publicades als articles següents:

- A. Fornells, E. Golobardes, X. Vilasís i J. Martí. Integration of strategies based on relevance feedback into a tool for retrieval of mammographic images. Al *7th International Conference on Intelligent Data Engineering and Automated Learning*, volum 4224 de LNCS, planes 116-124. Springer-Verlag, 2006.

- G. Corral, A. Fornells, E. Golobardes i J. Abella. *Cohesion factors: improving the clustering capabilities of CONSENSUS*. Al *7th International Conference on Intelligent Data Engineering and Automated Learning*, volume 4224 de LNCS, páginas 488-495. Springer-Verlag, 2006.

Resum

El capítol ha presentat dues aplicacions concretes de les aportacions realitzades per la tesi en dos dels projectes on s'ha emmarcat. D'una banda, s'ha presentat la incorporació de mecanismes basats en *Relevance Feedback* dins de l'eina desenvolupada en el projecte HRIMAC per aconseguir millorar la precisió dels resultats retornats als experts. L'estratègia consisteix en aplicar mitjançant un procés iteratiu, una interacció amb l'expert on aquest marca els exemples que considera més rellevants. A partir d'aquesta informació, el sistema retorna els exemples més similars per cadascuna de les imatges marcades. El motor de cerca que es fa servir és el SOMCBR.

D'altra banda, SOM s'ha fet servir al projecte ANALIA com una eina de clustering per agrupar els dispositius segons les seves vulnerabilitats. Els resultats assolits usant SOM han estat positius i, a més a més, ha contribuït al desenvolupament de mètriques per mesurar el grau de cohesió entre els elements dels clústers.

Part III

Contribucions al disseny de funcions pel CBR ad hoc a un domini

Capítol 11

Disseny de funcions de similitud pel CBR usant GP i GE

Com s'ha explicat a la primera part, el CBR és un paradigma basat en l'aprenentatge analògic capaç de resoldre problemes a partir d'altres prèviament resolts. Per dur a terme això es fa necessària la definició d'una mètrica, anomenada funció de similitud, capaç de mesurar la similitud entre els casos. Tanmateix, no hi ha una funció de similitud universal que funcioni bé per qualsevol domini ja que cada domini és un 'nou món'. A més a més, els dominis reals presenten coneixement imprecís, incert, parcialment vertader i aproximat, aspectes que dificulten la definició manual de mètriques específiques al domini. Aquest capítol presenta dues estratègies que aprofiten les capacitats de *Soft-Computing* i de *Knowledge Discovery* de la CE per definir de manera automàtica funcions de similitud ad hoc a un domini. La primera estratègia és una millora sobre la recerca prèvia realitzada dins del GRSI a través de la Programació Genètica. La segona estratègia és una nova contribució basada en la utilització de l'Evolució de Gramàtiques. L'aportació del nou enfocament respecte el que s'havia fet anteriorment és la capacitat d'introduir restriccions de manera transparent als operadors genètics.

11.1 Motivació: aprendre a saber comparar

El Raonament basat en casos (CBR) (Aamodt i Plaza, 1994) és un paradigma basat en l'aprenentatge analògic que resolt nous problemes a partir d'altres similars prèviament resolts. Aquest enfocament li permet justificar les seves propostes de classificació basant-se en el grau de similitud entre els problemes. Per tant, el percentatge d'encerts està altament relacionat amb la capacitat que té el sistema per establir semblances.

Les funcions de similitud tenen un paper clau dins d'aquest procés perquè són les encarregades d'establir el grau de semblança mitjançant una mètrica prèviament determinada per un expert del domini. La seva definició és complexa ja que molts cops les relacions entre les característiques que defineixen els exemples no són trivials i, a més a més, el seu domini només és parcialment conegut. La solució davant d'això molts cops es basa en fer servir funcions de similitud de propòsit general ajustades al problema a resoldre. L'adaptació consisteix principalment a ajustar els arguments de la funció, així com preparar les dades per poder aplicar-les en el domini de la funció. Tot i que aquesta alternativa permet solventar el problema sota un cert error, els resultats obtinguts no són sovint els desitjables. Per tal de superar aquesta problemàtica es necessitaria disposar d'algun mètode capaç de definir/cercar funcions de similitud vàlides pel domini que es vol tractar. D'aquesta manera, l'expert disposaria d'una funció base a partir de la qual pogués treballar i, a partir del seu coneixement del domini, pogués acabar d'ajustar-la per obtenir millors resultats.

Els algorismes que formen part de la branca de la Computació evolutiva (CE) (Holland, 1975) poden veure's com algorismes de cerca de propòsit general basats en els principis de l'evolució. A partir d'una població d'individus (conjunt de possibles solucions) es recombina la seva informació mitjançant operacions genètiques per tal d'obtenir una solució aplicant una certa pressió selectiva.

Dins el ventall de tècniques basades en la CE, la GP (Koza, 1992) es caracteritza perquè els individus són programes/funcions que poden executar-se directament. Per tant, pot aprofitar-se aquesta propietat per definir un sistema que cerqui funcions de similitud específiques per un problema. Sota aquest plantejament en (Golobardes et al., 2001) i (Camps et al., 2003) es va proposar un sistema híbrid CBR-GP que tenia aquesta finalitat. No obstant, degut a la immensitat de l'espai de cerca el problema pot convertir-se en *NP-Hard*. Caldria definir restriccions sobre l'espai de cerca per tal de reduir-lo i fer el problema resoluble. El problema d'aquest enfocament és la complexitat d'introduir restriccions.

D'altra banda, existeixen variants a la CE que incorporen mecanismes per definir restriccions de manera implícita. Concretament, l'evolució de gramàtiques (GE) (Ryan et al., 1998) es basa en definir una gramàtica en forma *Backus Naur* (BNF) per dirigir el procés de cerca. Aquesta tècnica parteix del cicle de vida clàssic dels algorismes genètics (GA) (Goldberg, 1989) amb la diferència que a l'hora de realitzar l'avaluació, fa servir una gramàtica per transformar l'individu en un programa. Aquesta separació de l'espai de cerca del de solucions li permet introduir restriccions sobre la cerca sense afectar a la resta de l'algorisme. Per tant, la seva integració amb el CBR permetria definir un sistema com l'anterior però amb la capacitat inherent d'introduir restriccions. Aquest és el punt de partida d'aquesta tercera part de la tesi.

11.2 Treballs previs

Al llarg de la literatura es troben moltes funcions de similitud de propòsit general com poden ser la distància de *Minkowsky* (Bachelor, 1978), *Mahalanobis* (Nadler i Smith, 1993), *Camberra*, *Chebychev*, *Quadratic*, *Correlation*, i *Chi-quadrat* (Michalski et al., 1981), *Context-Similarity measure* (Biberman, 1994), *Contrast Model* (Tversky, 1977), les funcions basades en *hyperrectangles* (Salzberg, 1991) (Domingos, 1997) i *heterogenous distance functions* (Wilson i Martinez, 1997) entre d'altres.

Els resultats que s'obtenen amb aquestes funcions són millorables si es poden aconseguir funcions a mida pel problema. En la literatura es troben molts exemples d'estratègies que intenten aconseguir això. Existeixen molts sistemes que fan servir estratègies basades en la CE amb la finalitat de millorar 'l'adaptació' de les funcions, com per exemple, ajustant la importància dels atributs en les funcions mitjançant esquemes de ponderació (Kuncheva, 1995; Kelly i Davis, 1991), o fent servir processos basats en tècniques de selecció de característiques (Siedlecki i Sklansky, 1998; Jarmulak et al., 2000). També hi ha altres enfocaments que els apliquen com a algorismes d'extracció de característiques (Raymer et al., 1996; Ahluwalia i Bull, 1999). No obstant, cap d'aquestes estratègies permet modelar la funció de similitud amb restriccions o coneixement específic, només ataquen a les dades del problema.

En treballs previs del GRSI (Golobardes et al., 2001; Camps et al., 2003) va proposar-se un sistema híbrid entre el CBR i la GP per definir de manera automàtica i adaptativa funcions de similitud pel domini de les mamografies del projecte HRIMAC (vegeu l'apèndix B), i per un conjunt de problemes sintètics definits on no podien aplicar-se correctament les funcions de similitud de propòsit general. El procés de cerca era automàtic perquè la GP buscava en l'espai de funcions possibles quina era la més adequada pel domini concret sense la necessitat de disposar d'un expert, i adaptativa perquè es permetia incorporar informació del domini posteriori a la seva definició. No obstant, el sistema tenia com principal inconvenient el fet d'haver de buscar en espais de cerca massa grans, que podien convertir el problema en *NP-Hard*. Aquest espai de cerca pot reduir-

se mitjançant la incorporació de restriccions sobre els individus, de tal manera que s'exploren només individus que són potencialment bons. Aquesta finalitat és la que tracten d'aconseguir els enfocaments de la CE basats amb gramàtiques BNF (Ryan et al., 1998), en el context de cada element (Whigham, 1995; Oltean, 2003), o simplement definint tipus de dades especials per representar dades complexes com vectors o matrius (Montana, 1993).

11.3 Integració dels cicles de la GP i de la GE dins el CBR

L'objectiu és disposar d'un sistema capaç de definir una funció de similitud específica per un problema concret. Per aconseguir-ho, es proposa integrar dins d'un mateix entorn un sistema amb capacitat d'exploració de solucions (GP o GE), i un altre que pugui avaluar-les (CBR). L'estratègia basada en CBR-GP permet trobar la funció sense fer servir coneixement específic del domini. En canvi, si fem servir aquest coneixement per reduir l'espai de cerca, l'estratègia basada en CBR-GE permet aprofitar aquesta informació per dirigir la cerca.

Tal com mostra la figura 11.1 ambdues estratègies tenen bàsicament les mateixes fases, i les seves diferències vénen provocades per la utilització de representacions diferents. Inicialment es parteix d'una població inicialitzada pseudo-aleatòriament, ja que cal respectar unes normes com es veurà més endavant, la qual va evolucionant al llarg de diferents generacions i sobre la qual s'aplica el cicle de vida de la GP/GE explicat anteriorment a l'apartat 4.2.

A l'hora de realitzar l'avaluació de la funció de similitud que representa l'individu és quan apareix la interacció amb el CBR. Aquest procés consisteix en transformar l'individu en una expressió que representi el programa (vegeu la figura 11.2). En el cas de la GP només cal fer un recorregut en inordre, i en el cas de la GE cal aplicar una transformació genotip-fenotip sobre una gramàtica BNF prèviament definida. A continuació, el CBR fa servir aquesta expressió com a funció de similitud dins el seu cicle de vida, tal i com es va explicar a l'apartat 2.2. En aquest cas, el CBR al comparar dos casos ha de reemplaçar les variables de l'expressió pels atributs dels casos, i a continuació processar l'expressió i d'aquesta manera obtenir la semblança entre ells.

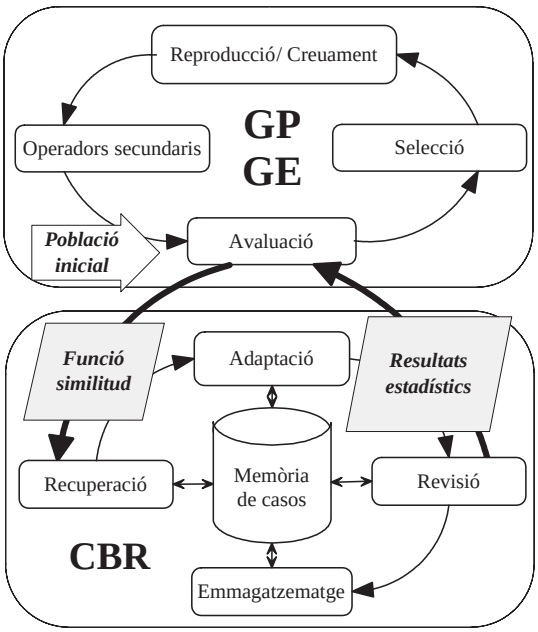

Figura 11.1: Integració de la GP i la GE en el CBR.

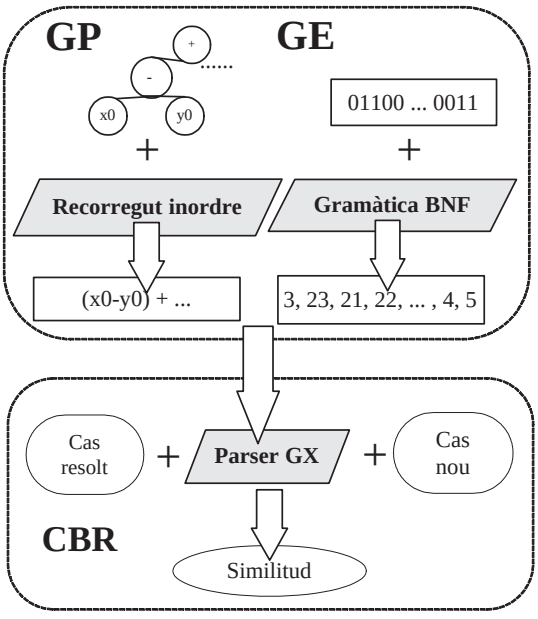

Figura 11.2: Transformació d'un individu en una funció.

Un cop que el CBR ha avaluat mitjançant un *N-fold stratified Cross Validation* l'expressió que representava la funció de similitud, aquest genera unes estadístiques a partir de les quals es mesura el grau d'adaptació de la funció de similitud, o el que és el mateix, el *fitness* de l'individu. Cal tenir present que durant tot aquest procés el CBR mai memoritza res, ja que sinó s'estarien falsejant els resultats.

11.4 Representació de les funcions

Els següents subapartats descriuen els elements que formen part de la representació dels individus en la GP i en la GE, així com els mecanismes que es fan servir per gestionar les restriccions en ambdues aproximacions.

11.4.1 Elements que intervenen

Els individus en els dos enfocaments representen funcions de similitud. Tenint en compte això, poden diferenciar-se els elements següents:

- **Variables**. Representen els atributs del cas nou (X_N), i el cas resolt (Y_N). En ambdós casos N indica el nombre d'atributs que té el cas pel problema que s'està tractant.
- **Constants**. Són nombres que afegeixen un cert *offset*. El seu valor és molt complex de definir ja que hi ha infinites constants tal i com es va exposar a l'apartat 4.7.
- **Operacions**. Representen les operacions aritmètiques que es permeten fer en la funció, per exemple: +, -, *, /, logaritme, sinus o cosinus entre altres.

És important remarcar que cada tipus d'element estableix unes relacions concretes amb la resta d'elements segons la seva tipologia i, per tant, cal vetllar pel principi de clausura (vegeu l'apartat 4.11). Per exemple, un logaritme no hauria de rebre un 0. La definició dels elements condicionarà l'èxit de l'algorisme ja que limitarà o estendrà l'espai de cerca on buscar, fent que el problema pugui convertir-se en *NP-Hard*.

A continuació es detalla la definició de la jerarquia dels elements per les dues aproximacions, així com diferents maneres d'introduir restriccions per reduir l'espai de cerca.

11.4.2 Aplicació de restriccions sintàctiques i semàntiques en l'arbre n-ari

Els individus en la GP es representen mitjançant arbres M-aris, de tal manera que les relacions entre els elements (variables, operacions i constants) es representen mitjançant les relacions de pare-fill en els nodes. Durant el cicle de vida de l'algorisme hi ha un conjunt de situacions que poden comprometre la 'correctessa sintàctica' de la funció de similitud i, per tant, cal controlar:

- **Inicialització dels individus.** Els individus al ser inicialitzats han de respectar les relacions entre variables, constants i funcions, per tal de construir expressions aritmètiques sintàcticament vàlides.
- **Creuament d'individus.** El/s punt/s de tall del segon pare han de ser seleccionats en funció del/s primer pare per construir expressions aritmètiques sintàcticament vàlides.
- **Mutació dels nodes.** Els canvis aleatoris han de fer-se només per elements que siguin equivalents. Les variables i les constants són tipus equivalents que poden intercanviar-se sense cap problema. Les operacions només es poden canviar per d'altres operacions.

La manera com aquests individus són alterats durant el cicle de vida pot semblar un procés aleatori, és a dir, que molts cops es realitzen combinacions que no tenen cap sentit i que l'única cosa que fan és endarrerir la convergència de l'algorisme, com per exemple, permetre un producte entre dos atributs que no tenen res a veure. Per tant, seria interessant evitar situacions que tot i ser sintàcticament correctes, no tenen cap sentit semàntic. És en aquest punt on entren en joc les restriccions semàntiques. Si es vol buscar una funció de similitud que sigui semànticament correcte, aquesta hauria de garantir un conjunt de propietats desitjables:

1. Tots els atributs han d'aparèixer.
2. Les operacions entre atributs només són vàlides si són equivalents (mateix tipus i significat).
3. Els atributs han de tenir un pes específic sobre el resultat. La ponderació 0 equival a que l'atribut té una contribució nul·la.
4. Els nodes funcions han de rebre tants paràmetres com indiqui la seva aritat.
5. Els nodes funcions han de rebre com a valors, elements que pertanyin al domini de la funció.

Aquestes regles permeten reduir considerablement l'espai de cerca, permetent que l'algorisme convergeixi abans i trobi solucions amb un mínim de significat. No obstant, aplicar restriccions també pot provocar problemes:

- **No es trobi la solució.** L'espai de cerca pot acotar-se massa si es defineixen restriccions massa estrictes.
- **Cost extra.** Forçar les restriccions semàntiques implica un cost addicional en tots els moments on l'individu pot ser alterat.
- **Control de les inicialitzacions**. Els individus han de ser inicialitzats amb estructures i valors vàlids sintàctica i semànticament.
- **Adaptació d'operadors.** El resultat dels operadors primaris i secundaris han de ser individus vàlids sintàctica i semànticament.
- **Increment de temps.** Tot i que l'algorisme pot convergir abans perquè ha d'explorar un espai de cerca més reduït, el temps d'execució pot ser superior degut als costs addicionals provocats pel control i la supervisió per garantir la sintaxi i la semàntica de les funcions.

Per tant, afegir restriccions sintàctiques i semàntiques té avantatges i inconvenients. A continuació proposem dues estratègies diferents per gestionar les etapes conflictives, i d'aquesta manera ser més o menys flexibles en el desenvolupament del cicle de vida.

11.4.2.1 Restriccions de nivell 1

Són restriccions que permeten mantenir les restriccions sintàctiques i semàntiques en la fase d'inicialització dels individus i, d'una manera més relaxada, a l'hora d'aplicar els operadors de creuament i mutació. Aquest criteri permet aplicar restriccions 'suaus' sobre els individus, per permetre una cerca més flexible.

De les propietats semàntiques que calia garantir, era important que les operacions entre atributs només es permetessin entre atributs equivalents per tal d'evitar generar funcions de similitud sense sentit. A més, era desitjable poder mesurar l'aportació de cada atribut sobre la similitud final. Per tal d'aconseguir-ho es replanteja la classificació dels tipus de nodes separada en nodes terminals (variables i constants) i nodes funcions (operacions). Ara els nodes terminals es separen

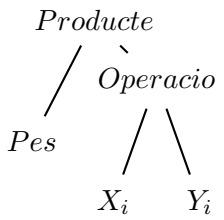

Figura 11.3: Esquema de l'abstracció que representa el node terminal restringit.

en nodes terminals simples (constants) i en nodes terminals restringits (vegeu la figura 11.3). Els nodes terminals restringits són un node atòmic que es descomposa en un petit subarbre, el qual defineix una operació ponderada entre dos atributs equivalents. El conjunt de terminals restringits estarà format ara per totes les combinacions possibles de ponderacions (discretitzades entre [0..1]), operacions (nodes funcions d'aritat dos permeses entre atributs) i un parell de nodes terminals (atributs del cas).

Aquesta representació especial de les relacions entre els atributs permet simplificar notablement les operacions de control d'inicialització, creuament i mutació. No obstant, per cadascuna d'aquestes cal tenir en compte:

Inicialització. Els individus han de respectar les següents propietats al ser definits:

- Han de ser presents tots els atributs, i per tant, cal garantir un nombre mínim de profunditat tal com indica l'equació 11.1:

$$profunditat_{\text{mínima}} = log_2(\text{número d'atributs}) \tag{11.1}$$

- La inicialització ha de garantir que tots els parells d'atributs apareixen al menys un cop. Això s'aconsegueix assignant nodes terminals restringits amb parells d'atributs diferents, fins que al menys existeixi una còpia de cada un.

Creuament d'individus. El creament que es permet en aquest cas no és del tot restrictiu, ja que la seva aplicació pot fer que en la funció no hi hagi al menys una còpia de cada atribut. Les restriccions que s'han de satisfer són:

- Respectar la profunditat màxima i mínima de l'arbre.
- L'esquema del node terminal definit a la figura 11.3 és atòmic, és a dir, no pot dividir-se. Per tant, mai podrà ser seleccionat qualsevol dels seus elements interns per produir-se un creuament, però sí ell de manera global.
- Si el punt de creuament és un node funció, només es pot escollir un altre punt de creuament que tingui la mateixa aritat i tipus de domini.
- Si el punt de creaument és un node terminal (restringit o simple) només pot canviar-se per un altre node terminal.

Mutació. Cal controlar els efectes de la mutació amb les següents regles:

- Mutar un node funció implica canviar el node funció per un altre equivalent, és a dir, mateixa aritat i tipus de domini.
- Mutar un node terminal restringit implica modificar l'operació entre els dos atributs per un altre equivalent o modificar el pes que el pondera.
- Mutar un node terminal simple consisteix en intercanviar el seu valor per un altre node terminal.

11.4.2.2 Restriccions de nivell 2

Les restriccions de nivell 2 redefineixen el funcionament dels operadors de les restriccions de nivell 1 per tal de garantir que els individus resultants de l'aplicació del creuament i la mutació garanteixin les propietats. En el cas del creuament cal garantir que al combinar els pares, els dos fills resultants han de tenir al menys una còpia de cada atribut. Per tant, la selecció del punt de tall del segon pare està condicionada pel fet que es respecti el nombre d'atributs en ambdós futurs fills.

D'altra banda, l'aplicació de l'operador de mutació permet a més la possibilitat de modificar l'atribut del node terminal restringit sempre i quan, després hi hagi al menys una còpia de cadascun dels atributs en l'individu.

11.4.3 Aplicació de restriccions sintàctiques i semàntiques amb la gramàtica BNF

La capacitat de separar l'espai de cerca del de solucions en la GE permet definir restriccions d'una manera molt senzilla sense haver de modificar els operadors, ja que només cal canviar la definició de la gramàtica BNF. En el nostre cas, la forma seria la següent:

- Sigui G una gramàtica BNF formada per {N, T, P, S}.
- Sigui T ={$x_0..x_{N-1}$, $y_0..y_{N-1}$,$constants, +, -, *, /, \%, abs, cos, exp, ln, log, sin, sqr, sqrt, tan, (,)$}, on:
  - N és el nombre d'atributs.
  - Els operadors $\{ln, log, sqrt\}$ s'apliquen sempre amb valors absoluts sobre els arguments. Si el valor és igual a 0, l'avaluació de l'individu s'atura.
  - Els atributs són sempre valors reals.
  - La divisió entre 0 implica que l'avaluació de l'individu s'atura.
  - L'argument del sinus, cosinus i tangent es calcula en radians.
- Sigui N = {<expressio>, <operador_unari> ,<operador_binari>, <variable>}
- Sigui S la producció inicial, {<expressio>}
- Siguin P les produccions,

  <**expressio**> → (<expressio> <operador_binari> <expressio>)
  → <operador_unari> (<expressio>)
  → <variable>

  <**operador_binari**> → + | - | * | / | %

  <**operador_unari**> → abs | cos | exp | log | ln | sin | sqr | sqrt | tan

  <**variable**> → x_0 | x_1 | ... | x_{N-1} | y_0 | y_1 | ... | y_{N-1} | constants

No obstant, apareix el mateix problema d'abans ja que poden generar-se funcions de similitud sintàcticament vàlides però semànticament incorrectes. Per tant, caldria modificar les definicions anteriors per garantir:

- Que apareguin tots els atributs.
- Operacions permeses només entre atributs equivalents.
- Tots els atributs han de tenir un pes específic sobre la similitud resultant.

- Obtenir una sortida normalitzada en funció del nombre d'atributs que es fan servir.

Aquests requeriments impliquen modificar les produccions P per definir operacions binàries entre expressions, i altres entre els atributs (operador_binari_bis). Les produccions P queden de la manera següent:

<**S**> → <expressio>

<**expressio**> → (<expressio> <operador_binari> <expressio>)
→ <operador_unari> (<expressio>)
→ <constants> * (<variable>)

<**operador_binari**> → + | - | * | / | %

<**operador_unari**> → abs | cos | exp | log | ln | sin | sqr | sqrt | tan

<**variable**> → x_0 <operador_binari_bis> y_0 | ... | x_{N-1} <operador_binari_bis> y_{N-1}

<**operador_binari_bis**> → + | - | * | / | %

<**constants**> → 0 | 0.1 | ... | 1

No obstant, només amb això no es garanteixen tots els requeriments ja que no es pot saber quins i quants atributs s'han fet servir:

- Les regles de la producció <variable> són seleccionades aleatòriament.
- El mapeig pot finalitzar sense que es facin servir tots els atributs.

En la GE els codons es mapejen a partir de la resta de la divisió entre el codon actual i el nombre de regles de la producció actual. Això genera un índex entre 0 i MAX_REGLES-1 que permet seleccionar una regla. En el nostre cas ens interessaria:

- Que cada entrada de <variable> aparegui al menys 1 cop per cada parell d'atributs.
- Que la producció <variable> es cridi al menys N cops, essent N el nombre d'atributs.

Aquests requeriments poden ser assolits si es modifica la manera en la qual es realitza el procés de mapeig. Per fer-ho, es defineix un flag anomenat '*producció utilitzada*' per cada regla de la producció <variable>. Aquest flag indica si la regla s'ha utilitzat i, per tant, si ja s'ha introduït la variable. Quan el codon s'ha de mapejar mitjançant l'enter que selecciona la nova regla, si aquesta ha estat prèviament utilitzada es fa servir la següent que no s'hagi fet servir. Això permet seleccionar al menys sempre regles diferents de la producció <variable> fins que al menys hi hagi una còpia de totes.

L'altre problema pot solucionar-se si es garanteix que al menys es facin N-1 crides de les producció <expressio> de la primera regla, i després aquestes es refereixin a la crida <variable>, és a dir, els primers N-1 símbols de l'individu es podrien forçar a '0' per permetre N expressions, i els N següents al valor '2' per permetre cridar la regla <variable>. Gràcies a aquestes propostes es poden aconseguir validar els requeriments explicats anteriorment. El fet de garantir això implica:

- Han de ser inicialitzats amb al menys 2N-1 elements, els N-1 primers assignats a '0', i els altres N assignats a '2'. La resta de codons poden tenir qualsevol element.
- El creuament s'ha de fer sempre sobre elements que no estiguin en les 2N-1 primeres posicions.
- La mutació no pot aplicar-se sobre els 2N-1 primers elements.

Aquests canvis dels operadors són tan subtils que no suposen una carrega addicional en l'aplicació dels operadors.

11.5 Avaluació de la precisió de la funcions de similitud

La finalitat d'aplicar el CBR en el cicle de vida de la GP i la GE és permetre avaluar la validesa de la funció de similitud que l'individu representa. Aquesta validesa es mesura fent servir les estadístiques de classificació, les quals poden estar formades pel % de classificacions correctes, el % de classificacions incorrectes, el % de no classificats, el % d'especificitat, el % de sensitivitat, desviacions típiques, o qualsevol altre paràmetre que es vulgui tenir en compte.

En funció del grau d'importància que s'assigni a cada estadística es defineix una estratègia per modelar el comportament de la funció. Per exemple, l'equació 11.2 modela un comportament on es premia la fiabilitat. A l'apartat D.3.2 s'introdueix la sensitivitat i l'especificitat com a paràmetres que mesuren els vertaders positius i negatius, essent això una manera d'avaluar com de crític és el fet que el sistema s'equivoqui. En canvi, si es vol minimitzar el % error premiant, el % no classificats davant del dubte es podria fer servir l'equació 11.3. En ambdós casos, w_i representa la importància de cada paràmetre segons el comportament que es vulgui assignar, essent $\sum w_i = 1$.

$$fitness = w_0 \cdot \%sensitivitat + w_1 \cdot \%especificitat - w_2 \cdot \%casos\_no\_classificats \quad (11.2)$$

$$fitness = w_0 \cdot \%encerts + w_1 \cdot \%casos\_no\_classificats \quad (11.3)$$

Mitjançant aquestes estadístiques ponderades es pot modelar el comportament del sistema per tal d'adaptar-lo a les necessitats del domini del problema.

11.6 Consideracions prèvies a tenir en compte

Per poder aplicar aquest híbrid en un problema cal tenir en compte tots els aspectes que s'havien comentat anteriorment a l'apartat 2.9 del CBR, i a l'apartat 4.11 de la GP i la GE.

Tot i que crear un sistema híbrid entre dos models d'aprenentatge tan diferents com són el CBR i la GP/GE ens proporciona més potència per poder trobar la solució al problema, també augmenta la complexitat a l'hora de plantejar com abordar el problema. Aquestes dificultats vénen provocades principalment per:

Parametrització. El CBR i la GP/GE tenen molts paràmetres de configuració, cal estudiar com es veuen afectats entre ells segons el domini, i d'aquesta manera trobar la combinació que ens proporciona millors resultats.

Espai de cerca. La gran quantitat de combinacions entre variables i operacions fa que hi hagi una gran quantitat de funcions de distància, cal cercar mecanismes per reduir l'espai de cerca i tenir en compte només les que són potencialment bones mitjançant la definició de restriccions.

Memòria de casos del CBR. L'avaluació d'una funció de distància implica executar un *N-fold stratified Cross Validation* en el CBR. El temps d'avaluació està relacionat directament amb la quantitat d'exemples *train* i *test* que ha de fer servir, així com el nombre d'atributs. Cal estudiar si totes les dades de la memòria són rellevants, així com la importància dels atributs dels casos. Una reducció del volum de dades de la memòria farà més eficaç i eficient el procés d'avaluació.

11.7 Avaluació de la capacitat en trobar funcions específiques

Al llarg dels punts següents s'avalua la capacitat que tenen els dos enfocaments per definir funcions de similitud específiques per un domini de manera automàtica. Un cop descrita l'experimentació a realitzar, es presenten i s'analitzen els resultats.

11.7.1 Experimentació

Un cop presentats els enfocaments CBR-GP i CBR-GE arriba el moment de la pregunta inevitable: quina proposta és millor? Com succeeix sempre a l'hora de comparar tècniques, el terme 'millor' és molt relatiu i engloba moltes propietats. Per aquest motiu, ambdues propostes s'avaluaran des de diferents punts de vista responent a:

- Quin impacte té el nombre d'atributs del problema en la cerca? El nombre d'atributs − suposant que es tenen en compte només els que aporten informació útil − és un aspecte vital de la funció. Per tant, cal estudiar el seu impacte sobre diferents dominis que tinguin diferents nombre d'atributs.

- Les funcions trobades pels sistemes tenen sentit? Les funcions haurien de tenir un mínim de coherència, és a dir, relacionar aspectes del domini que tenen sentits. A més a més, els atributs seran tots numèrics per tal d'evitar la influència de la discretització de les dades.

- Quina proposta proporciona funcions de similitud que generen resultats més fiables en termes de sensitivitat i especificitat? Aquest aspecte és molt important per estudiar el grau de precisió en dominis complexes i incerts com el del càncer de mama del projecte HRIMAC.

- Són significatius els resultats respecte les funcions de similitud de propòsit general? Està molt bé voler definir funcions que siguin específiques al domini, però cal avaluar si realment aporten alguna millora. Aquesta avaluació es realitzarà a partir del càlcul del t-test amb un nivell de significància del 95% respecte el rendiment que ofereixen les funcions de distància més comunes a la literatura.

- Quina proposta convergeix més ràpid? En el hipotètic cas de que ambdues propostes aportessin funcions de similitud que fossin equivalents, quin mètode és més costós?

Els *datasets* utilitzats per avaluar les funcions generades per les dues propostes provenen de l'*UCI Repository* (Asuncion i Newman, 2007) i dels projectes FIS (Garrell et al., 1998) i HRIMAC (vegeu l'apèndix B), i es troben descrits a la taula 11.1. La taula conté el nom, nombre d'atributs, distribucions de les classes i el nombre total d'instàncies. Tots els *datasets* tenen només dues classes per tal de realitzar l'estudi basat en la sensitivitat i l'especificitat. Tots els *datasets* són normalitzats i corregits abans de ser utilitzats.

Taula 11.1: Descripció dels *datasets* utilitzats en la comparativa CBR-GP vs CBR-GE.

Codi	***Dataset***	**Atributs**	**Distribució de les classes**	**Instàncies**
TA	tao	3	negre (944), blanc (944)	2500
WS	wisconsin	10	benigne (458), maligne (241)	699
MX	multiplexer	12	0 (1024), 1 (1024)	2048
HS	heart-Statlog	15	absent (150), present (120)	270
MF	μCa	22	benigne (121), maligne (95)	216
BI	biopsies	25	0 (530), 1(497)	1027
IO	ionosphere	36	benigne (126), maligne (225)	351
SO	sonar	62	roca (97), mina (111)	208

D'altra banda, com que les aproximacions de la GP i la GE depenen de molts paràmetres, a l'experimentació s'han avaluat un conjunt ampli de configuracions per maximitzar les possibilitats de trobar la millor configuració pel problema en qüestió. Aquestes dades es troben resumides a la taula 11.2. La gramàtica utilitzada es descriu a la figura 11.4.

Tot i que, s'ha implementat una aproximació CBR-GP anomenada JACK, la que s'utilitza a les proves és la de (Golobardes et al., 2001; Camps et al., 2003). El motiu és perquè la plataforma JACK està basada en Java ja que quan estava pensada per ser integrada en un entorn Java d'algorismes genètics (KEEL - TIC2002-04036-C05), i això fa que no es pugui comparar la convergència

Sigui **G={N, T, S, P}** una gramàtica BNF on:
N = {<expressio>,<operador_binari>,<operador_unari>,<variable>, <operador_bin_bis>,<constants>}
T = $\{x_0 \dots x_{N-1}, y_0 \dots y_{N-1}, +, -, *, /, abs, \ ^2, \sqrt{\ }\}$
S = <expressio> / (#Atributs utilitzats)
P = <expressio> → (<expressio> <operador_binari> <expressio>)
→ <operador_unari> (<expressio>)
→ <constants> * (<variable>)
<operador_binari> → + | - | * | / | %
<operador_unari> → abs| ^{2}|$\sqrt{\ }$
<variable> → x_0 <operador_binari_bis> y_0 | $\cdots$ | x_{P-1} <operador_bin_bis> y_{P-1}
<operador_bin_bis> → + | - | * | / | %
<constants> → 0 | 0.1 | ... | 1

Figura 11.4: Gramàtica BNF de la GE que mapeja els individus en funcions.

Taula 11.2: Configuracions pel CBR, CBR-GP i CBR-GE.

Paradigma	**Arguments**	**Valors**
CBR	Funció	Clark i Minkowsky (r=1, 2, 3)
	Ponderació	Sense Pes (SP), Anàlisi de Components Principals (PCA), Correlació Mostral (CM)
	K-NN	1, 3, 5
	Resultats	Mitja dels resultats del *10-fold stratified Cross-Validation*
GP & GE	Població	250
	Acabament	500 generacions or 0.95% del *fitness* ideal
	Operadors	Prob. Creuament (0.8), Prob. Reproducció (0.2), Prob. Mutació (0.3)
	Selecció	Torneig amb 2 individus (ST) i per Rang (SR)
	Avaluació dels individus	CBR en *10-fold stratified Cross-Validation* fent servir l'equació 11.2 amb diferents valors w_i ($\sum w_i = 1$).
	Inicialització	*Grow* (GI), *Full* (FI), *Ramped half and half* (RI)
	Reeplaçament	*Steady-State* (SR), Generacional (GR)
	Llavors aleatòries	5
Només la GP	Profunditat max. arbre	7 nivells
	Nodes Terminals	$x_0 \dots x_{N-1}, y_0 \dots y_{N-1}$
	Nodes Funció	$+, -, *, /, \| \ \|, \ ^2, \sqrt{\ }$
Només la GE	# Codons	Hi ha 10 codons per atribut
	Operador Wrapping	S'aplica com a màxim dos cops
	Gramàtica	Figura 11.4

dels algorismes en quant a temps computacional perquè l'aproximació CBR-GE s'ha implementat en C++ sota la plataforma BRAIN. Ambdues plataformes es troben descrites a alt nivell al següent capítol. A més a més, això permetrà comparar un esquema sense restriccions (CBR-GP) amb un altre que sí (CBR-GE).

11.7.2 Anàlisi i discussió dels resultats

L'apartat analitza la bondat de les funcions de similitud trobades pel CBR-GP i el CBR-GE respecte les funcions de propòsit general més habituals. Els resultats d'executar el CBR amb totes aquestes funcions estan resumits a la taula 11.3, i s'analitzen des dels diferents punts de vista esmentats anteriorment.

Mesura de la fiabilitat i la significància dels resultats proporcionats per les funcions de similitud

La taula 11.3 és un resum del % de sensitivitat, el % d'especificitat, i % d'encerts de les millors configuracions descrites a la taula 11.2. Entenem per configuració, el millor resultat obtingut amb una funció de similitud fent servir diferents paràmetres. A partir dels resultats poden extraure's les següents observacions:

***Dataset* TA.**

- Els millors resultats en les estadístiques són els proposats per la funció proposada per l'híbrid CBR-GE.
- No hi ha millora significativa (*t-student* amb un nivell de confiança del 95 %) respecte la millor funció de similitud de propòsit general i la funció CBR-GP.
- La funció CBR-GE té les desviacions típiques més petites.

***Dataset* WS.**

- Els millors resultats en les estadístiques són els proposats per la funció Minkowski (r=1).
- No hi ha millora significativa (*t-student* amb un nivell de confiança del 95 %) respecte les funcions CBR-GE i CBR-GP.
- La funció Minkowski (r=1) té les desviacions típiques més petites.

***Dataset* MX.**

- Els millors resultats en les estadístiques són els proposats per la funció Clark.
- Hi ha millora significativa (*t-student* amb un nivell de confiança del 95 %) respecte totes les altres funcions.
- Els resultats són molt dolents perquè el problema no és linealment separable.

***Dataset* HS.**

- Els millors resultats en les estadístiques són els proposats per la funció proposada per l'híbrid CBR-GE.
- Hi ha millora significativa (*t-student* amb un nivell de confiança del 95 %) respecte la millor funció de similitud de propòsit general i la funció CBR-GP.
- La funció CBR-GE té les desviacions típiques més petites.

Taula 11.3: Resultats de les millors configuracions fent servir diferents funcions de similitud. Per cada configuració s'indica el % de sensitivitat, el % d'especificitat, i el % d'encerts, juntament amb les seves desviacions típiques respectives. Els símbols ↑ i ↓ indiquen si la funció CBR-GP o CBR-GE millora o no significativament el millor resultat de les funcions de propòsit general, marcades en **negreta**, al aplicar el *t-student* amb un nivell del 95% de confiança. D'altra banda, el símbol √ indica quina proposta CBR-GP o CBR-GE és millor.

Codi	**Funció**	**Configuració**	**%Sensitivitat**	**%Especificitat**	**%Encerts**
TA	Clark	K-NN=3, PCA	95.1(2.1)	96.1(2.7)	95.3(1.4)
	Mink.(r=1)	K-NN=5, SP	96.4(2.2)	96.5(2.1)	96.4(1.5)
	Mink.(r=2)	**K-NN=5, SP**	**96.7(2.4)**	**96.5(1.9)**	**96.6(1.4)**
	Mink.(r=3)	K-NN=5, PCA	96.9(2.8)	96.4(2.1)	96.6(1.6)
	GP-CBR	RI, GR, RS, w{.4,.4,.2}	94.2(1.7)	96.1(2.1)	95.7(2.1)
	GE-CBR	FI, SR, TS, w{.4,.4,.2}	97.6(0.9)	95.3(1.5)	96.8(1.4)
WS	Clark	K-NN=1, PCA	94.3(2.4)	97.2(1.9)	96.1(2.1)
	Mink.(r=1)	**K-NN=3, SP**	**95.4 (2.8)**	**97.8(0.9)**	**97.0(2.9)**
	Mink.(r=2)	K-NN=3, PCA	94.84 (2.9)	98.2(1.2)	97.0(2.6)
	Mink.(r=3)	K-NN=3, PCA	94.5(3.8)	98.0(1.4)	96.71(3.1)
	GP-CBR	RI, SS, TS, w{.4,.4,.2}	96.4(2.4)	95.8 (2.4)	96.1 (2.9)
	GE-CBR	FI, SS, TS, w{.35,.35,.3}	95.9(3.1)	96.2(2.1)	95.99(2.8)
MX	**Clark**	**K-NN=3, PCA**	**95.1(2.1)**	**95.4(1.1)**	**95.2(1.5)**
	Mink.(r=1)	K-NN=3, SP	50.0 (0.15)	50.0(0.15)	50.0(0.24)
	Mink.(r=2)	K-NN=3, SP	50.0 (0.15)	50.0(0.15)	50.0(0.24)
	Mink.(r=3)	K-NN=3, SP	50.0 (0.15)	50.0(0.15)	50.0(0.24)
	GP-CBR	RI, SS, TS, w{.4,.4,.2}	50.0 (0.15) ↓	50.0 (0.15) ↓	50.0 (0.24) ↓
	GE-CBR	RI, GR, TS, w{.4,.4,.2}	50.0 (0.15)↓	50.0(0.0) ↓	50.0(0.24) ↓
HS	Clark	K-NN=3, CM	44.8(1.1)	0 (0)	44.44(0)
	Mink.(r=1)	**K-NN=3, CM**	**81.1(9.9)**	**82.8(6.5)**	**81.4(6.4)**
	Mink.(r=2)	K-NN=3, PCA	81.2(8.2)	78.9(9.8)	79.2(6.8)
	Mink.(r=3)	K-NN=5, PCA	81.6(9.5)	79.2(8.6)	80.0(9.6)
	GP-CBR	RI, GR, RS, w{.4,.4,.2}	65.8(9.6) ↓	82.6(8.5)	75.18(9.4) ↓
	GE-CBR	RI, GR, TS, w{.4,.4,.2}	85.1(8.5) ↑ √	↑85.8(5.4)	85.2(6.4) ↑ √
MF	Clark	K-NN=3, CM	61.2(9.7)	70.5(7.1)	65.7(8.6)
	Mink.(r=1)	**K-NN=3, PCA**	**62.3(6.9)**	**73.7(11.3)**	**68.1(10.0)**
	Mink.(r=2)	K-NN=5, PCA	62.0(8.6)	72.3(8.7)	67.1(12.4)
	Mink.(r=3)	K-NN=3, SP	61.5(7.3)	71.1(7.2)	66.6(8.1)
	GP-CBR	RI, GR, RS, w{.4,.4,.2}	67.7(9.3)↑	61.8(6.8) ↓	64.2(7.2)↓
	GE-CBR	RI, SR, TS,w{.4,.4,.2}	67.6(7.5)↑	76.3(9.9) √	71.4(7.9)↑ √
BI	Clark	K-NN=3, CM	84.4(5.2)	80.6(3.3)	82.2(3.8)
	Mink.(r=1)	K-NN=5, CM	85.4(4.5)	83.6(3.7)	84.3(3.7)
	Mink.(r=2)	**K-NN=5, CM**	**86.4(5.5)**	**83.5(2.6)**	**84.6(3.4)**
	Mink.(r=3)	K-NN=5, CM	86.6(5.8)	83.2(3.0)	84.4(3.2)
	GP-CBR	RI, SS, TS, w{.4,.4,.2}	88.2 (5.2) √	75.3 (7.7) ↓	81.5 (4.9) √
	GE-CBR	RI, GN, TS, w{.35,.35,.3}	73.4(2.1) ↓	75.4(3.3) ↓	74.3(2.5) ↓
IO	Clark	K-NN=5, CM	0.0(0)	0.0(0)	0.0(0)
	Mink.(r=1)	**K-NN=3, CM**	**89.4(5.2)**	**98.1(3.7)**	**91.7(4.9)**
	Mink.(r=2)	K-NN=3, CM	86.3(5.2)	97.2(4.2)	88.8(4.9)
	Mink.(r=3)	K-NN=1, CM	85.8(5.3)	93.2(5.8)	87.4(4.9)
	GP-CBR	RI, SS, TS, w{.4,.4,.2}	82.4 (5.8) ↓	82.6 (9.5) ↓	85.1 (4.9)↓
	GE-CBR	RI, SS, TS, w{.35,.35,.3}	94.8(3.1) ↑ √	90.6(5.3)↓ √	92.9(3.1)) √
SO	Clark	K-NN=3, CS	77.4(6.6)	94.6(6.5)	82.6(8.9)
	Mink.(r=1)	**K-NN=1, SC**	**88.0(7.4)**	**91.2(8.1)**	**88.9(7.7)**
	Mink.(r=2)	K-NN=1, CS	88.2(6.3)	88.5(11.7)	87.9(11.4)
	Mink.(r=3)	K-NN=1, SP	86.0(9.2)	88.1(10.6)	86.5(12.1)
	GP-CBR	RI, GR, RS, w{.4,.4,.2}	75.2(11.2) ↓	71.3(9.1) ↓	74.1(9.6) ↓
	GE-CBR	RI, SR, TS, w{.2,.2,.6}	85.7(9.1) √	89.2(8.5) √	86.7(8.5) √

***Dataset* MF.**

- Els millors resultats en les estadístiques són els proposats per la funció proposada per l'híbrid CBR-GE.
- Hi ha millora significativa (*t-student* amb un nivell de confiança del 95 %) en l'estadística d'especificitat respecte la millor funció de similitud de propòsit general i la funció CBR-GP.
- El % d'encerts de la funció CBR-GE és signicativament millor respecte la funció CBR-GP, però no respecte la millor funció de propòsit general.
- El % de sensitivitat de la funció CBR-GP és signicativament millor respecte la funció CBR-GE i les funcions de propòsit general.

***Dataset* BI.**

- Els millors resultats en les estadístiques són els proposats per la funció Minkowski (r=2).
- Hi ha millora significativa (*t-student* amb un nivell de confiança del 95 %) respecte les funcions CBR-GE i CBR-GP.
- La funció CBR-GE té les desviacions típiques més petites.
- La funció CBR-GP proporciona millors resultats que la CBR-GE.

***Dataset* IO.**

- Els millors resultats en les estadístiques són els proposats per la funció proposada per l'híbrid CBR-GE i la funció de Minkowski (r=1).
- El % de sensitivitat de la funció CBR-GE és significativament millor (*t-student* amb un nivell de confiança del 95 %) respecte la millor funció de similitud de propòsit general i la funció CBR-GP.
- El % d'especificitat de la funció Minkowski (r=1) és significativament millor (*t-student* amb un nivell de confiança del 95 %) respecte la millor funció CBR-GE i la funció CBR-GP.
- En el % d'encerts no hi ha millora significativa.
- La funció CBR-GE té les desviacions típiques més petites.

***Dataset* SO.**

- Els millors resultats en les estadístiques són els proposats per la funció Minkowski (r=2).
- No hi ha millora significativa (*t-student* amb un nivell de confiança del 95 %) respecte la funció CBR-GE.
- Hi ha millora significativa (*t-student* amb un nivell de confiança del 95 %) de la funció CBR-GE respecte la funció CBR-GP.
- La funció Minkowski (r=2) té les desviacions típiques més petites.

Tenint en compte tot això, se n'extreuen les reflexions següents:

- Les funcions del CBR-GE s'adapten generalment millor que les altres ja que disposen de la desviació típica més petita gairebé sempre.
- La funció trobada pel CBR-GE 'guanya' en els *datasets* HS i IO.
- La funció Minkowski funciona millor en el *dataset* BI.

- En els *datasets* TA, WS, MF i SO, els resultats són similars.
- El modelatge proposat no resol correctament el domini del MX perquè no és linealment separable. Caldria canviar el modelatge de la gramàtica.
- Independentment del nombre d'atributs i mostres, els sistemes CBR-GP i CBR-GE s'han entrenat en les mateixes condicions. Una adaptació millor dels paràmetres i un increment de l'entrenament permetria possiblement millorar els resultats.

Per tant, els resultats proporcionats per la funció CBR-GE són en general millors que els de la funció CBR-GP i, fins i tot, alguns cops millors que les funcions de propòsit general. Aquests resultats positius són deguts a les restriccions imposades a través de la gramàtica. A continuació s'analitza la coherència de la funció proposada per analitzar si realment la funció proposada és millor.

Coherència de les funcions de similitud trobades

Per poder saber si les funcions de similitud proporcionades pels sistemes híbrids són fiables, es fa necessari la presència d'un expert del domini, a partir del qual es puguin treure conclusions respecte la funció que s'ha construït. Com que no disposem d'un expert en els diferents *datasets*, ens limitarem a analitzar el 'sentit' i 'coherència' de les funcions de similitud.

Dels *datasets* anteriors ens centrarem en aquells on les noves funcions de similitud han millorat a priori els resultats. Les funcions trobades pel CBR-GP són: TA - Eq. 11.4, HS - Eq. 11.5, MF - Eq. 11.6, IO Eq. 11.7). D'altra banda, les funcions del GE-CBR són: TA - Eq. 11.8, HS - Eq. 11.9, MF - Eq. 11.10, IO - Eq. 11.11.

Un dels grans avantatges, ja descrits, del CBR-GE respecte el CBR-GP és la capacitat d'introduir restriccions mitjançant una gramàtica BNF que permet a l'usuari fixar una estructura a partir de la qual es realitza l'exploració de la solució. Les equacions 11.4, 11.5, 11.6 i 11.7 al no partir de cap model tenen un aspecte 'estrany' que dificulta molt la tasca de trobar el motiu pel qual funcionen bé. Aquestes funcions són el resultat d'un procés que ha creat una expressió massa especialitzada en les dades, de tal manera, que es fa gairebé impossible poder generalitzar aquesta funció per qualsevol altre nou exemple.

En canvi, les equacions 11.8, 11.9, 11.10 i 11.11 al basar-se en un model i respectar una sèrie de restriccions fixades en la gramàtica, són capaces de generar expressions molt més fàcils d'interpretar i, conseqüentment, es fa possible extrapolar el model un cop s'han analitzat les diferents contribucions i relacions dels atributs. En aquest sentit, pot veure's que l'estructura que modela el comportament de la cerca permet trobar una funció semblant a la Minkowski. La GE tendeix a introduir els atributs que són importants per comparar, així com assignar un criteri de la seva rellevància a la contribució de la similitud. Per tant, a partir d'això l'expert pot conèixer els factors que són rellevants i reafirmar el significat de la funció. D'aquesta manera pot concloure's que les funcions CBR-GE són molt més fàcils d'entendre i validar que les generades pel CBR-GP.

$$f(case\_x, case\_y) = \sqrt{(Y_1 + Y_1) - (X_1 - (Y_1 - X_2))} \tag{11.4}$$

$$f(case\_x, case\_y) = (X_2 X_{11} \sqrt{Y_9})^2 - ((X_5 + X_4)\sqrt{X_{12}})) \tag{11.5}$$

$$f(case\_x, case\_y) = Y_2 Y_{10} - (X1^2 + (Y18 * X20)/X7) \tag{11.6}$$

$$f(case\_x, case\_y) = X_5 - (Y_{31} - Y_{28} + Y_{30} Y_{27}) + (Y_{28} - Y_{28} X_{34}) * X_{28} \tag{11.7}$$

$$f(case\_x, case\_y) = 0.5(X_1 - Y_1) + 0.5(X_0 - Y_0) + \sqrt{\|(X_1 - Y_1)\|)} + \sqrt{\|0.9(X_0 - Y_0)\|} \tag{11.8}$$

$$f(case\_x, case\_y) = \|0.3(X_2 - Y_2) + 0.1(X_{11} - Y_{11}) + 0.5(X_{12} - Y_{12})\| \tag{11.9}$$

Problema	**Atributs**	**Instàncies**	$\overline{T_{exe}}$ **del CBR**	$\overline{T_{exe}}$ **del CBR-GP**	$\overline{T_{exe}}$ **del CBR-GE**
TA	3	2500	0.09 sec	173 min (20 gen.)	16 min (15 gen.)
WS	10	699	0.03 sec	660 min (250 gen.)	7 min (25 gen.)
MX	12	2047	0.29 sec	1300 min (250 gen.)	450 min (250 gen.)
HS	15	280	0.01 sec	250 min (250 gen.)	23 min (250 gen.)
MF	22	116	0.01 sec	300 min (250 gen.)	29 min (250 gen.)
BI	25	1027	0.14 sec	3060 min (250 gen.)	290 min (250 gen.)
IO	36	351	0.01 sec	530 min (250 gen.)	30 min (250 gen.)
SO	62	208	0.02 sec	370 min (250 gen.)	38 min (250 gen.)

Taula 11.4: Quadre resum del temps mitjà d'execució de cada *dataset* en el CBR en mode *10-fold stratified Cross-Validation*. També s'inclou el temps i nombre de generacions que tarden les propostes CBR-GP i CBR-GE en trobar la funció de similitud de la configuració representada en la taula 11.3, així com el nombre de generacions que han estat necessàries.

$$f(case\_x, case\_y) = 1.10 * (X4 - Y4) + 0.30 * (X8 - Y8) \tag{11.10}$$

$$\begin{aligned} f(case\_x, case\_y) = (\|(((0.1 * (X_0 - Y_0) + ((0.7 * (X_{10} - Y_{10}))^2 + \\ +1.1 * (X_{19} - Y_{19}))) * 0.9 * (X_{33} - Y_{33})) - (((0.9 * (X_3 - Y_3))^2 + \\ +abs(0.7 * (X_2 - Y_2))) + 0.3 * (X_{13} - Y_{13})))\| + \|0.5 * (X_{24} - Y_{24})\| \\ +\|0.5 * (X_{20} - Y_{20}) + \|0.5 * (X_{19} - Y_{19})\|) \end{aligned} \tag{11.11}$$

Comparativa dels temps d'execucions

El temps d'execució per trobar una funció de similitud depèn del nombre d'atributs i casos del problema, ja que això influeix directament en el temps d'executar el CBR, o el que és el mateix, a l'hora d'avaluar la funció de similitud que representa l'individu. A més casos en la memòria de casos, més comparacions s'ha d'efectuar per cada exemple que es vol classificar. Aquest fet es reflecteix en la taula 11.4, on el temps d'execució del CBR (en mode *10-fold stratified Cross-Validation*) s'incrementa notablement si s'augmenta el volum d'informació del problema.

Un altre aspecte important a analitzar de la taula és el temps empleat en trobar la funció de similitud de la configuració seleccionada com més 'fiable' (vegeu la taula 11.3). Tal com es va explicar, el CBR-GP es caracteritza per tenir una representació basada en arbres que representen directament la funció, fent que no calgui cap procés de transformació per obtenir-la. No obstant, aquesta representació implica una adaptació, control i supervisió dels operadors primaris i secundaris que penalitza el temps necessari per realitzar la seva aplicació. D'altra banda, el CBR-GE fa servir una representació lineal d'enters que separa l'espai de cerca del de solucions. Això li permet aplicar els operadors primaris i secundaris d'una manera eficient ja que no s'ha de verificar cap restricció. En canvi, es requereix d'un procés de transformació per poder obtenir la funció de similitud que representa l'individu.

Què requereix més temps, controlar els operadors o transformar el programa? La taula de resultats parla per si sola, ja que clarament el temps necessari per trobar la funció mitjançant la proposta CBR-GE és molt més eficient, concretament, al voltant de 10 cops més ràpida.

Finalment, comentar que totes les execucions s'han realitzat sobre un clúster format per 8 màquines (P-IV 2.6Ghz amb 1 GB de RAM) gestionades mitjançant *OpenMosix* (openMosix, 2007).

11.8 Conclusions i línies futures

La creació d'un sistema híbrid a partir de dos paradigmes de la intel·ligència artificial tan diferents com són l'Aprenentatge analògic (CBR) i la Computació evolutiva (GP i GE), proporciona una nova dimensió a l'hora de solventar els problemes.

Fins ara en el CBR era necessari disposar d'un expert del domini el qual fos capaç de proposar una funció de similitud per tal de poder comparar dos casos i poder recuperar-los de la memòria de casos per tal d'executar el procés d'analogia. Si s'incorpora un element capaç de generar funcions de similitud que després són avaluades pel CBR, aleshores la figura d'aquest expert ja no és tan crítica perquè el sistema pot cercar aquesta funció de manera autònoma.

No obstant, la figura de l'expert sempre és important tenir-la en compte ja que encara que no sàpiga proposar-nos una funció, ens pot ajudar a configurar el sistema per tal de trobar abans la solució com per exemple indicar quins atributs i operadors poden ser necessaris o establir relacions.

Tot i semblar que aquest sistema és capaç de resoldre de manera autònoma els problemes, té un desavantatge: l'espai de cerca on busca és molt gran, fet que pot conduir a convertir el problema en *NP-hard*, és a dir, que sigui impossible resoldre'l amb els recursos computacionals disponibles. Per tant, cal avaluar la possibilitat d'introduir restriccions per reduir aquest espai, i tenir més probabilitats d'èxit. Per exemple, no té sentit generar funcions on només hi ha atributs que fan referència a un cas, o funcions que realitzen operacions entre atributs semànticament diferents. Al mateix temps, introduir restriccions implica el desavantatge que pot condicionar el cicle de l'algorisme evolutiu, per exemple l'aplicació dels operadors genètics, fent que el procés d'avaluació sigui més lent. Aquest problema es compensa pel fet que tot i requerir més temps en fer una generació, de manera global es necessita menys generacions ja que es treballa amb individus que tenen una mínima qualitat, la qual està fixada mitjançant les restriccions. L'altre problema que apareix a l'aplicar restriccions és que es poden perdre solucions de manera accidental, per això és molt important com es fixen les restriccions.

La proposta CBR-GP permet trobar funcions de similitud ad hoc al domini, la gran dificultat que presenta és la complexitat en introduir restriccions perquè no és trivial la seva incorporació en el tractament dels operadors primaris i secundaris. D'altra banda, la proposta CBR-GE incorpora el concepte de restricció de manera inherent gràcies a la definició d'una gramàtica BNF que dirigeix l'espai de cerca. A més a més, a diferència de la GP no té cap cost vinculat a l'hora de mantenir les restriccions quan s'apliquen els operadors.

L'avaluació del sistema CBR-GE ha proporcionat millors resultats que CBR-GP, possiblement perquè que busca en un espai de solucions més reduït que li permet trobar una bona solució amb més èxit. A més, tot i que incorpora restriccions, és molt més ràpid degut a l'estratègia que fa servir per representar les funcions. Per tant, la proposta CBR-GE és més eficaç, més eficient, més fàcil introduir restriccions, i els individus que genera són més fàcils d'analitzar. Les contribucions d'aquest capítol es troben publicades als articles següents:

- A. Fornells, J. Camps, E. Golobardes i J.M. Garrell. *Comparison of strategies based on evolutionary computation for the design of similarity functions.* Al llibre *Artificial Intelligence Research and Development*, volum 131, planes 231-238. IOS Press, 2005.

- A. Fornells, J. Camps, E. Golobardes i J.M. Garrell. *Incorporación de conocimiento en forma de restricciones sobre algoritmos evolutivos para la búsqueda de funciones de similitud.* Al *IV Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*, planes 397-404. Thomson, 2005.

Les línies futures es centren en continuar potenciant la proposta CBR-GE de diferents maneres:

- Avaluar el funcionament de CBR-GE fent servir el model semàntic de mapeig de les funcions de similitud que garanteix que al menys hi ha una còpia de cada atribut.

- Definir noves maneres d'avaluar els individus.
- Ampliar l'estudi a més *datasets*.
- Incorporar millores en el cicle de la GE per tal de millorar la diversitat de la població.
- Avaluar l'aplicació d'altres operadors de creuament i operados secundaris.
- Avaluar una paral·lelització en la cerca de les funcions mitjançant un mapeig simultani sobre diferents gramàtiques.

Resum

El CBR és un paradigma d'aprenentatge analògic capaç de resoldre problemes nous basant-se en problemes prèviament resolts. Com va explicar-se en el capítol 2.2, un dels punts més crítics és la comparació dels casos en la fase de recuperació mitjançant una funció de similitud, la qual estableix la semblança entre aquests. El problema radica en la dificultat de definir com es comparen dos casos, ja que cada problema té el seu propi domini i les seves peculiaritats i, per tant, no existeix una funció de similitud universal. Aquesta funció ha de ser proposada per gent experta en el domini, però això no sempre és possible perquè els dominis reals normalment no es coneixen totalment, i les relacions entre els seus atributs no són trivials. Caldria algun mecanisme que ajudés a l'expert a definir com comparar els casos.

Aquest capítol ha presentat dues propostes per millorar els treballs previs que s'havien fet anteriorment en el grup per a la definició de funcions de similitud ad hoc al domini. Un dels principals problemes de l'anterior enfocament era que l'espai de cerca a buscar era massa gran, fent que fos molt difícil trobar la solució desitjada. Per aquest motiu, s'ha plantejat incorporar restriccions en el procés de cerca de dues maneres. D'una banda, s'ha proposat introduir restriccions sintàctiques i semàntiques en la part de la GP que cerca les funcions. Tot i que això permet reduir l'espai de cerca, la seva aplicació no és trivial al veure's afectats els operadors primaris i secundaris. D'altra banda, s'ha presentat un nou enfocament on el paper de la GP es canvia per la GE. El gran avantatge d'aquest nou punt de vista és la capacitat transparent per introduir restriccions.

Un cop plantejades les dues propostes apareix la pregunta inevitable: quin enfocament és millor, CBR-GP o CBR-GE? Per avaluar això s'ha realitzat una bateria de proves sobre un conjunt de *datasets* de diferents característiques de l'*UCI repository* i dels projectes FIS i HRIMAC per avaluar la resposta segons: (1) el grau de comprensibilitat de la funció, (2) el seu rendiment, (3) i el cost de trobar-la. Els resultats han mostrat clarament que l'enfocament CBR-GE proporciona millors resultats que l'enfocament CBR-GP i, a més a més, la precisió de les funcions és igual o millor que les funcions tradicionals. D'altra banda, el sistema CBR-GE troba la funció en molt menys temps que l'enfocament CBR-GP. No obstant, tot té sempre un 'però'. En aquest cas el 'però' és que la definició de la gramàtica ha de ser el suficientment flexible per trobar la solució, i el suficientment estricte per reduir l'espai de cerca. Això vol dir que una incorrecta definició tindrà com a resultat la no definició de cap funció.

Capítol 12

Plataformes JACK & BRAIN

Les propostes dels sistemes híbrids CBR-GP i CBR-GE cobren vida en aquest capítol sota les plataformes JACK i BRAIN respectivament. Ambdues plataformes tenen la finalitat de trobar de manera automàtica funcions de similitud específiques per un domini. Per fer-ho, apliquen algorismes basats en la Computació evolutiva, GP i GE, que exploren un espai de cerca on cada individu és avaluat sobre un CBR. La plataforma JACK implementa l'híbrid CBR-GP en Java, i la plataforma BRAIN implementa l'híbrid CBR-GE en C++. El motiu de l'elecció dels llenguatges de programació està relacionada amb els projectes duts a terme dins del GRSI.

12.1 Motivació dels desenvolupaments

La finalitat d'aquest capítol és presentar les plataformes JACK i BRAIN dissenyades i implementades durant la recerca. Ambdues es fan servir com a sistemes capaços de definir de manera automàtica una funció de similitud específica per un domini. En els dos casos s'aprofita la capacitat de la GP i la GE per generar funcions, i la capacitat d'avaluació del CBR per determinar la validesa de les funcions generades.

La plataforma JACK (*JAva Case base reasoning Kernel*) respon a la necessitat d'estudiar el funcionament del CBR i el comportament de l'híbrid CBR-GP proposat en (Golobardes et al., 2001) (Camps et al., 2003) per tal de millorar els resultats obtinguts en el projecte HRIMAC. A més a més, JACK té com a novetat respecte la primera proposta de l'híbrid la incorporació de restriccions sintàctiques i semàntiques sobre la generació de funcions de similitud. Tot i que Java no és un llenguatge òptim per implementar algorismes basats en la Computació evolutiva, es va fer en aquest llenguatge degut a que es volia integrar dins del projecte KEEL (TIC2002-04036-C05-03).

D'altra banda, i a partir de l'experiència de JACK, l'aproximació CBR-GE es va implementar sota una nova plataforma anomenada BRAIN (*hyBRid system to discover And Improve similarity fuNctions*). Independentment de les virtuts de la GE respecte la GP que s'han comentat en el capítol anterior, aquesta nova plataforma és molt més ràpida que la anterior gràcies a la seva implementació en C++.

Els punts següents descriuen el disseny a alt nivell d'ambdues plataformes, així com les eines utilitzades pel seu desenvolupament.

12.2 Disseny

La figura 12.1 descriu el diagrama de mòduls per les plataformes JACK i BRAIN. Els dos dissenys es basen en el mateix esquema, amb l'única diferència del mòdul de Computació evolutiva que fan servir: la GP en la plataforma JACK i la GE en la plataforma BRAIN. En ambdós casos els

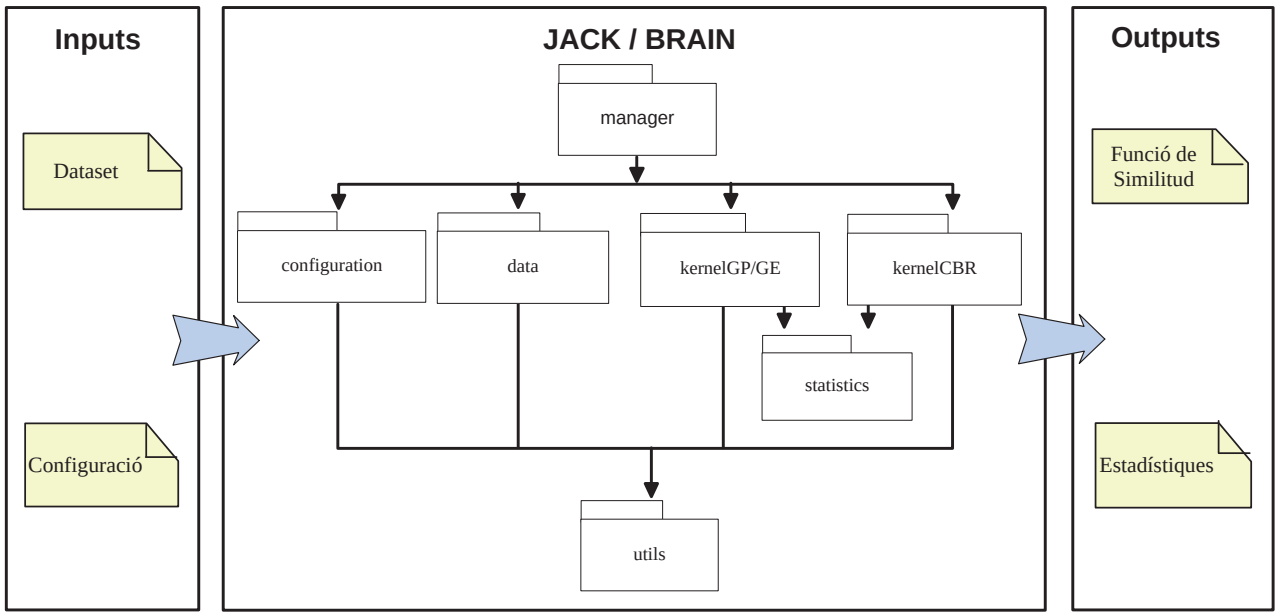

Figura 12.1: Diagrama de blocs general de les plataformes JACK i BRAIN.

sistemes poden funcionar en mode CBR o mode híbrid segons la configuració del mode d'execució. Als apartats següents es descriu a alt nivell la finalitat de cadascun dels elements i mòduls que formen part del diagrama de la figura 12.1.

12.2.1 Elements d'*Input* i *Output*

A la figura 12.1 es distingeixen dos tipus diferents de fitxers d'*inputs*:

- **Dataset.** És l'especificació del problema que es vol resoldre.
- **Configuració.** Defineix el valor dels arguments de cadascun dels elements que intervenen en l'execució.

De la mateixa manera, pels dos sistemes es generen els mateixos *outputs*, no obstant, segons el mode d'execució els resultats generats canvien:

- **Funció de similitud.** Aquesta informació només es genera en mode híbrid, i representa la millor funció de similitud que s'ha trobat per resoldre el problema.
- **Estadístiques de classificació.** Els resultats de classificació es generen tant en mode CBR com en mode híbrid. En el cas del CBR representa les estadístiques de classificació dels exemples de test: el % d'encerts, el % d'error, el % de no classificats, el % de sensitivitat, i el % d'especificitat. A més a més, si s'executa en mode *Cross-Validation* també es generen les mitges i desviacions estàndards associades. En canvi, en el cas de l'híbrid representa les estadístiques del CBR sobre els exemples de test fent servir com a funció de similitud l'individu amb millor *fitness*.

12.2.2 Especificació dels mòduls

En aquest apartat es defineixen les finalitats i funcionalitats de cadascun dels mòduls que integren les plataformes JACK i BRAIN.

12.2.2.1 Mòdul *manager*

És el mòdul principal del sistema i s'encarrega de coordinar la resta de mòduls. En funció dels paràmetres de configuració, ja sigui per fitxer o per línia de comandes, instancia els elements necessaris per realitzar una execució en mode CBR o en mode híbrid.

12.2.2.2 Mòdul *configuration*

La finalitat d'aquest mòdul és gestionar la configuració de tots els paràmetres del sistema. Les seves principals tasques són:

- Carregar/Guardar la informació de configuració emmagatzemada en un fitxer en format XML (*eXtensible Markup Language*) (WebXML, 2007).
- Modificar la configuració de fitxer per línia de comandes.
- Proporcionar a la resta de mòduls els paràmetres de la seva configuració.

12.2.2.3 Mòdul *data*

El mòdul de dades gestiona les dades que representen el problema del domini sobre el qual es treballa. Les seves principals tasques són:

- Carregar les dades del problema emmagatzemades en el format estàndard ARFF (Witten i Frank, 2000).
- Aplicar sobre les dades algunes de les següents operacions de preprocessament:
  (a) Normalització per rang, diferència i escalat decimal.
  (b) Ponderació d'atributs mitjançant PCA o Correlació Mostral.
  (c) Selecció de característiques.
  (d) Gestió de valors desconeguts.
  (e) Correcció d'atributs erronis fora de rang.

12.2.2.4 Mòdul *kernelCBR*

El mòdul kernelCBR té l'objectiu de simular el cicle de vida del CBR mitjançant la coordinació de les diferents fases. Les funcionalitats que implementa per cada element i fases són:

- Memòria de casos del CBR:
  (a) Representació d'un cas genèric.
  (b) Inicialització de la memòria a partir de fitxers on s'indica la contribució del % *train* i % de test.
  (c) Inicialització de la memòria a partir d'un únic fitxer.
  (d) Inicialització de la memòria entrenant-la a partir d'un únic fitxer.
  (e) Aplicació d'operacions de processament sobre les memòries *train* i test.
  (f) Es guarden les dades preprocessades de *train* i test en un format destí.
  (g) Adaptació de les dades del problema segons la funció de similitud a utilitzar.
- Funcions de similitud:

(a) Definició de les funcions de similitud de propòsit general: Minkowski (r=1, 2, 3), Manhattan, Clark i Cosinus.

(b) Preprocessament de funcions de similitud a partir d'expressions regulars proposades pels nuclis de Computació evolutiva.

- La fase de recuperació:

(a) Recuperació dels casos mitjançant l'aplicació iterativa de diferents funcions de similitud.

(b) Recuperació d'1 a N casos similars (KNN - *K Nearest Neighbourn*).

(c) Recuperació de la informació a partir d'un cert llindar (paràmetre) de confiança.

- La fase d'adaptació:

(a) Es proposa la solució mitjançant la votació dels N casos recuperats.

- La fase de revisió.

(a) Validació de la solució a partir d'un mínim de semblança.

- La fase d'emmagatzematge:

(a) Política DiffTest.

(b) Política DiffClass.

(c) Política DiffSim.

12.2.2.5 Mòdul *kernelGP*

El mòdul kernelGP té per objectiu simular el cicle de vida de la GP mitjançant la coordinació de les diferents fases. Les funcionalitats que implementa per cada element i fases són:

- Representació dels individus:

(a) Individus representats físicament com arrays, però lògicament com arbres binaris.

(b) Terminals definits: variables (enter/real) i constans predefinides.

(c) Funcions definides: add, sub, mul, div, sin, cos, tan, exp, log ||, ln ||, pow, abs, sqr i sqrt.

(d) Assignació dels nodes terminals i funcions disponibles.

- Inicialització de la població:

(a) Inicialització *grow*.

(b) Inicialització *full*.

(c) Inicialització *ramped full*.

(d) Inicialització *ramped grow*.

(e) Inicialització *ramped half and half*.

(f) Control d'inicialització amb restriccions de nivell 1 opcional.

- Avaluació dels individus:

(a) Transformació de l'individu en una expressió aritmètica avaluable pel CBR.

(b) Comunicació i coordinació amb el CBR.

(c) Execució del cicle de vida del CBR en mode normal o *Cross-Validation*.

(d) Avaluació del *fitness* de la funció de similitud pel CBR basada en el % d'encerts, el % d'errades i el % de no classificats .

(e) Avaluació del *fitness* de la funció de similitud pel CBR basada en el % de sensitivitat, el % d'especificitat i el % dels no classificats.

- Selecció dels individus de la població:

(a) Selecció aleatòria.

(b) Selecció per ruleta.

(c) Selecció per torneig de 'P' elements.

(d) Control de *bloat* basat en la longitud.

- Operadors primaris i secundaris:

(a) Creuament.

(b) Reproducció.

(c) Mutació.

(d) Control dels operadors amb restriccions de nivell 1 i 2 opcional.

- Polítiques de reemplaçament:

(a) Generacional.

(b) *Steady-State*.

- Millores

(a) Elitisme.

(b) Reemplaçament dinàmic de la població.

12.2.2.6 Mòdul *kernelGE*

El mòdul kernelGE té l'objectiu de simular el cicle de vida de la GE mitjançant la coordinació de les diferents fases. Les funcionalitats que implementa per cada element i fases són:

- Representació dels individus:

(a) Individus representats com arrays d'enters.

(b) Definició de la gramàtica que modela els elements que formen part de la funció de similitud.

- Inicialització de la població:

(a) Inicialització aleatòria.

(b) Inicialització pseudo-aleatòria per garantir un nombre mínim d'elements.

- Avaluació dels individus:

(a) Mapeig de l'individu en la gramàtica BNF per generar una expressió aritmètica avaluable pel CBR.

(b) Comunicació i coordinació amb el CBR.

(c) Execució del cicle de vida del CBR en mode normal o *Cross-Validation*.

(d) Avaluació del *fitness* de la funció de similitud pel CBR basada en el % d'encerts, el % d'errades, i el % dels no classificats.

(e) Avaluació del *fitness* de la funció de similitud pel CBR basada en el % de sensitivitat, el % d'especificitat, i el % dels no classificats.

- Selecció dels individus de la població:

(a) Selecció aleatòria.

(b) Selecció per ruleta.

(c) Selecció per torneig de 'P' elements.

(d) Control de *bloat* basat en la longitud.

- Operadors primaris i secundaris:

(a) Creuament.

(b) Reproducció.

(c) Mutació.

(d) Control opcional dels operadors per garantir un nombre mínim d'elements.

- Polítiques de reemplaçament:

(a) Generacional.

(b) *Steady-State*.

- Millores:

(a) Elitisme.

(b) Reemplaçament dinàmic de la població.

12.2.2.7 Mòdul *statistics*

És el mòdul encarregat de recopilar la informació de les execucions per tal de generar els informes de resultats. Les seves tasques són:

- Recopilar informació sobre l'execució actual:

(a) Càlcul del % d'encerts, % d'errors i % dels no classificats.

(b) Càlcul del % de sensitivitat i % d'especificitat.

(c) Temps parcial de cada fase i total.

- Generació d'informes:

(a) Sortida per pantalla o fitxer.

(b) Generació d'arbres amb LaTeX de les funcions de similitud.

12.2.2.8 Mòdul *utils*

El mòdul *utils* es composa per un conjunt de llibreries de propòsit general que implementen funcionalitats habituals realitzades per la resta de mòduls. Les llibreries que engloba són:

- Accés de fitxers.
- Control de temps.
- Gestió de nombres aleatoris.
- Gestió del tipus abstracte *Date*.
- Conversions entre tipus de dades
- Operacions amb vectors i matrius.

12.3 Implementació i eines de desenvolupament emprades

El disseny explicat a l'apartat anterior separa clarament les parts i funcionalitats de cadascun dels diferents mòduls que componen les dues plataformes. Per tal de permetre una fàcil integració dels mòduls i incorporació de futures funcionalitats, totes les parts del sistema es relacionen mitjançant interfícies. Això permet separar de manera independent les funcionalitats que ha d'acomplir el mòdul, respecte de com les ha d'implementar.

La plataforma JACK s'ha implementat mitjançant el llenguatge de programació Java degut a la seva possible futura incorporació en el projecte KEEL. En total, la plataforma JACK està composada per unes 110 classes que defineixen tots els mòduls explicats anteriorment.

D'altra banda, la plataforma BRAIN s'ha implementat mitjançant el llenguatge de programació C++ amb la finalitat d'optimitzar el temps d'execució, ja que JACK era un executable interpretat. En total la plataforma BRAIN està composada per unes 95 classes que defineixen tots els mòduls explicats anteriorment.
A continuació es detallen les eines que s'han fet servir:

- Anàlisi i disseny
  - Borland Together 6.1. Eina que permet realitzar el procés d'anàlisi i disseny mitjançant la metodologia UML (Together, 2007).
- Implementació de la plataforma JACK
  - Jdk 1.4.2. Compilador de Java (WebJava, 2007).
  - JavaCC 2.0. Eina pel desenvolupament de *parsers* descendents a alt nivell per Java (WebJavaCC, 2007).
  - API JDom 8.0. Eina per la gestió d'XML amb Java (JDOM, 2007).
  - Netbeans 3.6.1. Entorn multiplataforma de desenvolupament i compilació per Java (Netbeans, 2007).
- Implementació de la plataforma BRAIN
  - Lex/Yacc. Llibreries per la construcció de parsers mitjançant C/C++ (LexYacc, 2007).
  - C++BuilderX 1.0.0.16. Entorn multiplataforma de desenvolupament i compilació per C++ (BuilderX, 2007).

Resum

El capítol ha presentat a alt nivell els mòduls de les plataformes JACK i BRAIN derivades de la recerca en l'àmbit del disseny de funcions de similitud ad hoc un domini.

L'estratègia CBR-GP va ser la primera en desenvolupar-se, i va fer-se sota la plataforma JACK. Tot i que hauria d'haver-se implementat en C++ per permetre una ràpida execució, es va optar per una implementació en Java a l'haver la possibilitat d'incorporar-la dins el projecte KEEL.

En canvi, l'altra estratègia basada en CBR-GE va implementar-se en C++ sota la plataforma BRAIN amb la finalitat d'incrementar la velocitat d'execució. Aquesta última plataforma és la que s'ha fet servir per la cerca de funcions de similitud específiques dins del projecte HRIMAC, gràcies a la potència que ofereix la possibilitat d'afegir restriccions, i a la seva velocitat d'execució.

Part IV

Cloenda

Capítol 13

Treball realitzat, conclusions i línies futures

Al llarg de la tesi s'han afrontat diferents reptes. D'una banda, la creació d'un marc integrador de les capacitats *Soft-Computing* i de *Knowledge Discovery* de SOM dins del cicle del CBR. D'altra banda, la utilització de les capacitats *Soft-Computing* i de *Knowledge Discovery* de la GE pel disseny de funcions de similitud i esquemes de cooperació específics per un domini. Aquest capítol recull la feina i contribucions realitzades per fer una reflexió global de la tesi. Finalment, es proposen algunes línies futures de recerca.

13.1 Marc de la tesi

El GRSI centre les seves línies de recerca en els paradigmes del CBR, la CE i el *Soft-Computing* en general per abordar problemes de classificació i diagnòstic, destacant els entorns mèdics i telemàtics. Això ha motivat que el desenvolupament de la tesi s'hagi vist influenciat per aquests paradigmes i dominis.

La finalitat de la present tesi ha estat la creació d'un marc integrador de les capacitats *Soft-Computing* i de *Knowledge Discovery* de SOM dins de tots els aspectes rellevants del CBR, per tal de potenciar el desenvolupament de sistemes CBR capaços de gestionar grans volums de dades i, a més a més, siguin robusts al soroll i a la incertesa de les dades. Això ha portat a estudiar i analitzar el seu impacte en l'organització de la memòria de casos, la fase de recuperació, la fase d'adaptació, la fase de revisió i la fase d'emmagatzematge. Tot aquest estudi ha conclòs amb el desenvolupament d'un entorn anomenat SOMCBR (*Self-Organizing Map in a Case-Based Reasoning system*), el qual ha permès l'avaluació sobre diferents *datasets* provinents de l'*UCI Repository* i dels diferents projectes on s'ha emmarcat la tesi. D'altra banda, els requeriments del projecte HRIMAC van fer necessari iniciar dues línies de recerca basades en l'optimització de funcions pel disseny de mètriques de distància i per la definició d'esquemes de cooperació. Ambdós enfocaments s'han abordat des de l'enfocament de la GE gràcies a les seves capacitats per limitar l'espai de cerca mitjançant restriccions transparents al sistema.

Aquest capítol pretén recopilar les diferents contribucions realitzades al llarg de la tesi, així com resumir els diferents 'entregables' que s'han obtingut: plataformes i articles. Finalment, es comenten les principals línies de recerca de cadascun dels camps estudiats.

13.2 Marc integrador de les capacitats de *Soft-Computing* i de *Knowledge Discovery* de SOM en el CBR

Les fases del CBR giren al voltant de l'experiència que té el sistema, o en altres paraules, sobre la memòria de casos. A més compacta, reduïda i representativa sigui la memòria, el rendiment és millor perquè el sistema és capaç de trobar la informació que es demana i, a més a més, de manera eficient. No obstant, al tractar amb dominis reals que presenten grans volums de dades, i que poden presentar incertesa i coneixement parcial, aquesta situació 'idílica' es fa gairebé impossible d'aconseguir. Això ha motivat l'aparició recent d'una nova línia de recerca anomenada *Soft Computing and Intelligent Information Retrieval* (Crestani i Pasi, 2000; Cordón i Herrera, 2003), la qual té per objectiu l'aplicació de tècniques *Soft-Computing* per dotar als sistemes de capacitats per tractar amb aquest tipus de coneixement. Concretament, aquest és el context en el qual s'ha situat la tesi.

SOM és una tècnica de clustering no supervisada que projecta l'espai original de les dades en un altre més reduït per tal definir agrupacions de dades segons la seva similitud. Cada agrupació es representa a través d'un vector director, el qual representa el valor mig que té cada atribut d'un cas dins del clúster. Les capacitats de *Soft-Computing* i de *Knowledge Discovery* de SOM per tractar coneixement complex, incert i parcialment vertader han fet que sigui una de les tècniques de clustering més utilitzades, ja que és capaç d'identificar les relacions amagades a les dades per adaptar-se millor a elles. És per aquest motiu que s'ha triat SOM com la tècnica *Soft-Computing* més adient per organitzar la memòria, amb la finalitat de disposar d'una jerarquia que ajudi al sistema a discernir entre els casos rellevants i els sorollosos.

La combinació dels paradigmes del CBR i de SOM s'ha abordat a la literatura en diverses ocasions amb bons resultats tal com va comentar-se al principi de la tesi (Hongkyu i Ingoo, 1996; Jha et al., 1999; Essam i Ahmed, 2001; Kim i Han, 2001; Chang i Lai, 2005; Mujica et al., 2005). No obstant, aquestes combinacions s'enfocaven sempre de manera puntual en alguna de les fases, i sempre de manera específica per algun problema concret. En canvi, aquesta tesi es volia anar més enllà amb la definició d'un marc integració de les capacitats de SOM en tots els aspectes del CBR i, d'aquesta manera, potenciar la construcció de sistemes més robusts i fiables davant de dades complexes i incertes. El resultat ha estat la construcció del sistema SOMCBR, on cada fase s'ha potenciat amb el coneixement que SOM descobreix de les dades tal com mostra la figura 13.1:

Clusterització de la memòria de casos. El punt d'entrada de SOM dins del CBR és la memòria de casos, concretament, mitjançant la seva clusterització. Això permet definir una jerarquia on els casos s'agrupen en grups segons les similituds per abordar dues fites. D'una banda, millorar el temps de la fase de recuperació perquè el sistema només buscava en una part de la memòria. D'altra banda, permetre al sistema descartar casos redundants que només introdueixen soroll.

El resultat d'aquest treball va ser la presentació del primer article del SOMCBR en el *8th European Conference on Case-Based Reasoning* titulat *Unsupervised Case Memory Organization: Analysing Computational Time and Soft Computing Capabilities* (Fornells et al., 2006b).

Metodologia per la definició de la recuperació òptima de casos. Un cop descoberts els beneficis potencials del sistema gràcies a la seva capacitat de reduir l'espai de cerca en favor dels casos potencialment útils, va arribar la qüestió següent: quants clústers i casos agafo? sota quines condicions? Això va motivar la definició del mapa d'estratègies com l'esquema a través del qual es podien representar totes les possibles situacions, les quals després calia estudiar. Fruït del gran ventall de configuracions possibles i de la complexitat d'interpretar

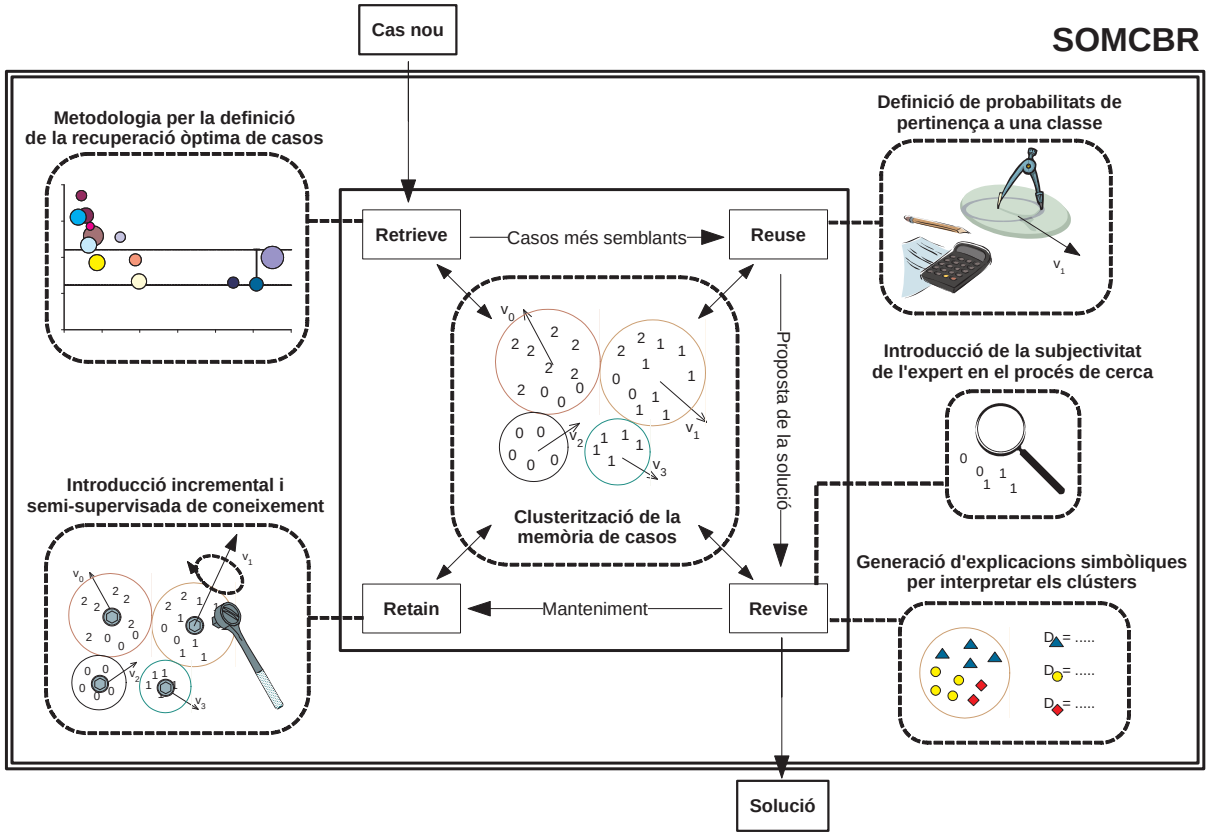

Figura 13.1: SOMCBR és un marc d'integració de les capacitats *Soft-Computing* i de *Knowledge Discovery* de SOM en el CBR per la construcció de sistemes més robusts i fiables davant de dades complexes i incertes.

els resultats, es va proposar un esquema de comparació basat en una representació 2D del rànquing de l'èxit del mètode respecte la millora de temps que aporta. Aquest nou esquema va permetre avaluar les configuracions d'una manera àgil i precisa. D'altra banda, la qualitat dels clústers per representar les dades està directament vinculat amb la geometria de les dades. Per tant, això va fer que a la recerca s'obrís una línia addicional basada en l'estudi de la geometria de les dades a partir de les mètriques de complexitat. Els resultats van ser la definició del mapa de complexitats, el qual ens permet segmentar l'estudi de les estratègies segons la tipologia de les dades.

Els resultats d'aquest treball es troben recopilats a dos articles. A l'article *Measuring the applicability of Self-Organizing Maps in a Case-Based Reasoning system* (Fornells et al., 2007b) es van publicar les primeres reflexions respecte la relació entre el SOMCBR i la complexitat de les dades dins del *3rd Iberian Conference on Pattern Recognition and Image Analysis*. Posteriorment, a l'article *A methodology for analyzing the case retrieval from a clustered case memory* (Fornells et al., 2007c) presentat al *7th International Conference on Case-Based Reasoning* va presentar la metodologia de selecció basada en els mapes d'estratègia, l'*scatter plot*, i el mapa de complexitat detallats al capítol 5. A més a més, aquest últim treball va estar nominat al *Best Paper Award* del congrés. D'altra banda, a l'article *Revisión sobre métricas de complejidad en el modelado de clústers de un sistema CBR* (Macià et al., 2007) va replantejar-se la feina anterior però des d'una perspectiva més de caire general i divulgativa al *V Taller nacional de minería de datos y aprendizaje*.

Estimació de probabilitats de pertinença a una classe. Tot i que amb SOM els casos s'agrupen segons la seva similitud, pot succeir que certs casos ambigus siguin recuperats degut a

la incertesa i complexitat del domini. Per aquest motiu, aquest punt s'ha centrat en aprofitar les relacions entre els clústers i els casos per definir graus de pertinença a les classes dels casos recuperats. Això permet al sistema identificar quins casos són més robusts des del punt de vista de pertinença al clúster i, conseqüentment, són potencialment més fiables per proposar la nova solució. La fiabilitat és un aspecte és vital en dominis tan crítics com per exemple la detecció del càncer de mama del projecte HRIMAC, ja que estan en joc vides humanes.

Els resultats d'aquest treball es troba en procés d'acceptació a la revista *International Journal of Neural Systems* sota el nom de *Management of relations between cases and patterns from SOM for helping experts in breast cancer diagnosis* (Fornells et al., 2007d).

Introducció de la subjectivitat de l'expert en el procés de cerca. El CBR justifica els resultats a partir de la similitud entre el cas nou i els resolts. El problema apareix quan aquest grau de similitud és complex de definir degut a la desconeixença del domini, o bé, a la complexitat de les dades. A més a més, cal tenir en compte que cada persona percep d'una manera diferent la realitat i, conseqüentment, el que és semblant per uns pot no ser-ho per altres. Tots aquests aspectes fan complexa la definició de la mètrica de similitud. Dins d'aquest context, les tècniques de *Relevance Feedback* són estratègies basades en introduir la subjectivitat de l'expert dins del procés de cerca per guiar la cerca segons les seves preferències. En aquesta part de la recerca es va estudiar com integrar aquestes estratègies dins del nucli CBR del projecte HRIMAC.

El resultat del treball va concloure amb la selecció de SOMCBR com el motor de cerca òptim per a implementar aquestes estratègies. El treball va presentar-se amb el títol *Integration of strategies based on Relevance Feedback into a tool for retrieval of mammographic images* (Fornells et al., 2006c) al *7th International Conference on Intelligent Data Engineering and Automated Learning*. L'article va ser seleccionat per realitzar una versió ampliada amb noves millores per la revista *International Journal of Neural Systems*.

Generació d'explicacions simbòliques per interpretar els clústers. La fase de revisió és on l'expert ha de validar si la proposta de solució és correcte o no. Aquesta tasca no és gens fàcil ja que si es conegués la solució, no caldria un sistema basat en CBR que la proposés. Per aquest motiu és important facilitar a l'expert mecanismes per validar el resultat. A banda de les millores de rendiment que SOM ofereix al CBR, les relacions entre els casos són molt útils per l'expert ja que li permeten conèixer perquè determinats casos són seleccionats, així com qui està relacionat amb qui. No obstant, el baix nivell de representació dels vectors directors pot fer difícil la seva comprensió. Per aquest motiu aquesta fita es basava en generar explicacions simbòliques que fossin més fàcils d'entendre per l'expert. Paral·lelament, aquesta nova interpretació dels clústers ha permès una ampliació del mapa d'estratègies basada en la combinació dels criteris de selecció segons la interpretació dels clústers.

Els resultats d'aquest treball es troben publicats a dos articles. D'una banda, a *Explanation of a clustered case memory organization* (Fornells et al., 2007a) publicat al llibre *Artificial Intelligence Research and Development* es presenten les explicacions com a mecanisme de selecció de clústers. D'altra banda, al treball *Data security analysis using unsupervised learning and explanations* publicat al llibre *Innovations in Hybrid Intelligent Systems* s'aborda les explicacions com a elements de suport per la detecció de vulnerabilitats (Corral et al., 2007). Aquest treball estava basat en un treball previ on va avaluar-se SOM com a mecanisme de clustering per aquest domini i on es va proposar una mètrica per avaluar el grau de cohesió dels dispositius telemàtics en un clúster segons un conjunt de paràmetres. Aquest treball es titula *Cohesion factors: improving the clustering capabilities of Consensus* i va presentar-se al *7th International Conference on Intelligent Data Engineering and Automated Learning*.

Introducció incremental i semi-supervisada de coneixement. Un cop el sistema ha resolt i validat el problema nou, arriba el moment on cal decidir si l'experiència és rellevant o no. El resultat d'aquesta decisió poden dividir-se: (1) no guardar res perquè ja es disposa d'aquest coneixement, (2) refinar el coneixement existent o (3) introduir un coneixement nou. El problema en aquesta etapa és que SOM no permet la introducció, la modificació o l'eliminació de coneixement dinàmicament, és a dir, si es volen fer canvis cal reentrenar el mapa des de zero amb la conseqüent despesa computacional. Per aquest motiu aquesta fita va centrar-se en com es podia aprofitar el *feedback* de l'expert sobre la validesa del resultat per permetre actualitzar de manera incremental el coneixement, i evitar així haver de reentrenar el mapa.

El resultat d'aquest treball es troba publicat amb el títol *Case-base maintenance in an associative memory organized by a Self-Organizing Map* (Fornells i Golobardes, 2007) al llibre *Innovations in Hybrid Intelligent Systems*.

13.3 Disseny de funcions de similitud

La motivació d'aquesta part de la recerca va estar provocada pel requeriments del projecte HRIMAC, on calia analitzar la necessitat de definir una mètrica que fos específica per comparar mamografies. Com s'ha explicat, la missió de la funció de similitud és establir el grau de semblança entre dos casos per tal de garantir una certa fiabilitat i precisió a l'hora d'aplicar el CBR. La seva definició és sovint complexa i solen estar implicats els experts del domini, els quals aporten la seva 'expertesa' per tal de validar la precisió. No obstant, la complexitat i incertesa del domini del càncer de mama fa que la definició de la funció per part dels experts sigui gairebé impossible perquè ni entre ells mateixos es posen d'acord a l'hora de percebre les mamografies. Fruït d'això, es va plantejar aplicar les capacitats de *Knowledge Discovery* i de *Soft-Computing* de la Computació evolutiva per tal de definir un esquema de cerca de funcions de similitud específic a un domini.

Tot i que aquesta línia de recerca havia estat abordada prèviament dins del GRSI a través de la GP, l'enfocament tenia un gran defecte: esdevenia *NP-hard* degut a l'immesitat de l'espai de cerca. Això va fer que es busqués un enfocament nou basat en la GE, el qual és un paradigma per definir funcions que permet introduir restriccions de manera transparent.

Els resultats d'aquest treball van ser dos. D'una banda, va presentar-se el sistema al *IV Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados* en el treball *Incorporación de conocimiento en forma de restricciones sobre algoritmos evolutivos para la búsqueda de funciones de similitud* (Fornells et al., 2005b). D'altra banda, la comparació d'aquests resultats respecte l'enfocament inicial basat en la GP van publicar-se en el llibre *Artificial Intelligence Research and Development* sota el treball *Comparison of Strategies based on Evolutionary Computation for the Design of Similarity Functions* (Fornells et al., 2005a).

13.4 Disseny d'esquemes de cooperació

Aquesta línia també va estar motivada des del projecte HRIMAC. La fiabilitat és el punt més crític d'aquest projecte degut a l'element que està en joc: la vida del pacient. Això fa que el sistema tingui que prendre les mesures adients per garantir que el resultat és el més fiable possible. Per aconseguir-ho, es va plantejar d'abordar la millora de la fiabilitat a partir d'esquemes de cooperació per tal definir com una mena de comitè d'experts. Arrel d'això va sorgir la gran qüestió: com es posen d'acord?

L'enfocament que es va plantejar va ser considerar que es disposen de S sistemes experts, on cadascú és expert per predir un cert comportament. A partir d'això, calia cercar una estratègia que

indiqués segons la probabilitat de cada classe interpretada per cada expert, la solució més plausible. Concretament, es va proposar la GE com a mecanisme de cerca del esquema de cooperació que millor representes el 'comitè'.

Els resultats van ser publicats en el *8th International Conference on Enterprise Information Systems* sota el títol *Decision Support System for Breast Cancer Diagnosis by a Meta-learning Approach based on Grammar Evolution* (Fornells et al., 2006a).

13.5 Recull del treball realitzat

La figura 13.2 resumeix de manera esquemàtica les línies de recerca, paradigmes, aplicacions i projectes on s'ha emmarcat la tesi presentada al llarg del present document. Malgrat que la tesi ha tractat tres línies de recerca, el disseny de funcions i la cooperació de sistemes van deixar-se a un segon pla un cop van assolir-se els requeriments del projecte HRIMAC. Fruït d'aquestes línies van desenvolupar-se les plataformes BRAIN i MGE per definir automàticament funcions i esquemes específics mitjançant la capacitat de cerca de la GP i la GE, i la capacitat d'avaluació del CBR i d'altres paradigmes integrats dins del Weka (WK) (Witten i Frank, 2005).

El 'cos' de la tesi ha girat en torn al desenvolupament d'un marc integrador de les capacitats de SOM per potenciar les fases del CBR, i aconseguir construir sistemes més robusts i tolerants a grans volums de dades complexes i incertes. A banda de l'estudi del CBR i de SOM, s'han estudiat les mètriques de complexitat (MC), les tècniques de *Relevance Feedback* (RF) i l'operador anti-unificació (AU), els quals han sigut un complement de valor afegit molt important per potenciar encara més la integració del coneixement de SOM dins del CBR. El resultat d'això ha estat la definició de la plataforma SOMCBR, la qual està emmarcada dins dels projectes HRIMAC, ANALIA i MID-CBR.

A banda de les tres plataformes desenvolupades, els entregables divulgatius científics resultats de la tesi estan publicats a tretze treballs en diferents formats. El llistat de publicacions, agrupat per temàtiques i ordenat per ordre cronològic, és el següent:

Fase de recuperació.

- A. Fornells, E. Golobardes, J.M. Martorell, J.M. Garrell, E. Bernadó i N. Macià. *A methodology for analyzing the case retrieval from a clustered case memor.* Al *7th International Conference on Case-Based Reasoning*, volum 4626 de LNAI, planes 122-136. Springer-Verlag, 2007. Nominat al millor article del congrés.
- A. Fornells, E. Golobardes, J.M. Martorell, J.M. Garrell, E. Bernadó i N. Macià. *Measuring the applicability of self-organizing maps in a case-based reasoning system.* Al *3rd Iberian Conference on Pattern Recognition and Image Analysis*, volum 4478 de LNCS, planes 532-539. Springer-Verlag, 2007.
- N. Macià, E. Bernadó, A. Fornells, E. Golobardes, J.M. Martorell i J. M. Garrell. *Revisión sobre métricas de complejidad en el modelado de clústers de un sistema CBR.* Al *V Taller nacional de minería de datos y aprendizaje*, 2007. En impremta.
- A. Fornells, E. Golobardes, D. Vernet i G. Corral. *Unsupervised case memory organization: Analysing computational time and soft computing capabilities.* Al *8th European Conference on Case-Based Reasoning*, volum 4106 de LNAI, planes 241-255. Springer-Verlag, 2006.

Fase d'adaptació.

- A. Fornells, E. Golobardes, J.M. Martorell, J.M. Garrell i X. Vilasís. *Management of relations between cases and patterns from SOM for helping experts in breast cancer diagnosis.* A *International Journal of Neural Systems*, 2007. En impremta.

Fase de revisió.

- G. Corral, A. Fornells, E. Armengol i E. Golobardes. *Data security analysis using unsupervised learning and explanations.* Al llibre *Innovations in Hybrid Intelligent Systems*, volum 44. Editors: E. Corchado, J.M. Corchado, i A. Abraham. Springer-Verlag, 2007. En impremta.
- A. Fornells, E. Armengol, i E. Golobardes. *Explanation of a clustered case memory organization.* Al llibre *Artificial Intelligence Research and Development.* IOS Press, 2007. En impremta.
- A. Fornells, E. Golobardes, X. Vilasís i J. Martí. Integration of strategies based on relevance feedback into a tool for retrieval of mammographic images. Al *7th International Conference on Intelligent Data Engineering and Automated Learning*, volum 4224 de LNCS, planes 116-124. Springer-Verlag, 2006.

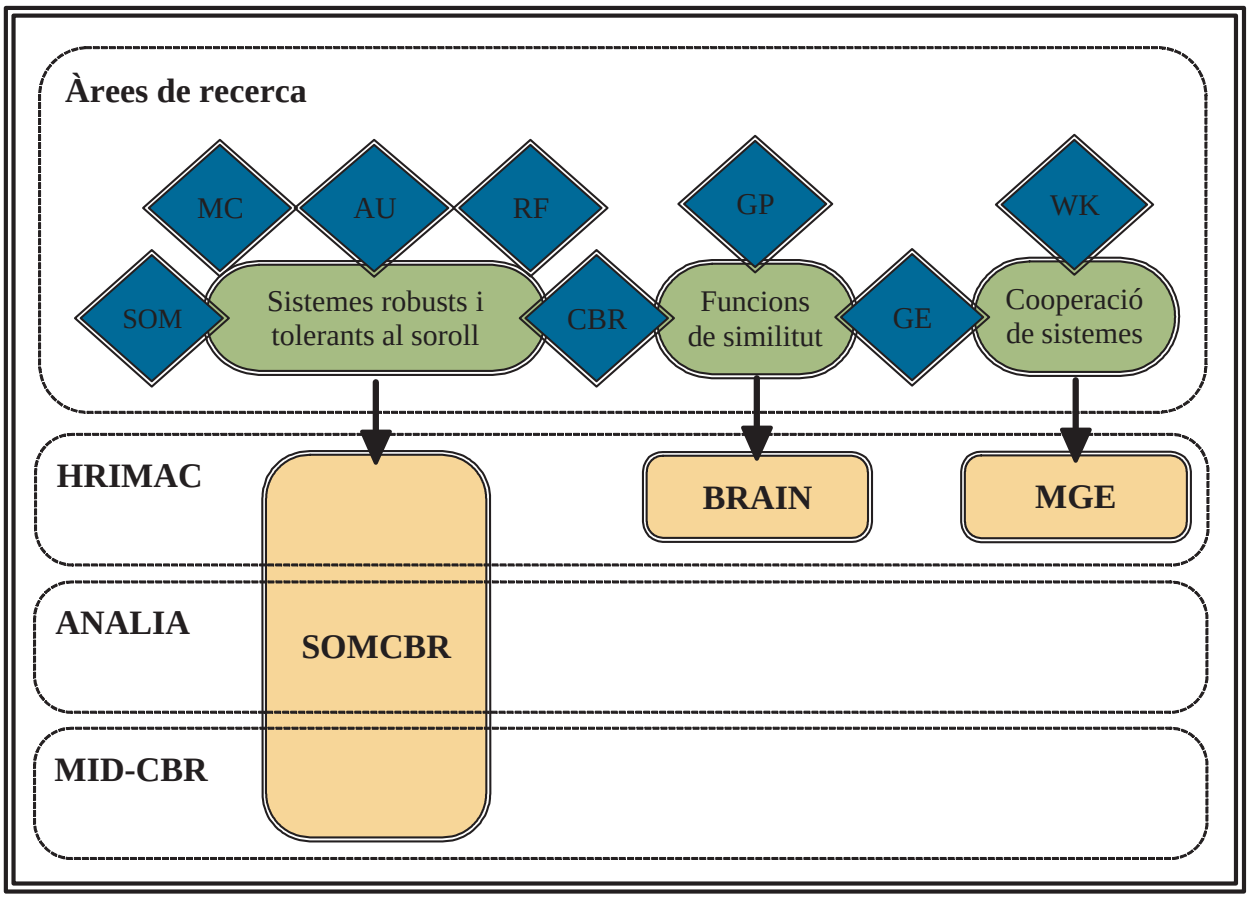

Figura 13.2: Vista d'ocell de les relacions entre les línies de recerca desenvolupades (verd), els paradigmes estudiats (blau) i les plataformes desenvolupades (taronja) pels 3 projectes on s'ha emmarcat la tesi.

Fase d'emmagatzematge.

- A. Fornells i E. Golobardes. *Case-base maintenance in an associative memory organized by a Self-Organizing Map.* Al llibre *Innovations in Hybrid Intelligent Systems*, volum 44. Editors: E. Corchado, J.M. Corchado, i A. Abraham. Springer-Verlag, 2007. En impremta.

Altres relacionats amb SOM.

- G. Corral, A. Fornells, E. Golobardes i J. Abella. *Cohesion factors: improving the clustering capabilities of CONSENSUS.* Al *7th International Conference on Intelligent Data Engineering and Automated Learning*, volum 4224 de LNCS, páginas 488-495. Springer-Verlag, 2006.

Disseny de funcions de similitut.

- A. Fornells, J. Camps, E. Golobardes i J.M. Garrell. *Comparison of strategies based on evolutionary computation for the design of similarity functions.* Al llibre *Artificial Intelligence Research and Development*, volum 131, planes 231-238. IOS Press, 2005.
- A. Fornells, J. Camps, E. Golobardes i J.M. Garrell. *Incorporación de conocimiento en forma de restricciones sobre algoritmos evolutivos para la búsqueda de funciones de similitud.* Al *IV Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*, planes 397-404. Thomson, 2005.

Disseny d'esquemes de cooperació.

- A. Fornells, E. Golobardes, E. Bernadó i J. Martí. *Decision support system for breast cancer diagnosis by a meta-learning approach based on grammar evolution.* Al *8th International Conference on Enterprise Information Systems*, volum 233, planes 222-239. INSTICC Press, 2006.

Com s'ha vist al llarg de la tesi, si hi ha una cosa que no hi falta és 'pluralitat de tècniques i paradigmes' utilitzats per la construcció del SOMCBR. Aquesta afirmació queda reflectida clarament en l'ampli equip de co-autors amb els quals s'ha fet la recerca. A més a més, un punt molt important a tenir en compte és la pluralitat i diversitat dels àmbits de la seva procedència. Concretament, el llistat de persones i àmbit de procedència ordenat alfabèticament és el següent:

- Àmbit de la computació evolutiva: Dr. Josep Maria Garrell i Joan Camps.
- Àmbit de les matemàtiques i l'estadística: Josep Maria Martorell.
- Àmbit de les mètriques de complexitat: Dra. Ester Bernadó i Núria Macià.
- Àmbit del raonament basat en casos: Dra. Elisabet Golobardes, Dra. Eva Armengol i David Vernet.
- Àmbit de la telemàtica: Guiomar Corral, Jaume Abella i Agustín Zaballos.
- Àmbit de les xarxes neuronals: Dr. Xavier Vilasís
- Àmbit de la visió per computador: Dr. Joan Martí.

Malgrat que treballar amb persones d'àmbits diferents implica sovint un esforç addicional per tal d'apropar postures tant dels punts de vista personals com de les metodologies per treball, les eines de treballar, els sistemes d'avaluació o la manera com expressar les coses, aquest no ha estat el nostre cas. Les ganes de treballar, col·laborar i fer sempre coses innovadores que ens beneficiessin a tots i, que a més a més fossin útils pels projectes, ens ha permès treballar com un gran equip multidisciplinar on cada component ha assolit els objectius personals i professionals que desitjava. Per tant, només tinc paraules d'agraïment a totes aquestes persones gràcies a les quals he pogut aprendre molt de tots aquests àmbits.

Finalment per acabar aquest punt, la figura 13.3 resumeix a alt nivell un GANT amb la cronologia de les tasques realitzades al llarg de la tesi. Per tal de facilitar la seva visualització d'una manera fàcil, aquest s'ha reagrupat en quatre grans parts corresponents a:

- Cursos de doctorat.
- Documentació i anàlisi de les tècniques de la intel·ligència artificial estudiats.
- Els treballs de recerca.
- Les dates en les quals s'han presentat els treballs citats anteriorment.

13.6 Conclusions i línies futures

La tesi tenia un objectiu principal: la definició d'un marc per integrar les capacitats de SOM en totes les fases del CBR per tal de permetre la construcció d'un sistema més robust i tolerat a grans volums de dades complexes i incertes. Aquest objectiu s'ha assolit satisfactòriament tal com s'ha demostrat als capítols del 5 al 8, que han conclòs amb la publicació de deu articles a revistes, congressos i llibres.

D'altra banda, a la tesi s'han tractat dos temes addicionals fruït de les necessitats del projecte HRIMAC: el disseny de funcions de similitud i el disseny d'esquemes de cooperació. Tot i que els treballs en ambdues línies estan encara força 'verdes', aquests han posat les bases per continuar treballant-hi des de dues tesis al GRSI. Tanmateix, els treballs realitzats van permetre abordar satisfactòriament les necessitats que es demanaven.

Per tant, tenint en compte tot el que s'ha explicat al llarg dels capítols de la tesi, els punts més importants a destacar són:

- SOMCBR és un sistema que permet oferir un rendiment superior al CBR gràcies a les capacitats de SOM per tractar dades complexes i incertes, així com les capacitats per identificar relacions ocultes a les dades.
- La capacitat de la GE per introduir restriccions en el procés de cerca de funcions de similitud i d'esquemes de cooperació permet trobar en un temps finit, alternatives que ofereixen un rendiment superior als mètodes tradicionals.

Finalment, abans d'exposar les línies futures de la tesi a l'últim apartat de la memòria m'agradaria fer una petita reflexió a partir de la tornada d'una cançó composada per en Pau Donés del grup *Jarabe de Palo*:

En lo puro no hay futuro
la pureza está en la mezcla
en la mezcla de lo puro
que antes que puro fue mezcla.

Què és un sistema CBR pur, o potser millor dit, un esquelet d'un sistema CBR bàsic? Un conjunt de quatre fases que interaccionen entre elles i sobre una memòria dinàmica per tal d'abordar un problema. No obstant, aquest sistema està combinant elements que provenen d'altres vessants 'pures'. Això fa que aquest sistema 'purista' abans de ser-ho hagi sigut el resultat d'una combinació d'altres sistemes. Al mateix temps, aquests sistemes purs han d'evolucionar cap a d'altres per no quedar-se endarrere i desaparèixer en un 'futur'. Per exemple, aquesta mateixa concepció d'evolució apareix dins del plantejament del projecte MID-CBR per tal de definir nous enfocaments basats en els d'ús intensiu de coneixement i ús restringit de coneixement.

Aquesta mateixa reflexió s'aplica al sistema SOMCBR 'pur' proposat, el qual és una combinació de diferents paradigmes 'purs' que abans de ser-ho han sigut també una combinació. Al mateix temps, el sistema SOMCBR requereix de noves millores per abordar amb èxit les noves necessitats que apareguin dels usuaris, per tal d'evolucionar cap a un nou 'futur'. Aquestes millores són les que estan englobades a les línies de futur explicades de cada capítol.

Per tant, aquesta tesi és una demostració clara de que els enfocaments híbrids són els que tenen més possibilitats de tenir èxit gràcies a la seva diversitat, la qual els hi permet abordar

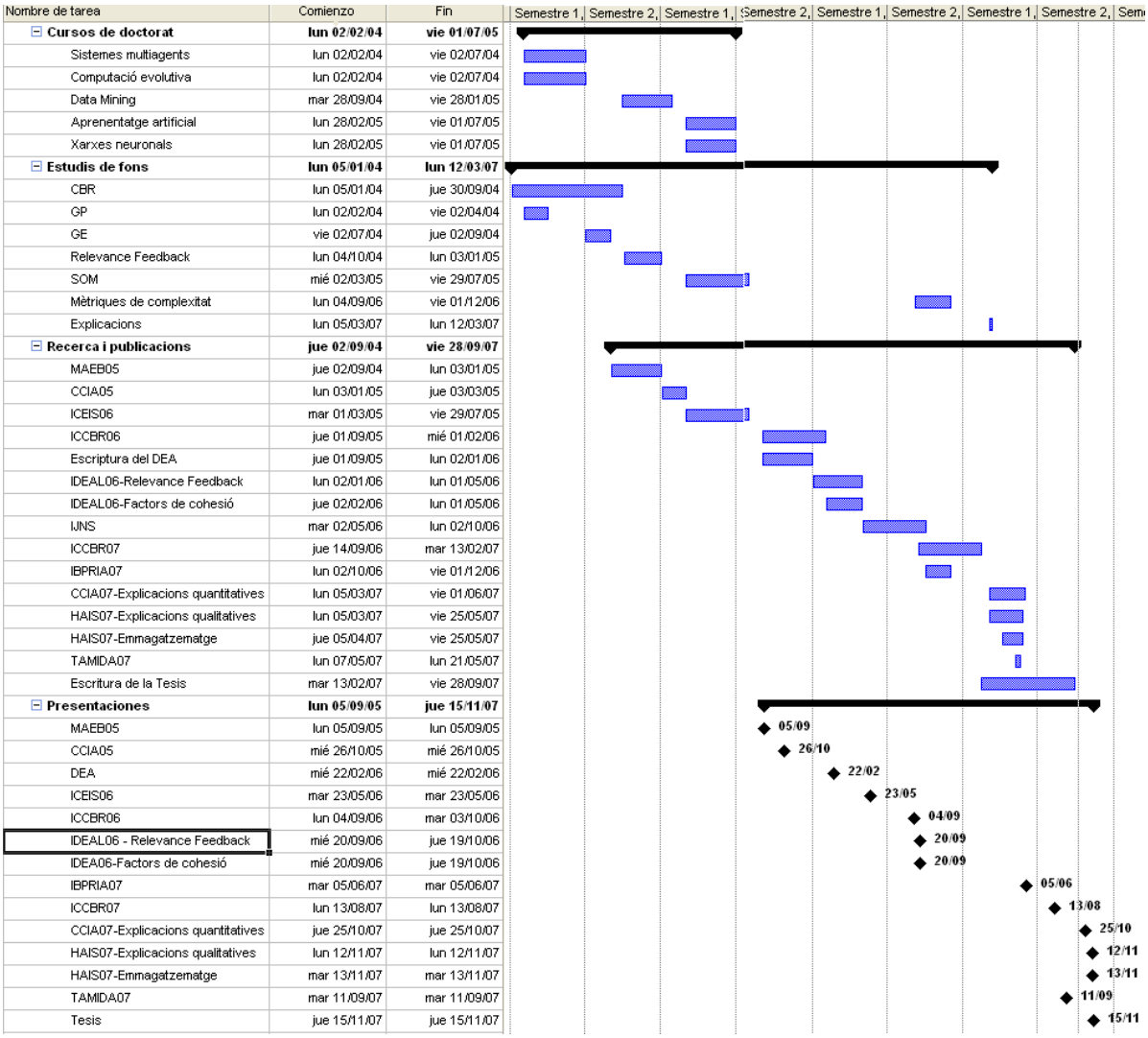

Figura 13.3: Resum de la planificació des dels inicis del doctorat, fins la seva presentació. La data de lectura de la tesi és aproximada, ja que encara no es coneix.

problemes mitjançant les virtuts d'altres sistemes. Precisament, la pluralitat de les persones i la seva capacitat per cooperar i complementar les mancances són els elements que ens han permès realitzar un desenvolupament tan ràpid del nostre 'domini' en el món. Ara només falta veure el nostre 'futur'.

13.7 Línies futures

Les línies de treball futur són molt àmplies fruit de l'ampli ventall de tècniques que es veuen afectades en la definició del SOMCBR. Millorar les tècniques, implica millorar el SOMCBR. En línies generals les més destacades són:

- Millorar l'arquitectura de SOM amb les noves millores que apareguin en el seu àmbit.
- Estudiar la definició de noves mètriques de complexitat, o bé, afinar-les per tal de modelar millor la relació entre la geometria de les dades i la representativitat dels clústers.
- Definir i analitzar nous punt de vista per seleccionar i recuperar els clústers i els casos com per exemple el punt de vista plantejat al capítol de les explicacions.
- Definir noves estratègies per definir probabilitats a partir dels clústers i la distribució de les dades.
- Potenciar la capacitat de les explicacions per discernir el contingut dels clústers, així com la classe dels casos.
- Permetre el canvi dinàmic de l'arquitectura de SOM per adaptar-se millor al manteniment del coneixement.

Pel que fa a la vessant de la CE, les línies de recerca estarien orientades a millorar el GE així com a definir estratègies per afinar el nivell de restricció de les explicacions.

Finalment, a partir de totes aquestes millores s'ha de continuar millorant l'eina de detecció de vulnerabilitats dins del projecte ANALIA. Al mateix temps, s'ha d'iniciar la definició d'una eina per la classificació del càncer de melanoma juntament amb els grups de recerca del IIIA i la UCM, amb la col·laboració de la Fundació clínica per a la recerca biomètida de l'hospital clínic de Barcelona dins del marc del projecte MID-CBR.

Resum

Aquest darrer capítol ha posat punt i final a la memòria de la present tesi. Al llarg del capítol s'han repassat les diferents fites assolides per tal de la consecució de l'objectiu principal de la tesi: definir un marc integrador de les capacitats de *Soft-Computing* i de *Knowledge Discovery* de SOM per potenciar les fases del CBR davant de grans volums de dades complexes i incertes. D'altra banda, també s'han repassat els dos objectius secundaris que van aparèixer fruït dels requeriments del projecte HRIMAC. D'una banda la definició de funcions de similitud específiques per un domini. D'altra banda, la definició d'esquemes de cooperació específics a un domini. Ambdues línies s'han afrontat des del punt de vista d'una variant de la CE anomenada GE. Finalment, el capítol ha conclòs amb un resum de les 13 publicacions realitzades, en les quals han participat 13 investigadors de diferents àmbits i universitats.

Part V

Apèndix

Apèndix A

Sigles

ACR - *American College of Radiology* - Col·legi americà de radiòlegs
BRAIN - *hyBRid system to discover And Improve similarity fuNctions* - Sistema híbrid per descobrir i millorar funcions de similitud
BIL - *Breast Imaging Lexicon* - Lèxic d'imatges mamogràfiques
BNF - *Backus Naur Form* - Forma Backus Naur
CD - *Critical Distance* - Distància crítica
CAD - *Computer Aided Diagnosis* - Diagnòstic assistit per computador
CBM - *Case Base Maintance* - Manteniment de la memòria de casos
CBR - *Case Base Reasoning* - Raonament Basat en Casos
CBIR - *Content Base Image Retrieval* - Recuperació d'imatges basades en el contingut
CSIC - *Consejo Superior de Investigaciones Científicas* - Consell superior d'investigacions científiques
DURSI - Departament d'Universitats, Recerca i Societat de la Informació
DVDM - *Discretized Value Difference Metric* - Mètrica de la diferència de valors discrets
EALS - Enginyeria i Arquitectura La Salle
EC - *Evolutionary Computation* - Computació Evolutiva
EE - *Evolutive Estrategies* - Estratègies Evolutives
EP - *Evolutive Programming* - Programació Evolutiva
GA - *Genetic Algorithm* - Algorismes Genètics
GE - *Grammar Evolution* - Evolució de Gramàtiques
GP - *Genetic Programming* - Programació Genètica
GRSI - *Group of Research in Inteligent Systems* - Grup de Recerca en Sistemes Intel·ligents
HEOM - *Heterogeneous Euclidean-Overlap Metric* - Mètrica de solapament euclidià heterogeni
HJT - *Hospital Universitari Doctor Josep Trueta de Girona*
HRIMAC - *Herramienta de Recuperación de Imágenes Mamográficas por Análisis de Contenido para el asesoramiento en el diagnóstico de cáncer de mama* - Eina de recuperació d'imatges mamogràfiques per l'anàlisi de contingut per l'assessorament en el diagnòstic de càncer de mama
HVDM - *Heterogeneous Value Difference Metric* - Mètrica de la diferència de valors heterogenis
IA - *Artificial Intelligence* - Intel·ligència Artificial
IIIA - *Research Institute of Artificial Intelligence* - Institut d'Investigació d'Intel·ligència Artificial
IRS - *Information Restrieval Systems* - Sistemes de recuperació d'informació
IVDM - *Interpolated Value Difference Metric* - Mètrica de la diferència de valors interpolats
JACK - *JAva Case base reasoning Kernel* - Nucli de raonament basat en casos de Java
KEEL - *Knowledge Extraction based on Evolutionary Learning* - Entorn per l'Extracció de Coneixement basat en Algoritmes d'Aprenentatge Genètic i Evolutiu
KBS - *Knowledge Base Systems* - Sistemes Basats en el Coneixement
mCP - *minimum complexity point* - Punt de mínima complexitat

MCP - *maximum complexity point* - Punt de màxima complexitat
MGE - *Meta-learning based on Grammar Evolution* - Meta aprenentatge basat en l'evolució de gramàtiques
ML - *Machine Learning* - Aprenentatge Artificial
NN - *Neural Networks* - Xarxes Neuronals
NNA - *Nearest Neighbor Algorithm* - Algorisme del veí més pròxim
PCA - *Principal Component Analysis* - Anàlisi de Components Principals
PE - *Program Evolution* - Programes d'Evolució
RL - *Reinforcement Learning* - Aprenentatge per reforç
ROC Curve - *Receiver Operator Characteristic Curve* - Corba característica de l'operador receptor
SC - *Sample Correlation* - Correlació Mostral
SOM - *Self-Organizing Map* - Mapa autoorganitzatiu
SOMCBR - *Self Organitzation Mapping integration in Case Base Reasoning* - Mapes autoorganitzatius integrats en el Raonament Basat en Casos
UdG - *Girona University* - Universitat de Girona
UCM - *Universidad Complutense de Madrid* - Universitat Complutense de Madrid
URL - *Ramon Llull University* - Universitat Ramon Llull
VDM - *Value Difference Metric* - Mètrica de la diferència de valors
WVDM - *Widowed Value Difference Metric* - Mètrica de la diferència de valors acotats
XML - *eXtensible Markup Language* - Llenguatge de marques estès

Apèndix B

Dades del projecte HRIMAC

HRIMAC (*Herramienta de Recuperación de Imágenes Mamográficas por Análisis del Contenido para el Asesoramiento del Cáncer de Mama* - TIC 2002-04160-C/02-02) és un projecte d'investigació finançat pel *Ministerio de Ciencia y Tecnologia i els fons FEDER* que va ser coordinat i realitzat pel grup de Visió per Computador de la Universitat de Girona i pel Grup de Recerca en Sistemes Intel·ligents de la URL, amb la participació de l'Hospital Universitari Dr. Josep Trueta de Girona.

La finalitat era desenvolupar una eina CAD per donar suport als metges a l'hora de realitzar diagnòstics de mamografies. L'eina es basa en permetre l'accés a una determinada tipologia d'imatges mamogràfiques digitals emmagatzemades en diverses bases de dades públiques, a partir del contingut d'una imatge exemple seguint determinats criteris d'afinitat. Poden diferenciar-se dues parts a l'eina:

Preprocessament de la informació. Mitjançant tècniques de visió per computador la mamografia es segmenta, reforça i digitalitza per obtenir la seva caracterització. Aquesta part ha estat liderada pel Dr. Joan Martí del grup de Visió per Computador. Les figures B.1 i B.2 mostren una mamografia abans i després de ser preprocessada.

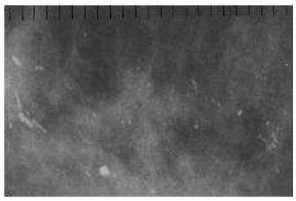

Figura B.1: Mamografia original.

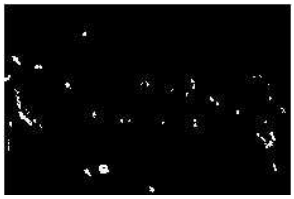

Figura B.2: Mamografia preprocessada.

Algorisme de recuperació d'imatges. L'aplicació d'algorismes d'aprenentatge artificial per determinar les associacions dels pacients que tenen símptomes similars. Aquesta part ha estat liderada per la Dra. Elisabet Golobardes del Grup de Recerca en Sistemes Intel·ligents.

Els metges distingeixen dos tipus de dades a tenir en compte a l'hora de realitzar un diagnòstic:

- Dades personals:
  - Edat, que marca el tipus d'evolució del càncer.
  - Sexe, si és una dona és important conèixer si ha tingut fills i si els hi ha donat el pit, així com l'edat de la primera menstruació.

  - Pes o índex de massa muscular.
  - Antecedents d'altres familiars amb càncer.

- Informacions de la mamografia:
  - Distorsions i anormalitats en la forma.
  - Massa muscular i densitat del teixit.
  - Informació sobre els clústers de les microcalcificacions (μCa).
  - Nombre i característiques de la μCa (vegeu la taula B.1).

Aquestes dades tenen principalment dos problemes: (1) algunes dades estan lligades a la subjectivitat de la persona que les obté, com per exemple saber en quina mesura la mare va donar el pit als seus fills, i; (2) la informació de les imatges mamogràfiques pot contenir soroll degut a que:

- No totes les mamografies estan fetes amb mamògrafs digitals. Moltes són fotografies de la radiografia amb diferents zooms, perspectives i qualitat.
- No hi ha una visió absoluta de la mamografia, sinó diferents tipus de vista (obliqua, craniocaudal, mig lateral-obliqua).
- Les mamografies són obtingudes amb diferents tècniques de segmentació d'imatges.
- La subjectivitat del metge influeix a l'hora de descriure les anormalitats de la mamografia.

A més a més, un pacient pot haver realitzat N visites, on a cadascuna pot haver-se fet M mamografies des de diferents vistes, i a cada mamografia poden aparèixer un nombre diferent de μCa. Per tant, decidir quines dades es fan servir i com es representen és vital per poder afrontar el problema amb un mínim de garanties. El problema pot enfocar-se des de diferents punts de vista tenint en compte:

Atribut	**Descripció**
Àrea	El nombre de píxels la μCa
Perímetre	La longitud dels costats de la μCa
Compacte	Derivat del perímetre P i l'àrea A d'una μCa és: $P^2/(4*\pi*A)$
Box Min. X,Y i Box Max. X,Y	Les coordenades dels extrems esquerra, dreta, sota i sobre dels píxels de la μCa
Feret X,Y	Les dimensions de la finestra més petita en la direcció horitzontal i vertical respectivament.
Diàmetre Mín. Feret	El diàmetre de feret més petit després de mirar els diferents angles (un màxim de 64)
Diàmetre Màx. Feret	El diàmetre de feret més gran després de mirar els diferents angles (un màxim de 64)
Diàmetre Mitja Feret	El diàmetre mitja de tots els angles mesurats.
Elongació Feret	Mesura la forma de la μCa, es calcula com : (feret max. diàmetre) / (feret min. diàmetre)
Nombre de forats	El nombre de forats de la μCa
Perímetre Convex	Una aproximació al perímetre convex de la μCa
Rugositat	Mesura la rugositat, es calcula com Perímetre/PerímetreConvex
Longitud	Mesura la longitud real de la μCa
Amplada	Mesura l'amplada real de la μCa
Centroide X,Y	La posició (x, y) del centre de gravetat de la μCa
Eixos Principals	L'angle en el que la μCa té el menor moment d'inèrcia (l'eix de simetria). Per μCa llargues, s'assigna l'eix més llarg
Eixos Secundaris	L'angle perpendicular a l'eix principal
Classificació	Benigne: 0, i Maligne: 1

Taula B.1: Característiques d'una microcalcificació.

- L'historial del pacient.
- Les mamografies de manera global.
- La descripció de les μCa de les mamografies.
- L'historial del pacient i les μCa de les mamografies.

En el nostre cas, donat l'enfocament que es vol donar i les dades de les quals es disposa, es treballa només amb la informació de les mamografies. Aquesta simplificació i relaxació ens permet abordar el problema, i en un futur si s'escau, introduir la resta de dades per tal de complementar el sistema.

La taula B.2 descriu els jocs de dades amb els que s'ha treballat en aquest projecte. Les diferències entre les dades resideixen en l'origen de les dades, els descriptors de la mamografia, i la classificació utilitzada. El *dataset* μCa (Martí et al., 2000) conté mostres provinents de pacients de l'Hospital Universitari Doctor Josep Trueta de Girona. Aquestes mostres estan descrites només pels atributs explicats a la taula B.1 i la seva classificació associada s'ha fet mitjançant una biòpsia que indica si la mostra és maligne o benigne.

DDSM (Heath et al., 2000) i MIAS (Suckling et al., 1994) contenen mostres provinents de bases de dades públiques d'imatges mamogràfiques, les quals han estat estudiades i processades en (Oliver et al., 2005b) i (Oliver et al., 2005a) respectivament. En ambdós casos les dades es classifiquen segons el grau de densitat del teixit mamari seguint l'estàndard BIRADS [1]. Originàriament DDSM i MIAS classifiquen en 4 i 3 classes respectivament. A més a més, els experts de l'Hospital Universitari Doctor Josep Trueta de Girona van reclassificar les mostres MIAS-3c en 4 classes a MIAS-Bi.

Codi	***Dataset***	**Atributs (Tipus)**	**Distribució de les classes**	**# Instàncies**
CA	μCa	22 (Numèrics)	benigne (121), maligne (95)	216
DD	DDSM	143 (Numèrics)	b1(61), b2(185), b3(157), b4(98)	498
MB	MIAS-Bi	153 (Numèrics)	b1(128), b2(78), b3(70), b4(44)	322
M3	MIAS-3c	153 (Numèrics)	gras(106), dens(112), glandular(104)	322

Taula B.2: Descripció dels jocs de dades proporcionats pel departament de Visió per Computador de la Universitat de Girona.

[1] BIRADS (*Breast Imaging Reporting and Data System*) (Kopans, 1998; Cardeñosa, 2001; Gamagami, 1996; Samuels, 1998) és un estàndard de classificació que proporciona els principis generals per la detecció i diagnòstic de càncer de mama definit pel col·legi americà de radiòlegs.

Apèndix C

Dades del projecte ANALIA

El projecte ANALIA és un projecte parcialment subvencionat pel Ministerio de Ciencia y Tecnología CIT-390000-2005-27, i té com a finalitat la introducció de tècniques d'intel·ligència artificial i de mineria de dades dins d'una eina telemàtica anomenada CONSENSUS. Aquesta eina va ser desenvolupada dins d'un projecte parcialment subvencionat per un PROFIT FIT-360000-2004-81, i on va col·laborar l'empresa ISECOM. Actualment, ISECOM és també una EPO del projecte MID-CBR-GRSI (TIN 2006-15140-C03-03).

CONSENSUS va néixer amb l'objectiu d'oferir als professionals de la seguretat una eina per la detecció automàtica de vulnerabilitats que poden existir a la xarxa, així com als dispositius que la composen. Per fer-ho, utilitza un sistema de testeig de seguretat distribuït, automàtic, modular i independent seguint la metodologia OSSTMM (*Open Source Security Testing Methodology Manual*). Aquest sistema permet l'avaluació dels sistemes de les xarxes corporatives de forma eficient i automàtica, l'emmagatzemament dels resultats obtinguts, i la presentació d'aquests resultats de forma que un professional de la seguretat pugui conèixer l'estat de la seva xarxa.

No obstant, CONSENSUS no inclou prestacions que ajudin a l'expert de seguretat a analitzar el gran i complex volum de dades recopilades per a cada dispositiu de la xarxa. És dins d'aquest marc, on l'aplicació de tècniques provinents de la intel·ligència artificial pot ajudar a recuperar la informació de les màquines més vulnerables respecte a la totalitat dels dispositius, de manera que afavoreixi el treball de l'analista de seguretat per trobar més ràpidament els dispositius a actualitzar i millorar.

Per tant, ANALIA aplica tècniques de la intel·ligència artificial i la mineria de dades als resultats emmagatzemats a CONSENSUS per millorar la fase d'anàlisi posterior al test, i ajudar a l'analista de seguretat en l'extracció de conclusions mitjançant un processament previ de la informació obtinguda. L'arquitectura global del sistema ANALIA es mostra a la figura C.1.

Una de les grans dificultats per estudiar la detecció automàtica de vulnerabilitats és que no existeix cap estàndard relacionat amb aquesta temàtica, el qual detalli tota la informació que cal per realitzar una mesura de la seguretat. No obstant, els experts de seguretat estan d'acord que el fet de recopilar *logs*, capturar mostres del tràfic de xarxa, i detectar potencials amenaces permet avaluar amb un cert grau la seguretat d'una xarxa (Dawkins i Hale, 2004). Degut a aquesta incertesa, el projecte ANALIA treballa amb diferents dades i representacions i, d'aquesta manera, amb l'aplicació de tècniques de clústering es pretén determinar quines són més precises per detectar les vulnerabilitats.

Els patrons extrets dels dispositius de la xarxa es representen en forma de vector multidimensional, on cada dimensió fa referència a una característica individual. A més a més, al ser un domini no supervisat on no es coneix la classe de vulnerabilitat dels dispositius, no existeix un atribut classe associat a la representació del dispositiu.

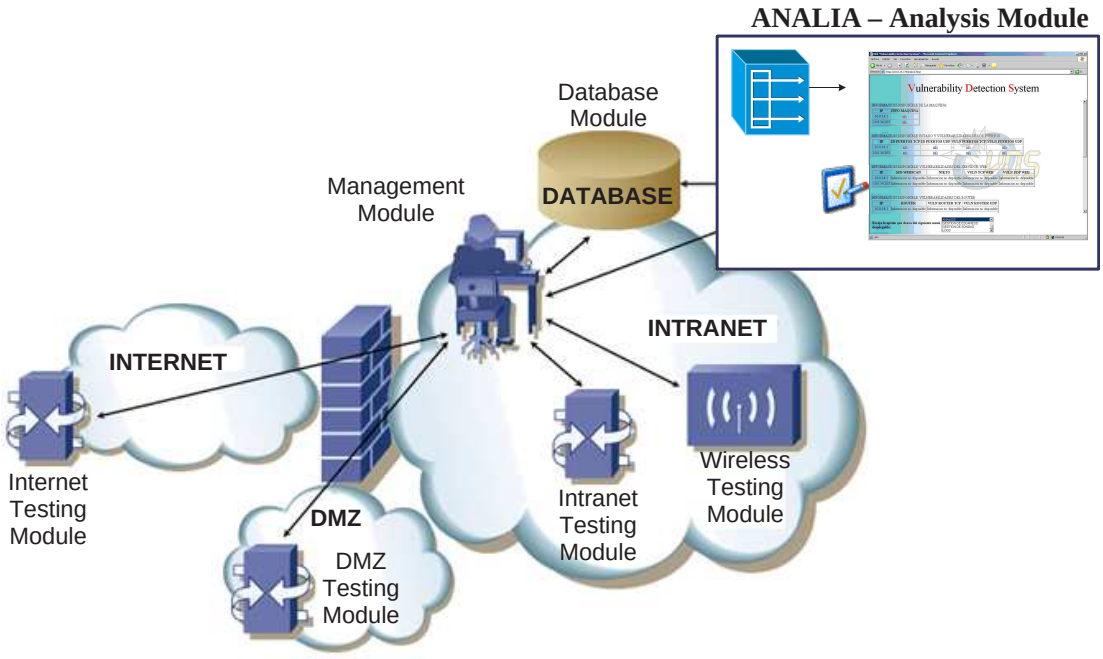

Figura C.1: Arquitectura d'ANALIA.

A través de l'interfície d'ANALIA es poden definir moltes representacions del coneixement amb les dades recopilades. No obstant, és important tenir present que la majoria de les tècniques de clustering requereix de dades quantitatives, i una part de les dades retornades per les eines que recopilen informació són paràgrafs de texts difícils de separar i unificar. Per aquest motiu, les dades amb les quals s'ha treballat han estat basades principalment en:

- Informació de l'estat dels ports oberts (enters).
- Probabilitat de que tingui un determinat operatiu (reals).
- Informació relacionada amb forats de seguretat detectats, o avisos de risc (enters).

Tot i que hi ha informacions que no es tenen en compte, aquestes dades són les més rellevants per determinar les vulnerabilitats en els serveis dels equips informàtics (Corral et al., 2005b). La taula C.1 mostra les tres representacions proposades que millors resultats han aportat.

La diferència entre les representacions està a la representació de la informació associada als ports, la qual pot ser des de molt extensa (a) fins molt compacte (c):

- La representació (a) té en compte un atribut per port estudiat. Un cas està representat per 165 atributs.
- La representació (b) conté el sumatori dels ports oberts. Un cas està representat per 61 atributs
- La representació (c) estudia el sumatori dels ports oberts per rangs de ports. Un cas està representat per 57 atributs.

Taula C.1: Representació del coneixement de la xarxa per detectar vulnerabilitats. La representació (a) conté els ports oberts, la (b) el sumatori dels ports oberts, i la (c) el sumatori dels ports oberts per rang. A més a més, cada representació conté la probabilitat de que un operatiu estigui instal·lat, així com informacions respecte forats de seguretat i avisos de riscs.

(a)

Ports					**Operating Systems**					**Vulnerability types**		
23	25	53	80	...	Linux	Solaris	XP SP1	XP SP2	...	Holes	Warnings	Notes
1	1	0	1	...	0.67	0.2	0.0	0.0	...	6	3	11

(b)

Ports	**Operating Systems**					**Vulnerability types**		
$\sum openports$	Linux	Solaris	XP SP1	XP SP2	...	Holes	Warnings	Notes
35	0.67	0.2	0.0	0.0	...	6	3	11

(c)

Port ranges				**Operating Systems**					**Vulnerability types**		
1-100	101-500	501-900	...	Linux	Solaris	XP SP1	XP SP2	...	Holes	Warnings	Notes
25	4	0	...	0.67	0.2	0.0	0.0	...	6	3	11

Apèndix D

Metodologia d'anàlisi de resultats

D.1 Paràmetres d'avaluació

Els sistemes d'aprenentatge estan orientats a resoldre el major nombre de problemes possibles. En base a les seves característiques, cal fer front a uns requeriments més o menys crítics segons la problemàtica i domini d'aplicació. A l'hora d'avaluar un sistema és important tenir en compte els paràmetres següents:

- Rendiment: pot mesurar-se des de tres punts de vista:
  1. Eficiència. El cost computacional necessari per solventar el problema.
  2. Competència. Defineix el rang de problemes objectiu que pot solucionar, és a dir, la capacitat que té per adaptar-se i modelar el comportament del problema.
  3. Qualitat. Es caracteritza per la capacitat de predicció. A més de proposar solucions que tinguin un error mínim, han de ser fiables. La qualitat també esta condicionada pel cost associat a cometre determinats errors.
- Credibilitat: mesura en quin grau els resultats que el sistema proporciona són correctes. Els sistemes han de ser capaços de generalitzar a partir de l'entrenament per permetre resoldre nous problemes i no limitar-se només als casos del conjunt d'entrenament.

A partir d'aquests criteris, com pot estimar-se el rendiment del sistema que s'ha proposat? En quin grau proporciona resultats més fiables i creïbles que altres propostes? Amb l'objectiu de respondre a aquestes qüestions, en els següents apartats s'exposaran mètodes amb els quals es podrà realitzar una valoració objectiva del sistema que s'està avaluant.

D.2 Errors dels sistemes d'aprenentatge

Els sistemes d'aprenentatge a partir d'una mostra representativa de la població han de ser capaços de modelar el comportament d'un problema, per tal de resoldre'l amb l'error mínim possible. No obstant, aquesta és una tasca complexa que molts cops es veu encara més dificultada degut a les dades amb les quals s'entrena el sistema. Poden definir-se diferents tipus d'errors:

Error inherent de les dades. Es produeix quan amb les característiques que s'extrauen del problema no es poden separar les classes. També es coneix com l'error mínim de Bayes.

Error propi de l'algorisme d'inferència. És l'error produït durant el procés d'obtenció de les dades.

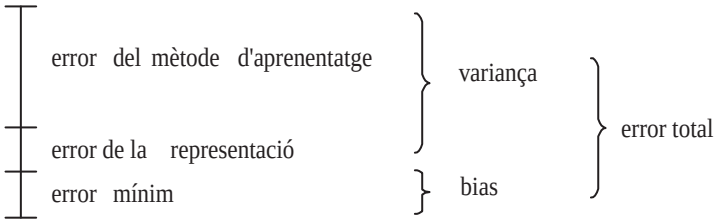

Figura D.1: Descomposició de l'error total al resoldre un problema.

Error intrínsec del llenguatge que representa el coneixement. S'introdueix com a conseqüència de representar mitjançant un tipus de dades la informació del problema.

Error aparent. És l'error que s'obté al fer servir només un conjunt reduït de totes les dades del problema a l'hora d'entrenar.

Error real. També conegut com biaix, és l'error mínim que es cometria si es disposessin de totes les dades.

Les dades normalment vénen proporcionades per uns tercers, de tal manera que poc pot fer-se sobre els dos primers tipus d'errors. L'error associat a la representació del coneixement pot reduir-se si es defineix correctament la representació de la informació segons el mètode d'aprenentatge que es faci servir. A més, sempre és necessari un preprocessament previ de les dades per preparar-les com es comentarà més endavant.

L'estimació de l'error total pot descomposar-se en el biaix i la variança (vegeu la figura D.1). El biaix és l'error mínim que comet el sistema degut a que possiblement no té les dades d'entrenament necessàries per modelar correctament el sistema. En canvi, la variança és l'error resultat d'avaluar el sistema amb les dades de test. A més, aquests errors estan influenciats per l'error introduït de la representació de les dades.

El fet d'augmentar les dades d'entrenament permet fer que el sistema modeli millor el seu comportament i, per tant, que es redueixi el biaix. Tanmateix, fer servir moltes mostres en l'entrenament pot fer aparèixer situacions de sobreaprenentatge. Això vol dir que el sistema aprèn a resoldre bé els problemes que estan relacionats amb l'entrenament i és incapaç d'extrapolar el problema. La conseqüència d'això és que la variança del sistema augmenta ja que hi ha més error en la fase de test.

En canvi, si es fan servir menys dades d'entrenament l'error del biaix serà major ja que el sistema potser no té tota la informació necessària per modelar perfectament el problema, però sí que serà capaç d'extrapolar més respostes, fet que implica una reducció de la variança.

Aquest problema es coneix com el dilema biaix-variança, ja que al disposar de conjunts de dades finits el fet de reduir un error comporta l'augment de l'altre. Existeixen estudis en els quals s'intenta estimar quin és el nombre de mostres necessari per aconseguir estimar l'error real amb l'error aparent com per exemple les tècniques estadístiques de l'anàlisi PAC (Shawe-Taylor i Williamson, 1997). L'estimador capaç d'aproximar a 0 el biaix i la variança s'anomena estimador ideal.

Per tant, l'estimació de l'error real amb l'error aparent ens dóna un error més baix que si realment es fessin servir més mostres. En aquests casos es diu que es fa servir un estimador esbiaixat optimista.

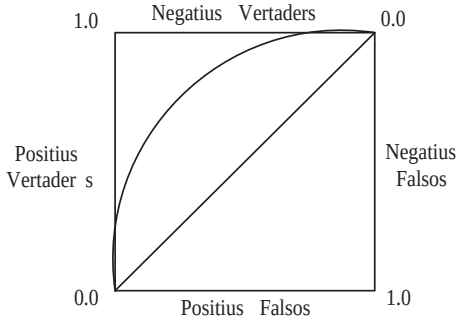

Figura D.2: Corba ROC (*Receiver Operator Characteristic*).

D.3 Avaluació de les estadístiques

Les estadístiques que generen els sistemes d'aprenentatge estan formades per moltes dades, de les quals les més interessants són els encerts, dades de fiabilitat i les matrius de confusió. A continuació, es realitza una breu descripció d'aquestes mesures, així com les seves implicacions.

D.3.1 Tant per cent d'errors, no classificats i encerts

Els tants per cent (%) dels resultats mesuren el grau d'encert (*succes rate*), error (*error rate*) o no classificats (*unknown rate*) respecte el nombre total d'exemples que s'han avaluat en una simulació (vegeu l'equació D.1). La no classificació d'un exemple pot ser deguda a diverses raons:

- La solució proposada no supera un llindar mínim que marca la validesa de la solució.
- No es tenen suficients dades per realitzar la solució.

$$\begin{aligned} \%encerts &= \frac{\#encerts}{\#avaluacions}(\times 100) \\ \%error &= \frac{\#errors}{\#avaluacions}(\times 100) \\ \%noclassificats &= \frac{\#noclassificats}{\#avaluacions}(\times 100) \end{aligned} \tag{D.1}$$

D.3.2 Corbes ROC: sensitivitat i especificitat

Les Corbes ROC (*Receiver Operator Characteristic Curve*) són una altra manera d'avaluar un sistema, concretament és tècnica estadística que permet avaluar en quin grau un sistema que ha d'assignar dues possibles classificacions (casos positius i negatius) és fiable, és a dir, calcula la capacitat de classificar correctament els exemples tant positius com negatius, tenint en compte els cops que els classifica incorrectament. Aquests conceptes s'engloben en la Sensitivitat (*Sensitivity*) i l'Especificitat(*Specificity*) que es calculen a partir dels següents elements:

- Sigui N el nombre d'exemples. Els exemples pertanyen a una de les dues classes disjuntes del problema (p.ex. benigne/maligne, cert/fals).
- TP (*True positive*): vertaders positius són el nombre d'exemples positius que han estat ben classificats.

- TN (*True negative*): vertaders negatius són el nombre d'exemples negatius que han estat ben classificats.
- FP(*False positive*): falsos positius són el nombre d'exemples negatius que han estat classificats com positius.
- FN(*False negative*): falsos negatius són el nombre d'exemples positius que han estat classificats com a negatius.

La Sensitivitat mesura la capacitat de classificar correctament els exemples positius respecte tos els exemples positius. L'Especificitat mesura la capacitat de classificar els exemples negatius respecte tots els negatius. A partir d'això es defineixen les seves equacions (vegeu l'equació D.2).

$$Sensitivitat = \frac{TP}{TP+FN}(\times 100) \quad Especificitat = \frac{TN}{TN+FP}(\times 100) \tag{D.2}$$

Aquests operadors es representen mitjançant la corba ROC, la qual mostra gràficament com l'única manera de reduir els falsos positius és incrementant els falsos negatius (la línia de punts representa variants aleatòries), tot mantenint el mateix poder d'observació.

D.3.3 Matrius de confusió

La finalitat d'aquesta mesura és avaluar la capacitat que té el sistema en discernir entre les diferents classes a l'hora de classificar una instància. Per tal de fer-ho es crea una matriu de NxN inicialitzada a zeros, on N representa el número de classes, i cada casella de la matriu representa la túpla <percentatge de la classe real, percentatge de la classe predita>.

Durant el procés de classificació de les mostres, per cada instància que es resolt s'incrementa la casella corresponent a la matriu. La capacitat de separar correctament les classes del problema es mesura en funció de la distribució que s'obté a la matriu. Serà millor si aconsegueix realitzar la major part dels increments en la diagonal o prop d'aquesta, és a dir, si comet errors ho fa perquè les instàncies poden pertànyer a classes que són molt similars i degut a que no es disposen de mostres suficients per entrenar el sistema, no pot modelar-se correctament la seva separació. En canvi, si els increments es produeixen lluny de la diagonal el sistema classifica malament.

D.4 Mecanismes per estimar l'error

En els sistemes d'aprenentatge es distingeixen dues fases de simulacions: entrenament (*train*) i avaluació (*test*). En la fase d'entrenament el sistema aprèn a generalitzar/especificar segons el tipus de concepte, i en la segona fase s'avalua la capacitat per avaluar nous problemes. En alguns casos es pot trobar una fase intermitja en la qual s'optimitzen els resultats de la fase d'aprenentatge.

La definició de conjunts de *train* i *test* permet determinar el grau d'influència de les dades en aquestes dues fases. El conjunt de *train* està format pels exemples que es fan servir per entrenar, i el conjunt de test per avaluar el resultat després de l'entrenament realitzat prèviament. És recomanable garantir la mateixa proporció d'instàncies per classe en els conjunts d'entrenament i test. Aquest procés s'anomena estratificació i permet millorar la precisió de l'estimador, especialment la variança.

A continuació es realitza una breu exposició de diferents maneres d'avaluar els conjunts de *train* i test amb la finalitat de proporciona resultats que siguin el més fiables possibles i que no falsegin la realitat (Witten i Frank, 2000) (Golobardes i Bernadó, 2005).

Algorisme D.1: Algorisme *Holdout*.

```
Funció Holdout() és
    Sigui E un exemple
    Sigui M el nombre d'exemples
    Sigui R els exemples que es fan servir per entrenar el sistema
    Sigui T = M - R, els exemples que es fan servir per avaluar el sistema
    Entrena el sistema amb R
    Per cada E de T fer
        Avalua E sobre el sistema
        Guarda estadística
    Estadístiques = Mitja de les estadístiques
    retorna Estadístiques
```

Algorisme D.2: Algorisme *Random Subsampling*.

```
Funció Random_SubSampling() és
    Sigui E un exemple Sigui M el nombre d'exemples
    Sigui I el nombre d'iteracions
    Sigui R el conjunt d'entrenament
    Sigui S el conjunt de test
    Per i=0 fins a N fer
        Inicialitza R amb un % aleatori dels exemples de M
        Inicialitza T = M - R
        Entrena el sistema amb R
        Avalua el sistema amb les instàncies de T
        Guarda estadístiques
    Estadístiques=Mitja de les N execucions
    retorna Estadístiques
```

D.4.1 *Holdout*

A partir d'un conjunt de 'M' mostres, es crea un únic conjunt d'entrenament per entrenar i un altre de test per avaluar el sistema (vegeu l'algorisme D.1). Aquest mètode requereix un nombre de mostres desitjables aproximadament de 1000, essent recomanable que 2/3 siguin per entrenar (R), i 1/3 per avaluar (T). El mètode dóna bons resultats per jocs de prova petits. El biaix obtingut és pessimista i la variança és elevada. Els efectes poden millorar-se al aplicar tècniques de *resampling*.

D.4.2 *Random subsampling*

Aplica un *Holdout* diferents cops sobre diferents conjunts d'entrenament i test (vegeu l'algorisme D.2). Tot i que millora el biaix i la variança, els resultats no són correctes si no s'aconsegueix que els conjunts d'entrenament i test siguin independents.

D.4.3 *N-Fold Validation*

L'objectiu és analitzar com afecta a l'avaluació del sistema el nombre de mostres dels conjunts d'entrenament i test. És molt important que els exemples de tots els conjunts de les dades siguin representatius per no falsejar els resultats.

L'error sobre el conjunt d'entrenament s'anomena *resubstitution error* i només permet conèixer l'error de l'aprenentatge. L'error interessant és el que s'obté sobre el conjunt de test, el qual mesura la capacitat per resoldre nous problemes.

Algorisme D.3: Algorisme *N-Fold Validation*.

Funció *N_Fold_Validation()* **és**
- Sigui M el nombre d'exemples
- Sigui N el nombre de conjunts a crear
- Es reparteixen els M exemples en N grups de manera equitativa
- Sigui $NTrain$ els conjunts d'entrenament, inicialment 1 conjunt
- Sigui $NTests$ els conjunts de test, inicialment N-1 conjunts
- **Per** *i=0* ***fins a*** N **fer**
  - Entrena el sistema amb les instàncies dels conjunts de $NTrain$
  - Avalua el sistema amb les instàncies dels conjunts de $NTest$
  - Guarda les estadístiques de la proporció
  - Elimina un conjunt de dades de $NTest$, i s'afegeix a $NTrain$
- **retorna** *Estadístiques de les proporciones avaluades*

Algorisme D.4: Algorisme *N-Cross Validation*.

Funció *N_Cross_Validation()* **és**
- Sigui M el nombre d'exemples
- Sigui N el nombre de conjunts a crear
- Es reparteixen els M exemples en N conjunts de manera equitativa i estratificada
- Sigui $NTrains$ els conjunts d'entrenament
- Sigui $NTest$ el conjunt de test
- **Per** *i=0* ***fins a*** N **fer**
  - $NTest$ = Mostres del conjunt i
  - $NTrain$ = M - $NTest$
  - Entrena el sistema amb les instàncies dels conjunts de $NTrain$
  - Avalua el sistema amb les instàncies dels conjunts de $NTest$
  - Guarda les estadístiques de la proporció
- Estadístiques=Mitja de les N execucions
- **retorna** *Estadístiques*

L'algorisme consisteix en definir N conjunts inicials a partir dels quals s'avalua els resultats fent servir de menys a més conjunts en la fase d'entrenament (vegeu l'algorisme D.3).

D.4.4 *N-Cross Validation*

En el mètode anterior segons l'ordre en el qual els conjunts són escollits els resultats poden variar perquè poden haver-hi conjunts de dades més representatius i significatius que d'altres, degut a que les dades de les que es disposa sovint són limitades. L'objectiu del *Cross Validation* és minimitzar l'efecte de l'ordre de les agrupacions. Per fer-ho executa el sistema N cops, on cada cop fa servir N-1 conjunts diferents per entrenar i un conjunt per fer el test. El funcionament de la tècnica es reflecteix en l'algorisme D.4.

D.4.5 *K-Iterative N-Cross Validation*

L'objectiu és reduir la variança de l'algorisme de *N-Cross Validation* anteriorment comentat, així com el cost computacional necessari per realitzar els càlculs. La idea consisteix en repetir el N-Cross Validation K cops reordenant els elements que formen part dels N/K conjunts. Per tant, un *10-Cross Validation* és equivalent a fer un *2 Iterative 5-Cross Validation*.

D.4.6 *Leave One Out*

És una variant del *N-Cross Validation* que s'aplica quan no es poden estratificar conjunts de dades perquè es disposen de poques. Consisteix en aplicar un *N-Cross Validation*, on N és el nombre

d'instàncies. El conjunt de test està format només per un dels N exemples i amb la resta es fa entrenament. L'estimació del resultat s'obté amb la mitja del nombre d'encerts (1) i d'errors (0).

És un mètode que requereix un cost computacional molt elevat, i que té un biaix molt petit perquè gairebé fa servir totes les dades per entrenar. Per tant, aquest mètode es fa servir quan disposem de poques dades, o volem estimar la influència en els resultats de la classe més representativa.

D.4.7 *Bootstrap*

El mètode anterior té un baix biaix i una alta variança perquè fa servir un alt nombre d'instàncies per entrenar i poques per avaluar. Amb l'objectiu de reduir la variança, el mètode *Bootstrap* permet que el conjunt d'entrenament tingui elements repetits, i el de test està format per totes les mostres que no estiguin al conjunt d'entrenament.

D.4.8 Pronòstic del rendiment

Les estadístiques que s'obtenen en quina mesura són certes? Per exemple, si tenim un tant per cent d'encerts del 75% i un tant per cent d'errors del 25%, en quina mesura es compleix? +10%? -5%? Això depèn en gran part del volum que tingui el conjunt d'entrenament del sistema.

Per solventar aquest dilema es fa servir el model estadístic de Bernoulli (Witten i Frank, 2000) que es basa en establir un interval de confiança. Per exemple, si el tant per cent d'encerts és del 75% (750 encerts sobre 1000 intents) i s'estableix un interval de confiança de 80% la proporció que s'espera està compresa entre el 71% i el 80%.

D.5 Significància dels resultats

Els mètodes d'avaluació ens permeten realitzar una estimació sobre la capacitat de predicció del sistema, però com de bo és respecte un altre? Per analitzar-ho es fan servir els tests d'hipòtesis estadístiques.

Suposem que es disposa de dos sistemes A i B, que tenen els errors aparents error_A=10%, i error_B=15%. És millor A que B? Per respodre això els test estadístics plantegen la hipotesi nul·la de l'equació D.3. Aquesta hipotesi estableix que els dos mètodes tenen el mateix error, si som capaços de rebutjar aquesta hipotesi es demostrarà que els métodes són diferents.

$$H_0 = e_A - e_B = 0 \tag{D.3}$$

Poden produir-se dos tipus d'errors:

- Error tipus I: rebutjar la hipòtesi nul·la quan aquesta és certa. Probabilitat que, suposant que la hipòtesi nul·la sigui certa, d'obtenir la diferència observada o una diferència superior (α).
- Error de tipus II: no rebutjar H0 quan aquesta es falsa (β).

No obstant, abans de veure si pot rebutjar-se o no la hipòtesi cal establir el nivell de confiança a partir del qual s'accepta. Aquest es defineix com la confiança a rebutjar H_0, i es calcula com 1-α (vegeu la figura D.3).

A continuació es descriu el funcionament del t-Student per avaluar si un sistema proporciona millors resultats que un altre. No obstant, existeixen molts altres tipus de test com per exemple: Wilcoxon, Bonferroni, Dunnett, Student-Newman-Keuls, Tukey, Holm, etc.

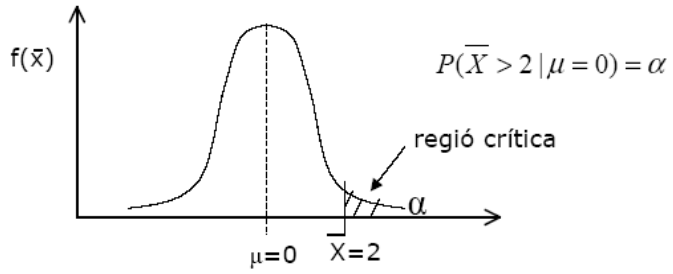

Figura D.3: Representació del nivell de confiança en una distribució.

D.5.1 Test *t-Student* aparellat

El test *t-Student* aparellat és un dels més habituals quan es treballa amb els resultats provinents de treballar amb el *N-Cross Validation* a partir de les mateixes dades. Si hi ha 2 mètodes, A i B, que ens proporcionen N resultats parcials com a resultat d'aplicar el métode de *N-Cross Validation*, quin és millor?

Si suposem que les dades tenen una distribució normal i les diferències són variables aleatòries independents (els conjunts del *N-Cross Validation* són independents), pot calcular-se el valor t (vegeu l'equació D.4) que permet rebutjar la hipòtesi H_0.

$$t = \frac{\overline{\delta}\sqrt{K}}{\sqrt{\frac{\sum_{i=1}^{k}(\delta_i - \overline{\delta})^2}{K-1}}} \tag{D.4}$$

On:

- K representa el número de conjunts.
- δ_i és la diferència de l'error parcial i entre A i B.
- δ és la mitja dels errors parcials.

El nivell de confiança s'estableix com 1-α, i els graus de llibertat (v) s'estableixen com K-1. A partir del nivell de confiança (1-α) i els graus de llibertat(v) es consulta una taula que retorna un número ($t_{1-\alpha,v}$). Si $(t_{1-\alpha,v}) > t$, aleshores es rebutja la hipòtesi nul·la i es considera que A i B són diferents. Si el valor és positiu aleshores A és millor que B, sinó al revés.

D.5.2 Test *t-Student* no aparellat

Aquest test s'aplica quan els resultats dels mètodes (A i B) que es volen comparar no provenen de les mateixes dades. Tot l'anteriorment dit abans es manté amb la diferència que ara la t es calcula segons l'equació D.5.

$$t = \frac{\overline{\delta}}{\sqrt{\frac{\sigma_A^2}{K} + \frac{\sigma_B^2}{L}}} \tag{D.5}$$

On:

- A representa els resultats parcials '$(A_0, A_1, ..., A_{K-1})$'.
- B representa els resultats parcials '$(B_0, B_1, ..., B_{L-1})$'.
- K i L representen el número de conjunts d'A i B respectivament.
- σ_A^2 i σ_B^2 és la variança de les mostres de A i B respectivament.
- $\overline{\delta}$ és la diferència de les mitges dels valors de A i B.

Apèndix E

Preprocessament de les dades

E.1 Les dades

Les dades són conceptes que descriuen una determinada situació a partir d'un conjunt de propietats representades mitjançant atributs, els quals poden prendre tipus, rangs i importàncies de valors diferents. El conjunt de totes les possibles situacions existents constituiran el domini del problema. No obstant, les dades no han de perquè ser independents, ni complertes, ni aïllades del soroll. Aquests inconvenients fan necessari que hi hagi un procés previ de tractament de les dades per aconseguir minimitzar l'impacte sobre l'eina que ha de processar les dades, per exemple, un sistema de classificació.

Per tant, la manera de gestionar i tractar aquesta informació condiciona la capacitat per comprendre el domini, i conseqüentment, determinar l'èxit de solventar el problema. Al llarg d'aquest capítol s'aniran plantejant els problemes que poden aparèixer amb les dades, així com les tècniques que poden aplicar-se per minimitzar el seu impacte.

E.1.1 Tipus de dades

Cada una de les instàncies de les dades d'entrada del problema està formada per un conjunt de característiques anomenades atributs. Els estadistes consideren que el tipus de naturalesa dels atributs pot classificar-se en:

- **Nominals**: el valor és una paraula (Exemple: solejat: sí/no).
- **Ordinaris**: existeix una relació entre les categories (valors possibles de l'atribut) (Exemple calor > normal > fred).
- **Interval**: els atributs pertanyen a un interval i es pot operar matemàticament.
- **Ratio**: és un valor que es refereix a partir d'una base.

No obstant, de manera més general es consideren dues grans famílies:

- **Nominal**: són dades representades discretes que poden classificar-se en:
  - **Categòric**: el valor és un paraula.
  - **Enumerat**: existeixen un conjunt de valors representats per paraules, i s'estableix un ordre entre aquestes.
  - **Discret**: treballa amb un conjunt finit de valors numèrics.
  - **Dicotòmic**: té dos valors possibles (Exemple: cert/fals).
- **Ordinals**, **numèrics** o **continus**: números sobre els que es realitzen operacions matemàtiques.

```
% Indica comentari
@RELATION <nom_problema>

@ATTRIBUTE <nom_atribut> TIPUS
...

@DATA
2.2, 1.4, A
...
```

Figura E.1: Descripció del format ARFF.

E.1.2 Repositoris

Un repositori de dades (*Data Repository*) es defineix com un base de dades de problemes 'benchmark' que fan servir els experts per avaluar i comparar els seus sistemes respecte el d'altres. Els problemes s'emmagatzemen seguint algun tipus de format propi o quasi-estàndard, com per exemple l'ARFF (ARFF, 2007), on per cada problema es defineix:

- Nom del problema.
- Tipus i nom dels atributs que descriuen un exemple.
- Exemples de casos.

E.1.2.1 Format *Attribute-Relation File Format* (ARFF)

El format de representació ARFF (ARFF, 2007) amb el pas del temps i degut a la seva massiva utilització s'ha convertit en quasi un estàndard. Tal i com es detalla a la figura E.1, en el fitxer es diferencien tres parts:

- **RELATION**: indica el nom del problema.
- **ATTRIBUTE**: hi ha tantes línies com atributs hi hagi. Els tipus que poden definir-se són:
  - **NUMERIC**: representa els tipus enter i real (i.e.: 2.5, 3, 3.1E-1)
  - **STRING**: representa un valor que és un array de caràcters (i.e.:Nom d'una persona)
  - **NOMINAL**: representa un conjunt de valors enumerats(i.e.: cert, fals)
  - **DATE**: representa una data del calendari. La seva representació depèn del format de la data. El format per defecte és l'ISO-8601 (yyyy-Mdd'T'HH:mm:ss). També s'estableixen altres formats com dd/mm/yyyy. Els formats s'indiquen mitjançant '[ ]'.
- **DATA**: hi ha tantes línies com casos. Cada línia indica un exemple del problema representat per tots els seus atributs separats per comes.

E.1.2.2 Extensió d'ARFF

Tot i que el format ARFF és quasi un estàndard, els jocs de proves dels repositoris de dades tenen petites variacions que es centren principalment en la declaració del tipus d'atributs. Les modificacions més freqüents i que s'han de tenir presents són:

- El tipus NUMERIC pot definir-se com REAL o INTEGER.
- El tipus NUMERIC pot tenir opcionalment el rang que té l'atribut. S'indica després de declarar el tipus afegint al final de la línia '[valor mínim - valor màxim]'.
- El tipus BOOLEAN pot definir-se com un tipus simple enlloc del tipus NOMINAL.
- Els valors declarats com a NOMINALS es representen com cadenes de caràcter, encara que puguin interpretar-se com números (Per exemple: 0 o 1).
- Els valors desconeguts es representen amb el valor '?', o bé, deixant un espai buit.

E.1.3 Problemes amb les dades

Molts cops les dades de les quals es disposa a l'hora de resoldre un problema són pobres, tenen errors i alguns valors són desconeguts. Tots aquests factors afecten de manera decisiva al procés que s'alimenta d'aquestes dades. Per aquest motiu, es fa necessari estudiar mecanismes per minimitzar l'impacte d'aquests factors negatius en el sistema que processa les dades. Aquest conjunt de mecanismes s'anomenen processos de neteja de dades (*data cleaning*), i es centre principalment en:

- Adequació de la representació de les dades que descriuen el problema per ser usables per l'eina de preprocessament (atributs continus o discrets).
- Normalitzar les dades per disposar de rangs similars que no distorsionin els resultats.
- Detecció, i en el millor dels casos correcció, de l'efecte del soroll provocat per la representació o mecanisme de mesura.
- Gestió de valors desconeguts.
- Anàlisi i quantificació de la importància dels atributs.

A més, si les dades provenen de diferents fonts cal assegurar-se que totes s'han obtingut fent servir els mateixos criteris, i que han estat quantificades amb els mateixos atributs i unitats.

Per tant, el procés de neteja de les dades és un procés costós i laboriós necessari per garantir uns resultats fiables. Està estimat que el 60% de l'esforç en aplicar tècniques de data mining o processos sobre dades està destinat a la preparació de les dades (Cabena et al., 1997). A continuació, es farà un breu repàs sobre les tècniques més conegudes per abordar aquesta problemàtica.

E.2 Tècniques de preprocessament

Les dades proporcionades per l'expert a partir de les quals s'ha de resoldre un determinat problema no solen ser aplicables directament al sistema. El motiu d'això és que aquestes poden contenir soroll o tenir valors desconeguts entre d'altres problemes. Al llarg d'aquest apartat es fa un repàs dels principals aspectes que s'han de tenir en compte.

E.2.1 Soroll o inconsistència en els valors dels atributs

És fàcil trobar-se amb situacions de soroll o inconsistència en les dades quan es treballa amb grans volums de dades que requereixen processos externs per obtenir, tractar i digitalitzar la informació. El soroll es tracta de manera diferent si afecta als descriptors de l'exemple (que pot ser puntual o homogeni), o si afecta a la classificació de l'exemple. Pel cas d'errors puntuals és recomanable corregir el valor per evitar tractar amb informació que no és del tot correcte, la dificultat és

saber com estimar el seu valor. En canvi, si el soroll està present en tots els casos d'una manera homogènia, és millor no corregir aquest efecte perquè el sistema podrà adaptar-se a l'error perquè afecta d'una manera 'normal' a tots els casos.

En situacions que el soroll es presenta en l'atribut que defineix la classe no hi ha lloc per l'error, i només pot corregir-se si es disposa d'un expert humà o un mecanisme que realitzi una validació fiable. Si no pot corregir-se, s'haurà d'ignorar l'exemple.

E.2.1.1 Detecció i correcció de valors amb soroll, erronis o desconeguts

La detecció d'errors sovint implica saber què és correcte sense saber a priori què pot ser erroni. Aquesta és una tasca complexa que requereix un profund estudi estadístic i probabilístic de les dades per poder 'intuir' on pot haver un error amb una certa probabilitat, però sense cap seguretat d'encertar al 100 %. Aquestes inconsistències poden estar provocades per un canvi de dades, una introducció incorrecta, soroll en la mesura, subjectivitat de la persona que la descriu, etc.

Els errors que són més fàcils de detectar, i que sovint afecten més perquè influeixen més en el càlcul de mitges, són els que tenen a veure amb valors molt grans o petits en els atributs. Una manera d'evitar això és eliminar els límits dels valors de les mostres, les quals poden estar lligades a valors alts o baixos produïts pel soroll, encara que no sempre aquests valors han de ser erronis. Les tècniques que es centren amb això són anomenades tècniques robustes (Nieto, 2001). Per tant, davant valors grans o petits d'un atribut pot fer-se servir el rang de l'atribut (si es coneix el domini), o estimar els límits i reajustar o ajustar els exemples que no pertanyin en un 10%.

La detecció i correcció de la classe d'un exemple és una tasca també molt complexa que requereix d'un expert o un mecanisme que permeti validar amb una certa fiabilitat que la classificació és correcta o no. Una estratègia habitual per validar classificacions és fer servir esquemes multiclassificadors o multicombinadors (Witten i Frank, 2000), els quals a partir de la combinació dels resultats proporcionats per un conjunt de tècniques, proposen quina és la classificació més 'fiable' dins d'un marge d'error. Aquest error dependrà del biaix-variança del mètode sobre el problema a tractar.

A més d'existir valors amb soroll o erronis, pot succeir que falti informació (*missing values*). L'origen d'aquesta omissió es classifica en 3 categories:

- **Desconeguda**. És impossible obtenir-la o massa costós.
- **No guardada**. És coneguda però no es va guardar.
- **Irrellevant**. No calia guardar-la perquè amb les altres ja es podia calcular el resultat.

Normalment la manera de processar les dades desconegudes i les no guardades es tracta de la mateixa manera. El cas de dades irrellevants, segons el que es defineixi per irrellevància, pot ser tan senzill com no fer servir l'atribut.

Un cop s'ha detectat l'error o omissió del valor de l'atribut, cal donar el següent pas: què fem? L'estratègia que es triï dependrà de com de crític sigui l'atribut i el nombre de mostres del problema. Les solucions poden passar des d'ignorar l'atribut, ignorar l'exemple, reemplaçar el seu valor pel més proper vàlid, o fins i tot, reemplaçar pel valor més habitual segons la classe a la qual pertanyi. En aquest últim cas, si l'atribut és numèric es pot fer la mitja dels atributs dels exemples que pertanyen a la seva mateixa classe, i si l'atribut és nominal es substitueix per l'element nominal que més cops apareix en aquesta classe. No obstant, aquesta solució té la limitació que només es podran corregir els exemples dels que es coneix la classe. A més, aquesta solució pot induir biaixos importants en el procés de construcció del model i degradar-ne la qualitat final.

E.2.2 Normalització dels atributs

Quan es calcula la similitud entre dues abstraccions o casos, és important tenir en compte el rang dels atributs que els formen ja que rangs molt dispars introduiran molt soroll en els resultats. Per evitar aquesta desviació en els resultats s'apliquen tècniques de normalització sobre les dades, les quals s'apliquen de manera diferent segons la tipologia de les dades.

E.2.2.1 Normalitzacions típiques dels atributs numèrics

La normalització consisteix a posar les dades sobre una escala de valors equivalent que permeti la comparació d'atributs que prenen valors en dominis o rangs diferents. Els criteris de normalització més comuns són:

- **Normalització pel màxim**. Dividir el valor de l'atribut per la diferència en valor absolut del seu domini (vegeu l'equació E.1). D'aquesta manera es garanteix que tots els resultats estan compresos entre 0 i 1.

$$x_{normalitzat} = \frac{x}{|x_{màxim} - x_{mínim}|} \tag{E.1}$$

- **Normalització per la desviació típica**. Dividir el valor de l'atribut per la desviació típica (vegeu l'equació E.2).

$$x_{normalitzat} = \frac{x}{\tau} \tag{E.2}$$

- Aplicar algun dels mètodes anteriors però eliminant el 5% dels valors propers als límits. D'aquesta manera pretenem reduir la influència dels valors propers als límits que poden tenir valors elevats per algun tipus d'error. Per exemple, que normalment els atributs tinguin un rang [0..10] però aparegui algun valor de 50 que introdueix molt soroll en el rang. A més, aquest valor rarament alt pot ser erroni.

- **Normalització per la diferència**. Aquesta tècnica de normalització pretén compensar l'efecte de la distància del valor que es tracta respecte al màxim dels valors observats (vegeu l'equació E.3).

$$x_{normalitzat} = \frac{x - x_{mínim}}{|x_{màxim} - x_{mínim}|} \tag{E.3}$$

- **Escalat decimal**. Aquesta normalització consisteix en reduir en un cert nombre de potències de deu el valor d'un atribut. Resulta especialment útil en tractar amb valors grans (per exemple, rendes o volums de negoci) (vegeu l'equació E.4).

$$x_{normalitzat} = \frac{x}{10^j} \tag{E.4}$$

  On:

  - 'j' és el valor més petit tal que $max\,|x_{normalitzat}| < 1$.

Caldrà triar uns mecanismes o altres en funció del tipus de dominis.

E.2.2.2 Normalització simultània d'atributs heterogenis

Quan un problema està definit amb atributs nominals i numèrics, pot ser interessant aplicar alguna normalització especial per tal d'aconseguir normalitzar els dominis entre ells, i treballar només amb un únic rang de valors. La normalització d'atributs basada en el mètode *Value Domain Metric* (VDM) permet definir un rang de valors que aglutina els atributs numèrics i nominals del problema.

Una bona manera de normalitzar els atributs numèrics es dividint per la desviació típica (τ). Si el 95% dels valors d'una distribució normal està en un interval de 2τ, la diferència entre dos atributs la dividirem per 4τ per poder escalar-ho en un rang de 0 a 1. A l'equació E.5 podem veure com es càlcula la diferència normalitzada per dos valors de l'atribut 'a'.

$$diferència\ normalitzada_a(x,y) = \frac{|x-y|}{4\tau} \tag{E.5}$$

Per normalitzar els valors nominals podem fer servir algunes de les formules següents:

$$normalització\ vdm1_a(x,y) = \sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right| \tag{E.6}$$

$$normalització\ vdm2_a(x,y) = \sqrt{\sum_{c=1}^{C} \left(\left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right| \right)^2} \tag{E.7}$$

$$normalització\ vdm3_a(x,y) = \sqrt{C \cdot \sum_{c=1}^{C} \left(\left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right| \right)^2} \tag{E.8}$$

On:

- '$N_{a,x}$' és el nombre d'instàncies del conjunt d'entrenament amb el valor 'x' en l'atribut 'a'.
- '$N_{a,x,c}$' és el nombre d'instàncies del conjunt d'entrenament que tenen el valor 'x' en l'atribut 'a' i la sortida de la classe 'c'.
- 'C' és el nombre de classes sortida en el domini del problema.

D'aquestes funcions es demostra en l'article (Wilson i Martinez, 1997) que l'equació E.7 serà més robusta que l'equació E.6 ja que té en compte les correlacions. Com es veu en els resultats de l'article, les mitges són més similars si es fa servir pels valors numèrics l'equació E.5 i pels nominals l'equació E.7, independentment del nombre de classes. En canvi, l'equació E.8 es veu més afectada en funció del nombre de casos.

Cal dir, que decidir quina és la funció més representativa sempre estarà lligat al domini, tot i que teòricament la que millors resultats hauria de proporcionar és l'equació E.7.

E.2.3 Discretització de les dades

El tractament de les dades no és el mateix si aquestes són contínues o discretes. Per aplicar algunes tècniques o mètodes cal poder tractar amb valors discrets encara que les dades no ho siguin.

En general, donat un atribut X amb un rang de valors $[x_{mínim}, x_{màxim}]$, la discretització consisteix a trobar una partició del rang de manera que cada subconjunt sigui de la màxima qualitat possible. La qualitat es pot mesurar en homogeneïtat de la partició o amb altres criteris. Els diferents aspectes on ens proporciona avantatges són:

- **Cost computacional**. La discretització comporta una reducció dels valors que cal tractar, el nombre de comparacions i càlculs que cal realitzar és més petit. Aquest aspecte és especialment important en mètodes com els arbres de decisió.

- **Velocitat en el procés d'aprenentatge**. La velocitat d'aprenentatge disminueix amb les dades discretitzades.
- **Emmagatzematge**. La discretització comporta menys memòria.
- **Mida del model resultant**. Els models amb dades discretitzades resulten més compactes.
- **Comprensió**. Els models són més compactes i més fàcils de copsar. La comprensió d'alguns models es millora descrivint els elements utilitzant menys termes.
- **Visualització de dades**. La representació és més fàcil i comprensible.
- **Distribució de les dades**. Si la distribució de les dades no és Gaussiana, treballar amb valors discrets millora els resultats que si treballem amb valors continus. Estudis d'aquest efecte sobre les Xarxes Bayesianes poden trobar-se en els treballs de Dougherty, Kohavi i Sahami (Dougherty et al., 1995).

El preu d'aquesta conversió és la pèrdua d'informació que sofreixen les dades, que es tradueix en la precisió o qualitat de la informació amb la qual es treballa després d'aplicar el procés.
Els mètodes de discretització es poden dividir segons criteris diferents en:

- **Supervisats o no supervisats**. Tenen en compte o no, respectivament, els valors de l'atribut classe.
- **Locals o globals**. Estan limitats a un atribut cada vegada (a un subconjunt de les dades originals) o tenen en compte tots els atributs i totes les dades.
- **Parametritzats i no parametritzats**. Els primers coneixen d'entrada el nombre màxim d'intervals que cal generar per a un atribut específic, mentre que els altres han de trobar aquest nombre automàticament.

La discretització pot realitzar-se en qualsevol moment, ja que en algunes aplicacions interessa que aquest procés es realitzi de manera incremental i segons un cert context, com per exemple en els arbres de decisió. Aquí la discretització es realitza en el moment de prendre la decisió de realitzar la divisió, de tal manera que aquesta depèn del context. D'aquesta manera, el procés genera diferents arbres segons la situació.

Un altre aspecte a tenir en compte és l'ordre a establir a l'hora de discretitzar els atributs, ja que si es realitza tenint en compte una visió global, pot afectar al resultat final.

E.2.3.1 Discretització no supervisada

Quan la transformació de les dades de valors continus a discrets, discretització, es realitza sense tenir en compte el coneixement de les classes de les instàncies, s'està treballant amb un enfocament no supervisat.

En molts problemes no es coneix les propietats de les classes, o com estan relacionades directament amb el domini dels seus atributs, i fins i tot, es desconeix les possibles classificacions.
Cal establir dos criteris:

- L'assignació de l'interval pel procediment de discretització a l'element.
- El nombre de grups òptim.

Altres tècniques:

Algorisme E.1: Algorisme *k-means*.

```
Funció creació_clústers(C: Conjunt entrenament) és
    Inicialitzar els k clúster amb les instàncies de manera aleatòria
    Mentre canvis dels elements en els clústers fer
        Per tot instància fer
            Reassignar-la al clúster més proper
            Recalcular els centres dels clústers afectats
    retorna clústers
```

- ***Equal-interval binning***. Parteix el rang en 'k' intervals de mida fixa.

$$mida\ cluster = \frac{|x_{màxim} - x_{mínim}|}{k} \quad \text{, on k és el nombre d'intervals} \tag{E.9}$$

  Distribueix els exemples d'una manera desigual, algunes agrupacions tenen molts exemples i d'altres en tenen pocs. Pot influir de manera negativa a l'analitzar la influència dels atributs en la decisió de la classificació de la instància.

- ***Equal-frecuency binning*** (o ***histogram equalization***). En funció del rang i el nombre d'instàncies (n) defineix un nombre d'intervals (k), on en cadascun hi ha el mateix nombre d'elements.

$$k = \frac{|x_{màxim} - x_{mínim}|}{n} \quad mida\ cluster = \frac{n}{k} \tag{E.10}$$

  Si tenim valors repetits, pot succeir que estiguin en intervals diferents. Pot corregir-se assignant un únic cop el valor en un interval, encara que això provoqui no tenir el mateix nombre d'elements en cada interval.

- ***K-means***. És un algorisme iteratiu que construeix clústers. Representa cada clúster per un centre que conté el valor mig per cada característica de les dades, i utilitza la distància mètrica per decidir l'assignació d'instàncies al clúster.

  Primer es creen 'k' conjunts (on 'k' és un valor d'usuari), s'examina per cada instància el centre on és més a prop. Això es va recalculant iterativament fins que no canvia l'assignació de les instàncies en els clústers (vegeu l'algorisme E.1).

Pot trobar-se un estudi d'aquests mètodes sobre 6 conjunts de dades en (Talavera i Gaudioso, 2001). Una de les conclusions d'aquests estudi és l'afirmació que és millor permetre assignar el nombre de clústers a l'algorisme de manera automàtica, ja que així no es limita l'espai de cerca.

El nombre d'intervals no ha de perquè ser el nombre de clústers, hi ha treballs sobre la cerca automàtica del número d'intervals (Dougherty et al., 1995).

E.2.3.2 Discretització supervisada

Quan la transformació de les dades de valors continus a discrets, quantització, es realitza tenint en compte el coneixement de les classes de les instàncies, es treballa amb un enfocament supervisat.

Al fer servir aquesta informació s'obté una millor discretització de les dades ja que la classificació és una característica molt diferenciadora. No obstant, aquesta característica per si sola en molts problemes no permet crear clústers clarament definits al no conèixer de la totalitat del domini. Algunes tècniques de discretització supervisades:

- **Discretització basada en l'entropia**. La idea és dividir el rang de valors del domini de l'atribut de manera recursiva fins que es troba un criteri d'aturada. Aquest criteri d'aturada

pot ser l'entropia, amb el qual es mesura la força de la informació quan es fan les agrupacions. Quan l'increment de la força d'agrupació no és significativa, no cal seguir agrupant.

- **Conversió de nominal a numèric**. Podem intentar buscar alguna codificació per cadascun dels possibles valors de l'atribut. A més, segons el tipus de domini fins i tot podem establir un ordre en funció del qual assignem valors. (Per exemple: fred=-1, normal=0, calor=1) Aquests valors són valors sintètics que es generen manualment per experts o mitjançant algun mecanisme que els assigni valors.

E.2.4 Rellevància dels atributs

Un exemple d'un problema està descrit per un conjunt d'atributs, on cadascun d'ells té un grau de rellevància diferent. Això vol dir que poden haver situacions descrites per molts atributs en les quals el 90% de la informació és descrita només per un parell d'atributs. Que un problema tingui més atributs que un altre no vol dir necessàriament que sigui més difícil de resoldre (sempre i quan es suposi que tenen una complexitat i tipologia similar), només indica que té més volum d'informació.

Està demostrat que el fet de tenir dades irrellevants pot degradar el rendiment del sistema entre 1-5% (Witten i Frank, 2000), ja que s'introdueix soroll. En altres casos, com per exemple el *Naive Bayes* (Mitchell, 1997), no són tan sensibles a aquest aspecte degut a què consideren tots els atributs independents, per contra, requereixen un cost computacional molt elevat per processar tots els atributs.

Per tant, realitzar una ponderació i selecció prèvia dels atributs rellevants evita introduir soroll, amb el conseqüent augment dels encerts, i a més estalvia càlculs innecessaris.
La reducció de la dimensionalitat pot enfocar-se des de dos punts de vista:

Extracció de característiques (*Feature extraction*). Es basa en crear un subconjunt de noves característiques (atributs) mitjançant la combinació dels existents. La gran dificultat és trobar la funció per transformar el conjunt de característiques actuals en un de nou que preservi el màxim la informació actual, essent el cas ideal aquell on l'error mínim es manté igual. Els criteris per cercar la funció de transformació s'agrupen en dues categories:

- **Representació del senyal (*Signal representation*).** L'objectiu és representar els exemples de mostra el més acuradament en un espai de dimensions més reduït.
- **Classificació (*Classification*).** L'objectiu és reforçar la informació discriminatòria que permet diferenciar les classificacions en un espai de dimensions més reduït.

Les tècniques de *Principal Component Analysis* (PCA) (Fornells, 2001) i *Linear Discriminant Analysis* (LDA) (McLachlan, 2004) són les més representatives de les dues categories anteriors respectivament. A la figura E.2 es pot veure com actuaria cadascuna de les tècniques si volguessin transformar un espai de dimensió N en un espai de dimensió 2, on hi ha dues possibles classificacions.

Selecció de característiques (*Feature selection*). Es basa en escollir un subconjunt de les característiques (atributs) format pels més rellevants. Aquest mètode permet solventar situacions que amb el mètode anterior no poden afrontar-se:

- El mètode funciona encara que hi hagin poques dades.
- Extraure regles significatives del classificador (p.ex. gestió valors desconeguts).
- Presència d'atributs no numèrics.

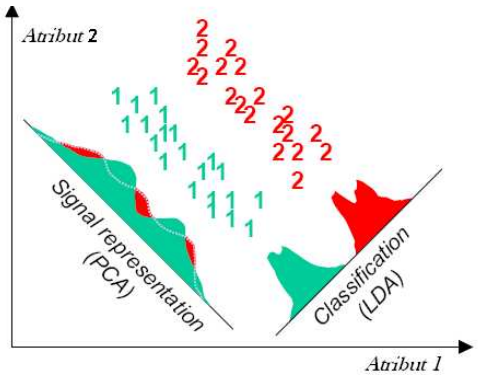

Figura E.2: Exemple de representació en un espai 2-D mostres segons PCA i LDA. PCA prioritza representar el màxim d'informació, LDA prioritza la separació de la classe (Gutierrez, 2004).

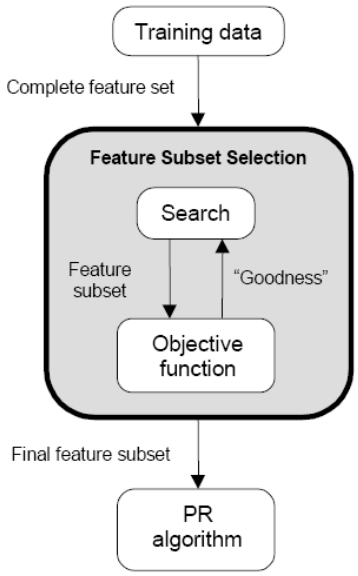

Figura E.3: Esquema general del procés de selecció de característiques (Gutierrez, 2004).

La figura E.3 representa els elements que intervenen en el procés de selecció d'atributs. Per una banda cal un procés de cerca 'intel·ligent' per proposar els subconjunts d'atributs, és a dir, no poden avaluar-se tots els conjunts de subconjunts de característiques possibles perquè seria inviable si el nombre d'atributs és gran (que es quan s'apliquen aquestes tècniques). Les estratègies de cerca poden classificar-se en tres grans grups:

- **Algorismes exponencials**. Aquests algorismes avaluen un nombre de subconjunts que creix exponencialment amb la dimensionalitat de l'espai de cerca. Els més representatius són *Exhaustive Search* (Nievergelt, 2000), *Branch and Bound* (Smith, 1984), *Approximate Monotonicity with Branch Bound* (Fischer et al., 2002) i *Beam Search* (E. Morales, 2006).
- **Algorismes seqüencials**. Aquests algorismes afegeixen i eliminen atributs de manera seqüencial, amb una tendència de quedar atrapats en mínims locals. Els algorismes més representatius són *Sequential Forward Selection* (Domingos, 1997), *Sequential Backward Selection* (Domingos, 1997), *Bidireccional Search* i *Sequential Floating Selection* (Jain i Zongker, 1997).

- **Algorismes aleatoris**. Són algorismes que incorporen accions aleatòries per evitar que el procés de cerca caigui en mínims locals. Els algorismes més representatius són *Random Generation plus Sequential Selection* (Gnedin, 2000), *Simulated Annealing* (Rajasekaran, 2000) i *Genetic Algorithms* (Goldberg, 1989).

D'altra banda, cal una funció d'avaluació que determini la 'bondat' del subconjunt candidat i que dirigeixi l'estratègia de cerca. Les funcions objectives es divideixen en dues categories principals:

- ***Filters*** (Herlocker et al., 1999). S'anomena així perquè es 'filtren' els atributs abans de realitzar cap procés d'aprenentatge. La funció objectiva avalua el subconjunt de característiques mitjançant la informació del propi subconjunt, típicament mitjançant la distància entre classes, dependències estadístiques, o mesures teòriques de la informació basades en la correlació. Els avantatges d'aquesta tècnica són que (1) és molt ràpida d'executar, i que (2) a l'avaluar les dades en si mateix proporciona una capacitat de generalització elevada al no lligar-se a un mètode concret d'avaluació.

  El principal desavantatge és que tendeixen a crear subconjunts d'atributs molt grans, traslladant a l'usuari la responsabilitat de decidir on 'tallar'.

- ***Wrapper*** (Kohavi i John, 1997). S'anomena així perquè l'algorisme d'aprenentatge està involucrat en la selecció dels atributs. La funció objectiu és un classificador de patrons, el qual avalua els subconjunts de característiques mitjançant la predicció dels encerts. Per exemple, es pot fer servir l'algorisme de construcció d'arbres de decisió per poder saber quins són els atributs que més energia tenen i d'aquesta manera escollir aquests com els més rellevants.

  Els avantatges d'aquesta tècnica són que (1) millora estadístiques de reconeixement perquè ajusta la interacció entre el mètode i el dataset, i (2) eviten problemes de sobreaprenentatge mitjançant polítiques d'avaluació basades de Cross-Validation per calcular els encerts.

  Els desavantatges són principalment que (1) té un alt cost computacional, i que (2) al ajustar tant el classificador al *dataset* d'entrenament es pot perdre capacitat generalitzadora.

A més de reduir el nombre d'atributs, és interessant poder establir en quin grau els atributs són rellevants per d'aquesta manera ajustar millor el sistema que ha de fer servir les dades, i aconseguir millorar la seva eficiència. Aquests 'pesos' dels atributs poden calcular-se a partir de qualsevol de les tècniques explicades anteriorment. Un altra manera d'estudiar les relacions entre les dades és mitjançant l'ús de programes per visualitzar-les, i d'aquesta manera detectar agrupacions sense que sigui necessari l'aplicació de cap tècnica estadística basada en l'estudi de les correlacions. No obstant, aquesta metodologia és difícil d'aplicar quan el nombre d'atributs és molt elevat.

En qualsevol dels casos, un bon indicador per mesurar la bondat dels atributs que formaran part de la nova representació és l'algorisme és analitzar si es mantenen les agrupacions inicials dels atributs reals.

E.2.5 Relacions d'alt nivell entre les dades

La millor manera de poder estudiar la relació entre les dades d'un problema és a partir de coneixement del domini, el qual permet d'una manera molt senzilla decidir què és important i què no ho és. Es distingeixen tres tipus de relacions d'alt nivell entre els atributs:

- **Semàntiques**. Si un atribut s'afegeix en una regla, l'altre atribut també. Per exemple: un atribut que mesura la quantitat de llet que tenim, estarà associat a un identificador de la vaca.
- **Causals**. Quan un atribut causa un altre. Per exemple: tenim una relació entre vaca i granger.
- **Funcionals**. És informació que s'intenta identificar per aconseguir normalitzar les relaciones entre les dades. Quan s'analitzen les dades, el significat de les dependències funcionals es veure si en una regla tenim un atribut, si és necessari que aparegui un altre.

Per poder estudiar aquests tipus de relacions és molt important tenir en compte el coneixement del domini, el qual s'obté a partir d'un expert en la majoria dels casos o sinó a partir de la teoria i l'experiència. A partir d'aquest coneixement del domini poden aplicar-se deduccions lògiques per tal de simplificar i eliminar redundàncies entre les regles. No obstant, el gran inconvenient és que en els problemes reals l'expert no té tota la informació, ja que si la tingués no caldria cap sistema d'aprenentatge.

E.3 Estratègies per gestionar grans volums de dades

Existeixen dues restriccions molt crítiques a l'hora de treballar amb grans volums de dades: l'espai de cerca en que es busca, i el temps d'exploració necessari per trobar la solució. Aquests dos factors estan directament lligats amb la representació i organització de la informació, ja que influeixen en la distribució i accés a la informació.

Per tant, és important seguir metodologies de treball per garantir que el volum de dades amb les que es treballa és adequat. Les principals tendències de com gestionar la càrrega de les dades es divideixen en:

- **Reduir el volum d'informació a tractar**. Es perd informació en el cas que hi hagin dades molt complexes o amb molts casos.
- **Paral·lelitzar el treball**. El problema es divideix en parts que s'executen de manera paral·lela i coordinada.
- **Desenvolupar algorismes amb un cost computacional reduït**. Poden aplicar-se algorismes estocàstics, regles o simplificacions per agilitar el procés de cerca.

Apèndix F

Funcions de distància

F.1 Introducció

Les funcions de distància són un mecanisme que ens permet mesurar com de pròxims estan dues abstraccions a l'espai. Una abstracció es defineix com la representació d'una situació o objecte que es vol analitzar, on cadascuna d'elles es defineix mitjançant un conjunt de característiques/atributs. La distància dins aquest espai determinarà com de semblants són. Les funcions de similitud es fan servir en molts altres àmbits a part del CBR, com per exemple:

- Algoritmes basats en el veí més pròxim (Cover i Hart, 1967; Hart, 1968; Dasarathy, 1991).
- Mètodes de raonament basats en la memòria (Stanfill i Waltz, 1986).
- Xarxes neuronals basades en *radial basis* (Broomhead i Lowe, 1988).
- Teoria de la ressonància adaptativa de les xarxes neuronals (Carpenter i Grossberg, 1987).
- Xarxes counterpropagation (Hecht-Nielsen, 1987).
- Mapes autoorganitzables (Kohonen, 1990).
- Aprenentatge competitiu (Rumelhart i McClelland, 1986)
- Patrons de reconeixement (Diday, 1974).
- Psicologia cognitiva (Tversky, 1977; Nosofsky, 1986).
- ...

És important tenir en compte que no existeix una funció de distància perfecte que vagi bé per qualsevol problema. L'error mínim que pot cometre un sistema per generalitzar la sortida en relació a l'entrada (Mitchell, 1980) s'anomena biaix. Cada algorisme d'aprenentatge té el seu propi biaix inherent degut a la seva estratègia de resolució i, per tant, l'elecció de la funció de distància influirà el biaix propi de l'algorisme d'aprenentatge.

Tots els algorismes d'aprenentatge han de tenir una certa capacitat per generalitzar, i està demostrat que no pot haver un algorisme que pugui generalitzar més acuradament que un altre, quan es sumen tots els seus possibles problemes (Schaffer, 1994) (suposant que només es disposa de les dades d'aprenentatge).

En canvi, determinats problemes es veuen afavorits per algun tipus concret de funció de similitud (Wolpert, 1993). Per tant, és una tasca de l'expert determinar quina és la funció més adient pel problema que es vol resoldre. Cal tenir en compte que encara que tinguin la millor funció de

similitud de totes, si les dades no han estat prèviament preparades (hi ha errors, valors desconeguts, ...) els resultats obtinguts estaran per sota dels resultats desitjables.

A continuació, s'introdueixen algunes de les funcions més habituals dins els sistemes CBR: de propòsit general, basades en la distància sobre conjunts de dades, i per atributs heterogenis.

F.2 Funcions de distància tradicionals

Aquest apartat presenta les funcions més utilitzades dins del CBR usades per tractar dades numèriques. Tot i que aquestes funcions no estan pensades per treballar directament amb atributs nominals, el tractament d'aquests atributs pot fer-se de diferents maneres: (1) no tenint en compte els atributs nominals; (2) considerant els atributs com totalment iguals (diferència igual a 0) o totalment diferents (diferència igual a 1, si es considera un espai normalitzat); (3) fent una conversió de les dades nominals a numèriques mitjançant algun tipus de quantificació. En el cas que cap de les anteriors alternatives pugui aplicar-se, caldrà fer servir funcions de distància específiques per aquest tipus d'atributs (veure l'apartat F.4).

F.2.1 Funció de Minkowski

Es basa en la distància del veí més pròxim (*Nearest Neighbour Algorithm* - NNA) (Bachelor, 1978) a partir de la mesura de la distància geomètrica en un espai multidimensional.

$$distància(x, y) = \sqrt[r]{\sum_{i=1}^{p} w_i \left|x_i - y_i\right|^r} \tag{F.1}$$

On:

- x i y són els casos a comparar.
- x_i i y_i representen l'atribut i dels casos x i de y respectivament.
- w_i és la ponderació de l'atribut i.
- p és el número d'atributs del cas.
- r el valor que dóna el nom a la funció: hamming ($r = 1$), euclidiana ($r = 2$) i cúbica ($r = 3$).

F.2.2 Distància de Chebychev

És apropiada en aquells casos en què es vol diferenciar entre els objectes a partir d'una dimensió.

$$distància(x, y) = max_{i=1}^{p} \left|x_i - y_i\right| \tag{F.2}$$

On:

- x i y són els casos a comparar.
- x_i i y_i representen l'atribut i dels casos x i de y respectivament.
- p és el número d'atributs del cas.

F.2.3 Distància entre dues matrius

La distància entre dues matrius és un càlcul que pot arribar a ser complex segons sigui la naturalesa de les matrius. Per exemple, si es disposa de dues matrius iguals pot aplicar-se l'algorisme F.1. En canvi, pot succeir que tinguin dimensions diferents fent que l'algorisme anterior no pugui aplicar-se. Davant d'això, pot ser interessant tenir en compte la distància més gran o la més petita entre dues files de la matriu com es fa a l'algorisme F.2. La correctesa d'aquesta estratègia dependrà del significat de les files. Una altra alternativa pot ser fer la mitja dels elements de les matrius, i calcular la distància com si fossin dos vectors.

Algorisme F.1: Càlcul bàsic de la distància entre dues matrius.

```
Funció distancia matrius v1(A,B és vector) és
    Per tot element del vector A fer
        Per tot element del vector B fer
            Calcula la distància entre el vector A i el vector B
            Acumula la distància
    distància=acumulat/número de comparacions
    retorna distància
```

Algorisme F.2: Càlcul millorat de la distància entre dues matrius.

```
Funció distancia matrius v2(A,B és vector) és
    Per tot element del vector A fer
        Per tot element del vector B fer
            Calcula la distància entre vector A i el vector B
            Acumula distància només si és la mínima
    distància=acumulat/número de comparacions
    retorna distància
```

F.2.4 Distància de Camberra

$$distància(x,y) = \sum_{i=1}^{p} \left| \frac{x_i - y_i}{x_i + y_i} \right| \tag{F.3}$$

On:

- x i y són els casos a comparar.
- x_i i y_i representen l'atribut i dels casos x i de y respectivament.
- p és el número d'atributs del cas.

F.2.5 Distància Quadràtica

$$distància(x,y) = (x-y)^T Q(x-y) = \sum_{j=1}^{p} \left(\sum_{i=1}^{p} (x_i - y_i) q_{ij} \right) \cdot (x_j - y_j) \tag{F.4}$$

On:

- x i y són els casos a comparar.
- x_i i y_i representen l'atribut i dels casos x i de y respectivament.
- p és el número d'atributs del cas.
- Q és la relació dels atributs.
- q_{ij} és la ponderació de la relació entre l'atribut i i j.

F.2.6 Distància basada en la Correlació mostral

$$distància(x,y) = \frac{\sum_{i=1}^{p} (x_i - \bar{x_i})(y_i - \bar{y_i})}{\sqrt{\sum_{i=1}^{m} (x_i - \bar{x_i})^2 (y_i - \bar{y_i})^2}} \tag{F.5}$$

On:

- x i y són els casos a comparar.
- x_i i y_i representen l'atribut i dels casos x i de y respectivament.
- p és el número d'atributs del cas.
- $\bar{x}$ i $\bar{y}$ són les mitges dels atributs x i y.

F.2.7 Distància Chi-Quadrat

$$distància(x,y) = \sum_{i=1}^{p} \frac{1}{sum_i} \left(\frac{x_i}{size_x} - \frac{y_i}{size_y} \right)^2 \tag{F.6}$$

On:

- x i y són els casos a comparar.
- x_i i y_i representen l'atribut i dels casos x i de y respectivament.
- p és el número d'atributs del cas.
- sum_i és la suma de tots els valors de l'atribut i que hi ha en el conjunt d'entrenament.
- $size_x$ és la suma de tots els valors del cas x.

F.2.8 Distància de la correlació de la classificació de Kendall's

$$distància(x,y) = 1 - \frac{2}{n \cdot (n-1)} \sum_{i=1}^{p} \sum_{j=1}^{i-1} sign(x_i - x_j) sign(y_i - y_j) \tag{F.7}$$

On:

- x i y són els casos a comparar.
- x_i i y_i representen l'atribut i dels casos x i de y respectivament.
- p és el número d'atributs del cas.
- $sign(x) = -1$ si $x < 0$, $sign(x) = 0$ si $x = 0$, i $sign(x) = 1$ si $x > 0$.

F.3 Funcions de distància sobre conjunts de dades

En alguns casos, per exemple en aplicacions de clustering, és interessant calcular la diferència d'un element respecte tot un conjunt d'elements. Aquest apartat introdueix dues de les tècniques més utilitzades per solventar aquesta tasca.

F.3.1 Distància de Mahalanobis

Aquest mètode crea clústers que aglutinen dades que tenen algun tipus de propietats en comú (e.g. una classificació o solució a un problema). Els clústers es construeixen a partir de la matriu de covariàncies, la qual indica com varia un atribut d'un cas en relació a un altre (Nadler i Smith, 1993).

Per cada clúster que s'ha definit, es calcula la distància de l'element a mesurar respecte tots els clústers independentment de la forma que tenen a l'espai (esfera, el·lipsoide) i del rang dels atributs que el formen (el mètode contempla tot això en les fórmules, té implícit translacions espaials i la normalització de les dades). L'element pertanyerà al clúster respecte el que tingui la distància més petita. Té com a dificultat que necessita tenir informació precisa sobre mitges i desviacions típiques de les dades.

$$distància(x,y) = [detV]^{\frac{1}{m}} (x-y)^T V^{-1} (x-y) \tag{F.8}$$

On:

- x i y són els casos a comparar.
- V és la matriu de covariàncies de $A_1..A_m$, on A_j és el vector de valors de l'atribut j del conjunt d'entrenament.

Un exemple de l'aplicació d'aquesta tècnica s'analitza en (Vallespí, 2002). En aquest treball es vol estudiar si les μCa de les mamografies són benignes o malignes. Es creen dos clústers (a partir de la matriu de covariàncies de les μCa que són cancerígenes i les que no) que intenten representar aquests conjunts, i es calcula la mesura de cada μCa respecte aquests conjunts.

F.3.2 Distància a partir de la tècnica de *k-clustering*

El *k-clustering* (Vallespí, 2002) està basat en la filosofia de la clusterització, és a dir, en l'agrupació d'exemples que compleixen unes propietats.

A partir del conjunt d'exemples es busquen les potencials agrupacions que poden formar per determinar 'k' clústers. Inicialment es creen uns centroïds (centres del clúster) de manera aleatòria (són determinats per les mitges) i fins que quedi estable la ubicació dels exemples es va recalculant la seva ubicació. A partir de la ubicació final dels exemples i de la determinació dels centres dels clústers, es fa servir un classificador per associar la solució del clúster més proper. L'algorisme F.3 detalla aquest funcionament.

El fet de crear agrupacions entre els exemples encara que no estiguin agrupats per classes pot aprofitar-se per reduir espais de cerca. Suposem que es disposa d'una base de dades d'exemples molt gran i que està clusteritzada. Si hem de fer una cerca sobre la base de dades d'exemples, es pot buscar a quin clúster pertany l'exemple a buscar i, a partir d'això, comparar només amb els exemples d'aquest clúster.
No obstant, el mètode presenta diferents problemes o dificultats:

- La inicialització de les mitges pot condicionar el resultat final.
- Poden haver-hi centroids sense cap exemple assignat.
- Una classe pot tenir més d'un centroide.
- És difícil determinar la funció de distància que mesura a quin clúster pertany un exemple.
- És difícil determinar quants clústers es fan servir.

F.4 Funcions de distància per atributs heterogenis

La gran majoria de les funcions de distància estan basades en dades numèriques. No obstant, molts cops les dades amb les quals es treballa contenen informació en forma d'atributs nominals i numèrics conjuntament. Això implica la necessitat d'aplicar processos sobre les dades nominals per convertir-les en numèriques, o definir mecanismes per poder treballar de manera simultània amb atributs de naturaleses diferents. Aquest punt detalla les funcions introduïdes per Wilson (Wilson i Martinez, 1997) per tractar amb dades heterogènies.

Les funcions heterogènies HVDM, IVDM, DVDM i WVDM que es detallen a continuació requereixen d'una definició per tal de comparar dades nominals. Aquest pas el realitzen a través de la mètrica VDM explicada també en aquest apartat.

Algorisme F.3: Càlcul dels centroides en *k-clustering*.

```
Funció calcula_cetròids() és
    Sigui k el nombre de clústers
    Sigui m_i la mitja dels exemples en el clúster i
    S'inicialitza m_i de manera aleatòria
    Mentre hi hagi canvis en les mitges fer
        //Fer servir les mitges estimades per classificar els exemples en clústers
        Per i=1 fins a k fer
            Reemplaça la mitja del clúster i pel resultat de calcular la mitja de tots els exemples assignats a
            aquest clúster
    retorna distància
```

F.4.1 *Heterogeneous Euclidean-Overlap Metric*

La mètrica *Heterogeneous Euclidean-Overlap Metric* - HEOM (Wilson i Martinez, 1997) permet treballar de manera simultània amb atributs numèrics i nominals. El mecanisme que fa servir consisteix a aplicar fórmules diferents segons la naturalesa de l'atribut (vegeu les equacions F.9 i F.10).

$$distància_a(x,y) = \begin{cases} 1 & \text{, si } x \text{ o } y \text{ són desconeguts} \\ overlap(x,y) & \text{, si } a \text{ és nominal} \\ rn_diff(x,y) & \text{, altrament} \end{cases}$$

$$overlap(x,y) = \begin{cases} 0 & \text{, si } x{=}y \\ 1 & \text{, altrament} \end{cases} \tag{F.9}$$

$$rn_diff_a = \frac{|x-y|}{max_a - min_a} \tag{F.10}$$

Normalment el valor retornat per 'd' està comprès entre 0 i 1, independentment del seu tipus (numèric o nominal). La distància respecte tots els atributs es calcula com:

$$HEOM(x,y) = \sqrt{\sum_{a=1}^{m} d_a(x_a - y_a)^2} \tag{F.11}$$

F.4.2 *Value Difference Metric*

Value Difference Metric - VDM (Wilson i Martinez, 1997) és una mètrica definida per comparar dades nominals, tot i que també pot aplicar-se per dades numèriques. En aquest últim cas, cal discretitzar-les (Lebowitz, 1985; Schlimmer, 1987). El processament està basat en comptar quants atributs del mateix valor hi ha. Cal anar amb compte perquè aquest pas previ pot degradar els resultats si no es fa una discretització correcta (Ventura i Martinez, 1995). La fórmula simplificada sense tenir en compte la ponderació dels pesos és la següent:

$$vdm_a(x,y) = \sum_{c=1}^{C} \left| \frac{N_{a,x,c}}{N_{a,x}} - \frac{N_{a,y,c}}{N_{a,y}} \right|^q = \sum_{c=1}^{C} |P_{a,x,c} - P_{a,y,c}| \tag{F.12}$$

On:

- $N_{a,x}$ és el nombre d'instàncies del conjunt d'entrenament amb el valor x en l'atribut a.
- $N_{a,x,c}$ és el nombre d'instàncies del conjunt d'entrenament que tenen el valor x en l'atribut a i la sortida de la classe 'c'.
- C és el nombre de classes sortida en el domini del problema.
- q és una constant que val 1 o 2.
- $P_{a,x,c}$ és la probabilitat condicional que si és de la classe c, l'atribut a té valor x.

D'aquesta fórmula podem extrapolar:

- $N_{a,x}$ és la suma de totes les $N_{a,x,c}$.
- La suma de totes les $P_{a,x,c}$ serà 1, donat un valor de a i x.

La fórmula considera més similars aquells casos que tenen la mateixa classificació, és a dir, tenen una major correlació en la sortida. Per exemple, si tenim 3 colors (vermell, verd i blau) són més similars el vermell i verd, que el vermell i blau si la classificació a realitzar és identificar una 'poma'. L'equació original fa servir pesos en els atributs.

Aquesta fórmula presenta una deficiència que pot ser crítica en alguns casos. Què succeeix si en la fase de test apareixen nous valors de l'atribut nominals que no eren presents en l'entrenament? Si succeeix això aleshores $P = 0/0$, operació que té un valor indeterminat. Un criteri pot ser assignar a la probabilitat el valor de 0, ja que no tenim manera de calcular la probabilitat. Una altra opció pot ser considerar $P = 1/C$, on C serà el nombre total de classes. Fixeu-vos que si realitzem el sumatori d'aquest últim pas tindrem un 1, valor que s'ha comentat abans.

Pot comprovar-se que si s'aplica directament amb atributs numèrics, $P = 0/0$ serà present molts cops, ja que és molt difícil tenir valors repetits en un domini numèric. Una manera de resoldre aquest problema és discretitzant els valors del domini, tot i que això pot implicar una pèrdua d'informació i precisió que pot ser crítica o no, segons com es realitzi la discretització i com de crítiques siguin les dades. Per tant, aquest mètode no és el més apropiat per dominis majoritàriament numèrics.

F.4.3 *Heterogeneous Value Difference Metric*

Heterogeneous Value Difference Metric - HVDM està basat en VDM però utilitza un esquema diferent de càlcul. Té l'avantatge de permetre treballar amb atributs nominals i numèrics de manera correcta. A més a més, s'obté una millora en el rendiment (Wettschereck et al., 1995) (Atkeson et al., 1996) a l'introduir ponderació en els atributs. La distància es calcula de la manera següent:

$$HVDM(x,y) = \sqrt{\sum_{a=1}^{m} distància_a(x_a - y_a)^2}$$

on la distància es calcula segons:

$$distància_a(x,y) = \begin{cases} 1 & \text{, si } x \text{ o } y \text{ són desconeguts} \\ normalized\_vdm_a(x,y) & \text{, si } a \text{ és nominal} \\ normalized\_diff_a(x,y) & \text{, si } a \text{ és numèric} \end{cases} \quad \text{(F.13)}$$

On:

- m és el nombre d'atributs.
- normalized_vdm i normalized_diff són les funcions de normalització de l'apartat E.2.2.2 de l'apèndix.

Tot i que la fórmula té una arrel quadrada, aquesta no es fa servir quan es comparen distàncies de diferents abstraccions pel veí més pròxim, ja que l'arrel no afecta l'ordenació dels elements. En canvi, per alguns models molt concrets sí que és necessari, per exemple en *distance-weighted k-nearest neighbour* (Dudani, 1976). Encara que HVDM és similar a HEOM, HVDM proporciona millors resulats (Wilson i Martinez, 1997).

F.4.4 *Interpolated Value Difference Metric*

L'objectiu de *Interpolated Value Difference Metric* - IVDM (Wilson i Martinez, 1997) és introduir funcions de distància que permetin aplicar directament VDM sobre atributs continus. La VDM original fa servir l'estadística calculada dels conjunts d'entrenament per determinar la probabilitat $P_{a,x,c}$, que indica la probabilitat que sigui de la classe c si es presenta el valor x de l'atribut a.

Quan es fa servir IVDM els valors continus es discretitzen en s intervals de la mateixa amplada, on s és un valor indicat per l'usuari. No existeix un criteri per assignar el valor de s, valors molt alts redueixen la força estadística dels valors de la probabilitat, mentre que un valor petit no permet discriminar correctament el valor. Es pot considerar que s no és crític si és més petit que

Algorisme F.4: Càlcul de les Pa, v, c en IVDM.

```
Funció calcul_probabilitats(C: conjunt_entrenament) és
    Per tota instància del conjunt d'entrenament fer
        Per cada instància del conjunt d'entrenament fer
            Sigui x el valor de l'entrada de l'atribut a de l'element i
            v=discretitza_a(x)
            Sigui c la classe a la que pertany
            Incrementa N_{a,v,c} una unitat
            Incrementa N_{a,v} una unitat
        Per tot valor v discret de l'atribut a fer
            Per tota classe c fer
                Si (N_{a,v} == 0) llavors
                    P_{a,v,c} =0
                Sinó
                    P_{a,v,c} = N_{a,v,c} / N_{a,v}
        retorna P_{a,v,c}
```

el nombre d'instàncies i major que el nombre de classes possibles del problema. A partir del valor s es calcula l'amplada de cada interval dels atributs, la qual es farà servir per discretitzar el valor de l'atribut (vegeu l'equació F.15).

$$w_a = \frac{|max_a - min_a|}{s} \tag{F.14}$$

$$discretitza(x) = \begin{cases} x & \text{, si } a \text{ és discret} \\ s & \text{, si } x \text{ és igual } max_a \\ [x - min_x / w_a] + 1 & \text{, altrament} \end{cases} \tag{F.15}$$

On:

- a representa un atribut continu.
- s representa el nombre d'intervals.
- w_a és l'amplada de l'atribut a.
- max_a i max_i representa els valors màxim i mínim de l'atribut.

A l'algorisme F.4 es mostra com a partir dels valors discretitzats es calculen les probabilitats de pertànyer a una classe segons els valors de l'atribut a. IVDM fa servir una nova funció de normalització la qual té en compte la naturalesa de l'atribut. Ara s'interpola la influència de les probabilitats dels valors dels atributs associats a cada possible classificació.

$$ivdma(x, y) = \begin{cases} vdm_a(x, y) & \text{, si } a \text{ és discret} \\ \sqrt{\sum_{c=1}^{C} |P_{a,c}(x) - P_{a,c}(y)|^2} & \text{, altrament} \end{cases} \tag{F.16}$$

On:

- vdm_a serà la fórmula de l'equació E.7 de la normalització basada en VDM.
- $P_{a,c}$ representa la probabilitat interpolada.

Quan l'atribut és de naturalesa no discreta, es calcula la interpolació de les probabilitats dels dos rangs en el qual es troba.

$$P_{a,c}(x) = P_{a,u,c} + \left(\frac{x - mid_{a,u}}{mid_{a,u+1} - mid_{a,u}} \right) (P_{a,u+1,c} - Pa, u, c)$$

tal que:

$$mid_{a,u} \leq x \leq mid_{a,u+1}$$
$$mid_{a,u} = min_a + witdh_a(u + 0.5) \tag{F.17}$$

On:

- mid_a representa el punt mig dels intervals entre els que està el valor real de l'atribut a.
- min_a representa el valor mínim de l'atribut a.
- $witdh_a$ representa la diferència entre el valor mínim i màxim de l'atribut a.
- $P_{a,u,c}$ és la probabilitat del valor en el rang discret u.

Finalment es calcula la contribució que fan tots els atributs amb l'equació F.18. Aquesta interpolació permet ajustar d'una manera més precisa la probabilitat que un cert valor pertanyi a un interval o altre, obtenint d'aquesta manera millors resultats.

$$IVDM(x, y) = \sqrt{\sum_{a=1}^{m} ivdm_a(x_a, y_a)^2} \tag{F.18}$$

F.4.5 *Discretized Value Difference Metric*

Discretized Value Difference Metric - DVDM es centra en la discretització dels atributs continus com IVDM, però no té en compte les probabilitats. En aquest cas els atributs continus després no es tornen a fer servir.

La semblança dels atributs es pot calcular amb qualsevol de les fórmules explicades a l'apartat de normalització basada en VDM, tot i que és recomanable fer servir l'equació E.7. En l'equació F.19 es representa la manera com es calcula la diferència entre dos exemples, la qual es basa en l'equació F.15.

$$DVDM(x, y) = \sqrt{\sum_{a=0}^{m} |vdm_a(discretitza_a(x_a), discretitza_a(y_a)|^2} \tag{F.19}$$

F.4.6 *Widowed Value Difference Metric*

Widowed Value Difference Metric - WVDM (Wilson i Martinez, 1997) es fonamenta en la idea de la tècnica IVDM amb l'objectiu de trobar més punts per fer més precís el valor de p segons el valor de l'atribut.

En comptes de calcular la probabilitat només pels punts mitjos dels intervals com en IVDM, busca el valor de $P_{a,x,c}$ per a cada valor x de l'atribut a del conjunt d'entrenament. En WVDM els rangs de discretització no es fan servir pels atributs continus, només es fan servir per calcular la finestra w_a com es fa a DVDM i IVDM. Per a cada valor de x de l'atribut a del conjunt d'entrenament, es calcula la p trobant les instàncies que tinguin un valor $x \pm w_a/2$, i d'aquesta manera calcular les $N_{a,x,c}$ i $N_{a,x}$. Ara el nombre de punts és variable en comptes de tenir s punts sempre. A l'algorisme F.5 es mostra el procés d'aprenentatge d'aquestes probabilitats.

A partir d'aquests càlculs, es calcula la interpolació de les probabilitats tal i com es mostra a l'algorisme F.6. L'algorisme agafa un valor x de l'atribut a i retorna un vector de c probabilitats. Primer realitza una cerca per trobar dues instàncies consecutives per a l'atribut a que envolten el valor de x. La probabilitat per a cada classe s'interpola per aquests límits. Amb aquestes probabilitats, els càlculs es realitzen com a IVDW pel càlcul de la similitud de dos atributs:

$$wvdma(x, y) = \begin{cases} vdm_a & \text{, si } a \text{ és discret} \\ \sqrt{\sum_{c=1}^{C} |P_{a,x,c}(x) - P_{a,c}(y)|^2} & \text{, altrament} \end{cases} \tag{F.20}$$

Algorisme F.5: Aprenentatge de l'algorisme WVDM.

Funció *aprenentatgeWDVM(C: conjunt d'exemples d'entrenament)* **és**
- Sigui n el nombre d'instàncies
- Sigui *instància[a][1..n]* una llista ordenada de totes les instàncies T ordenades ascendentment per l'atribut a
- Sigui *instància[a][i].val* el valor de l'atribut a de la instància i
- Sigui x el valor central de la finestra
- Sigui $p[a][i][c]$ és la probabilitat de $P_{a,x,c}$
- Sigui $N[c]$ el nombre $N_{a,x,c}$ d'instàncies de la finestra actual de la classe c
- Sigui N el nombre $N_{a,x}$ d'instàncies de la finestra actual
- Sigui *instància[a][in]* la primera instància de la finestra
- Sigui *instància[a][out]* la primera instància fora de la finestra
- Sigui $w[a]$ l'amplada de la finestra per l'atribut a
- **Per tot** *atribut continu a* **fer**
  - Ordena *instància[a][1..n]* en ordre ascendent amb un *quicksort*
  - Inicialitza $N = N[c] = 0$ i $in = out = 1$ (comencen per la finestra buida)
  - **Per** *i=0* **fins a** n **fer**
    - Sigui x = *instància[a][i].val*
    - //S'amplia la finestra fins al seu límit
    - **Mentre** *(out<n) & (instància[a][out].val<(x + w[a]/2))* **fer**
      - Incrementa $N[c]$, on c és la classe de *instància[a][out]*
      - Incrementa N
      - Incrementa *out*
    - //S'elimina qui no pertanyi al rang per darrera
    - **Mentre** *(in<out) & (instància[a][in].val< (x-w[a]/2))* **fer**
      - Decrementa $N[c]$, on c és la classe de *instància[a][in]*
      - Decrementa N
      - Incrementa *in*
    - //Es calculen les probabilitats
    - **Per tota** *classe, on c=1..C* **fer**
      - $p[a][i][c]=N[c]/N$
- **retorna** p

Algorisme F.6: Càlcul de la interpolació de les probabilitats en l'algorisme WVDM.

Funció *Probabilitat_WVDM(atribut a, valor continu x)* **és**
- //Cal trobar $P_{a,x,1..c}$ per $c = 1..C$, a partir del valor x de l'atribut a
- Troba la i tal que instància[a][i].val<=x<= instància[a][i+1].val (cerca binària)
- $x1$=*instància[a][i].val[a]* (menys per i<1 ja que llavors $x1 = min[a] - (w[a]/2)$)
- $x2$=*instància[a][i+1].val[a]* (menys per i>n ja que llavors $x2 = max[a] + (w[a]/2)$)
- **Per** *c=1* **fins a** C **fer**
  - $p1 = p[a][i][c]$ (menys per $i < 1$, aleshores $p1 = 0$)
  - $p2 = p[a][i + 1][c]$ (menys per $i > n$, aleshores $p2 = 0$)
  - $P_{a,x,c}$=$p1 + ((x - x1)/(x2 - x1)) * (p2 - p1)$
- **retorna** $P_{a,x,1..c}$

On vdm_a es calcula segons l'equació E.7 de la normalització basada en VDM.

Finalment, es calcula la contribució que fan tots els atributs:

$$WVDM(x, y) = \sqrt{\sum_{a=1}^{m} wvdm_a(x_a, y_a)^2} \tag{F.21}$$

El mètode permet refinar més acuradament la probabilitat d'un atribut en relació a la classe a la qual pertany. L'inconvenient és que requereix realitzar més càlculs i emmagatzemar més dades.

Apèndix G

La complexitat de les dades i el SOMCBR

G.1 Introducció

SOMCBR (*Self-Organizing Map in a Case-Based Reasoning system*) és un CBR caracteritzat per organitzar la memòria de casos mitjançant un Mapa Autoorganitzatiu (SOM). Això permet potenciar l'accés i contingut de la memòria amb el coneixement descobert per SOM per tal de millorar el rendiment de totes les fases gràcies al nou coneixement.

No obstant, les millores fruït d'aquesta organització estan directament vinculades a la capacitat dels clústers per modelar la geometria de les dades. Per aquest motiu es fa necessari conèixer aquesta vinculació. L'objectiu d'aquest apèndix és definir un espai de complexitats que ens ajudi a controlar aquesta relació.

G.2 Mètriques de complexitat

La complexitat de les dades fa referència a la caracterització de la seva complexitat intrínseca, i a l'estudi del seu impacte sobre el rendiment del classificador (Basu i Ho, 2006). De manera general, la complexitat de les dades està vinculada a tres causes: (1) l'ambigüitat de classe, (2) la complexitat de la frontera, i (3) la diversitat del conjunt d'entrenament. No obstant, degut a la dificultat de determinar els aspectes 1 i 3, els estudis actuals es centren en l'aspecte 2.

Ho & Basu (Ho i Basu, 2002) van proposar al 2002 un espai de mesures per identificar els diferents aspectes de la complexitat de la frontera basat en:

El poder discriminant dels atributs. La propietat fa referència al pes específic que té un atribut per discernir entre diverses classes. Les mètriques més rellevants són:

- F1. Basada en el càlcul del discriminant de Fisher.
- F2. Basada en el càlcul del solapament de les cues de les distribucions de les classes.
- F3. Basada en l'eficiència individual dels atributs per discernir entre les classes.

La separabilitat de classes. La propietat fa referència a si les classes són linealment separables. Les mètriques més rellevants són:

- L1. Basada en la minimització de l'error pel mètode de la programació lineal (Smith, 1968).
- L2. Basada en l'error d'un classificador lineal pel mètode de la programació lineal.

- N1. Basada en el percentatge de punts que defineixen els límits de la classe.
- N2. Basada en la dispersió de les classes.
- N3. Basada en el percentage d'error d'un classificador 1-NN.

La topologia de les classes. La propietat es refereix a la possibilitat de què existeixin subestructures dins les dades, i a variacions en les distribucions.

- L3. Basada en l'estudi de la no linealitat d'un classificador pel mètode de la programació lineal.
- N4. Basada en l'estudi de la no linealitat d'un classificador 1-NN.

Com que no totes les mètriques estan igualment correlacionades amb el rendiment dels classificadors, es fa necessari un estudi per veure quines d'elles poden aportar-nos major informació per discernir entre les tipologies de dades existents segons el seu rendiment. Aquest és precisament l'objectiu del punt següent.

G.3 Estudi de la correlació entre les mètriques i el SOMCBR

La taula G.1 detalla els *datasets* de les diferents característiques i dominis seleccionats del *UCI Repository* (Asuncion i Newman, 2007) per tal d'avaluar la correlació entre les mètriques i el SOMCBR. Per la manera com estan definides les mètriques de complexitat utilitzades (Ho i Basu, 2002), els *datasets* de J classes han estat convertits a J *datasets* de dues classes (cada classe contra la resta de classes) per tal d'incrementar el joc de dades. Per això tots tenen el sufix cX, on X representa la classe que s'avalua respecte la resta.

L'estudi de la correlació entre ambdues parts consta de 3 etapes. Primer, cal executar el CBR i el SOMCBR sobre els *datasets* de la taula G.1 per poder comparar el canvi en el rendiment entre ambdós sistemes. El següent pas és avaluar les mètriques de complexitat de l'apartat anterior sobre els *datasets* de la taula G.1. Finalment, cal establir una relació entre els paràmetres p-value i %R i les mètriques.

Les configuracions a estudiar pel CBR i pel SOMCBR a estudiar es poden resumir en els punts següents:

- La funció de distància utilitzada tant per la construcció dels models, com per la comparació entre clústers i casos és la del complement de la funció Euclidiana (vegeu l'equació 5.1).
- La mida del mapa s'assigna de manera automàtica tal com s'explica al capítol 3, és a dir, es selecciona la mida que minimitza l'error. El rang de mides avaluades va de 2 a 6.
- Els models poden tenir diferent nombre de casos.
- La fase de recuperació fa servir només els casos del model més semblant perquè es vol avaluar la capacitat que té cada model per representar les seves dades.
- La fase d'adaptació proposa la nova solució fent servir el cas recuperat més semblant.
- La fase d'emmagatzematge no guarda nous casos.
- Cada resultat és el resultat d'un *10-fold stratified cross-validation*.
- Cada configuració és la mitja de 10 llavors per tal de compensar els efectes aleatoris de la construcció dels models.

Els resultats obtinguts estan resumits també a la taula G.1. %Encert és el percentatge d'encerts pel CBR i pel SOMCBR, i σ la seva desviació típica; %R és la reducció del nombre d'operacions entre el CBR i el SOMCBR. p-value és la probabilitat de rebutjar l'hipotesi nul·la que assumeix la igualtat entre els %Encert (Sheskin, 1997). Valors petits de p-value indican una alta probabilitat de que hi hagi diferències significatives entre els %Encert . Per tant, aquests dos últims paràmetres ens permeten saber la variació de rendiment. En funció d'aquests valors, la taula es divideix (mitjançant una línia horitzontal) en dues caterogies segons si SOM és capaç o no de segmentar el domini en patrons:

- El **Tipus 1** fa referència a situacions on el temps computacional es millora i el percentatge d'encerts es manté.
- El **Tipus 2** fa referència a situacions on el percentatge d'encerts és proporcional als casos recuperats i, conseqüentment, el percentatge depèn dels casos recuperats.

Pel que fa al càlcul de les mètriques, primer s'ha fet una preselecció de mètriques que són potencialment interessants. Aquestes han estat les mètriques F1, F2, F3, N1, N2, N3 i N3, les quals estan vinculades als classificadors basats en el K-NN per la manera d'estar definides.

Taula G.1: Resum dels *datasets* utilitzats (nom, nombre d'instàncies i d'atributs), els percentatges d'encerts (%Encert) del CBR i el SOMCBR amb les desviacions típiques (σ), el paràmetre de comparació entre els %Encert (p-value), i el percentatge de reducció del nombre d'operacions en recuperar l'element més similar (%R). La taula mostra els resultats de les mètriques més correlacionades amb els paràmetres p-value i %R, els quals divideixen els *datasets* en dos segments mitjançant una líniea horitzontal.

Dataset			**Mètriques**			**CBR**	**SOMCBR**	**Estadístiques**	
Nom	**Inst.**	**Attr.**	**N1**	**N2**	**F3**	**%Encert(σ)**	**%Encert(σ)**	**%R**	**p-value**
waveform c1	5000	41	0.24	0.86	0.23	83.2 (1.2)	81.1 (1.2)	89.2	0.00
vehicle c1	846	19	0.12	0.42	0.46	93.4 (2.4)	87.5 (4.7)	86.9	0.00
vehicle c4	846	19	0.09	0.54	0.22	96.0 (4.2)	89.2 (3.6)	87.7	0.00
balance c2	625	5	0.20	0.62	0.00	87.0 (3.1)	81.8 (4.1)	89.7	0.00
waveform c2	5000	41	0.27	0.90	0.15	80.2 (1.4)	78.8 (1.6)	89.7	0.01
pim	768	9	0.44	0.84	0.01	71.3 (3.4)	69.9 (3.4)	87.9	0.03
wpbc	198	34	0.42	0.91	0.18	73.7 (7.1)	73.2 (9.2)	82.5	0.03
waveform c3	5000	41	0.23	0.85	0.24	83.6 (1.8)	82.7 (1.6)	89.3	0.03
balance c3	625	5	0.20	0.62	0.00	86.9 (3.7)	82.3 (6.5)	89.5	0.04
tao	1888	3	0.07	0.16	0.36	95.4 (1.3)	94.9 (1.6)	81.8	0.06
wdbc	569	31	0.07	0.56	0.52	95.1 (3.2)	95.3 (2.7)	80.2	0.09
wbcd	699	10	0.06	0.34	0.12	95.3 (2.2)	94.6 (2.6)	86.9	0.09
vehicle c3	846	19	0.37	0.74	0.06	73.9 (4.1)	73.4 (4.5)	82.5	0.11
vehicle c2	846	19	0.37	0.71	0.04	75.3 (3.4)	75.4 (2.9)	81.9	0.11
bpa	345	7	0.58	0.91	0.03	62.9 (6.0)	63.2 (5.1)	52.6	0.17
heart-statlog	270	14	0.37	0.67	0.01	74.1 (6.4)	76.3 (8.3)	87.1	0.19
balance c1	625	5	0.21	0.65	0.00	83.7 (2.2)	86.1 (4.9)	89.0	0.21
wisconsin	699	10	0.06	0.33	0.12	96.1 (2.0)	96.9 (2.4)	84.5	0.33
ionosphere	351	35	0.23	0.63	0.19	86.9 (4.1)	88.1 (3.6)	64.0	0.41
iris c2	150	5	0.01	0.10	1.00	100.0 (0.0)	100.0 (0.0)	56.3	0.00
thyroids c2	215	6	0.06	0.23	0.81	98.1 (3.3)	97.2 (4.0)	52.8	0.01
thyroids c1	215	6	0.05	0.23	0.85	98.1 (3.3)	96.3 (4.3)	51.4	0.02
iris c1	150	5	0.09	0.17	0.75	95.3 (4.3)	93.3 (5.9)	60.7	0.04
wine c1	178	14	0.05	0.43	0.72	98.3 (3.7)	97.2 (5.1)	68.6	0.05
wine c2	178	14	0.07	0.49	0.76	97.2 (4.3)	97.2 (4.3)	67.9	0.05
thyroids c3	215	6	0.10	0.31	0.67	97.2 (4.0)	95.8 (4.4)	54.2	0.08
iris c3	150	5	0.10	0.21	0.56	94.7 (5.8)	93.3 (6.6)	60.9	0.08
wine c3	178	14	0.12	0.57	0.58	94.9 (5.2)	95.5 (4.9)	65.3	0.09

A partir dels resultats obtinguts, s'ha estudiat la definició d'un espai de complexitats mitjançant les mètriques que permetés la segmentació dels tipus 1 i 2 identificats anteriorment. El resultat d'aquest estudi ha estat la selecció de les mètriques N1, N2 i F3, els valors dels quals es troben també a la taula G.1. A més a més, com que N1 i N2 es refereixen a aspectes similars, es treballa amb el seu producte per potenciar el seu significat especialment en situacions de valors baixos.

La figura G.1 mostra les gràfiques més rellevants generades durant el procés de definició de l'espai de complexitats. La figura G.1 (a) mostra els valors de p-value i %R a través dels quals es realitza la definició dels tipus 1 i 2.

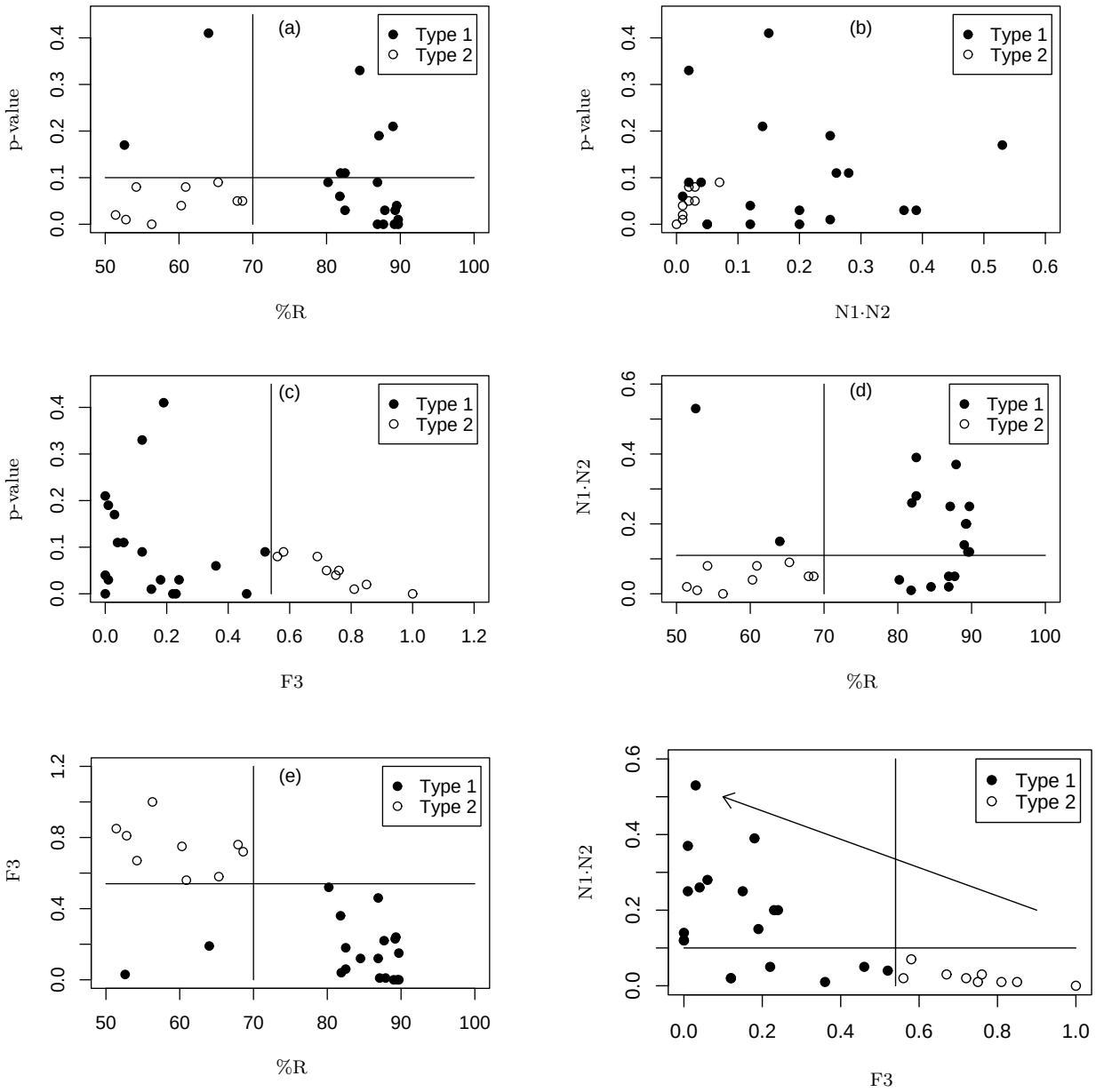

Figura G.1: Les gràfiques mostren les combinacions més destacades entre p-value, %R, i les mètriques de complexitt F3 i N1·N2. El gràfic (f) defineix un espai de complexitats que modela la viabilitat de SOM.

Les figures G.1(b, c, d, e) mostren la relació entre les mètriques (F3 and N1·N2) amb els valors p-value i $\%R$. A la figura G.1(b) pot observar-se que tots els *datasets* de tipus 2 estan a prop de la zona delimitada per valors petits de N1·N2 i p-value, però que hi ha alguns solapaments amb els *datasets* de tipus 1. D'altra banda la figura G.1(c) mostra que els problemes de tipus 2 estan seperats dels de tipus 1 respecte la mètrica F3. A més a més, els problemes de tipus 2 estan principalment relacionats amb valors alts de F3. Les figures G.1(d) i G.1(e) mostren resultats similars, on es dibuixa la relació però respecte el %R. En ambdues figures, els valors de F3 i N1·N2 defineixen regions separades en dos tipus.

Aquestes figures suggereixen diferents tendències: (1) Datasets amb alts valors de %R apareixen en regions amb valors baixos de F3. (2) Datsets amb alts valors de F3 tenen valors molt baixos de p-value. (3) Els valors baixos de %R estan estretament correlats amb els valors baixos del producte de N1 i N2.

La figura G.1(f) representa un espai de complexitats on les mètriques N1·N2 i F3 separen l'espai en 4 àrees. Aquestes mesures estableixen perfectament la regió pels *datasets* de tipus 2: alts valors de F3 (> 0.55) i valors molt baixos de N1·N2 (< 0.1). Un valor alt de F3 significa que hi ha una alta separabilitat de classes perquè els atributs no es troben solapats. Valors baixos de N1·N2 impliquen una alta separabilitat també. Tenint en compte això, la fletxa indica el sentit de la complexitat de les dades. Per tant, SOMCBR és una tècnica recomanable per dominis complexos (tipus 1), és a dir, dominis on les seves capacitats *Soft-Computing* permeten modelar molt bé aquest tipus de dominis.

Tot i que la figura G.1 parteix l'espai en quatre zones, hi ha una d'elles que geomètricament no té sentit (situació de valors alts de F3 i alts de N1·N2). Per tant, aquesta separació en quatre zones es podria mirar d'altra manera com mostra la figura G.2. El punt (1,0) es considera el punt de menor complexitat (mCP - *minimum complexity point*), i el punt (0,1) fa referència al punt de major complexitat (MCP - *maximum complexity point*). A partir d'aquests punts és possible definir tres tipus segons la zona:

- **Tipus** A: problemes amb una baixa complexitat (distància $<$ a 0.5 respecte el mCP).
- **Tipus** B: problemes amb una complexitat mitjana (distància entre 0.5 i 1 respecte el mCP).
- **Tipus** C: problemes amb una alta complexitat (distància > 1 respecte el mCP).

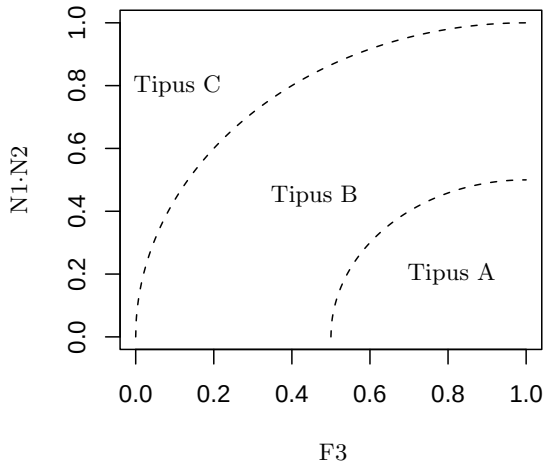

Figura G.2: El mapa de complexitats està definit mitjançant les mètriques N1, N2 i F3. La seva combinació defineix 3 zones, on A és la zona de menor complexitat i C la de major complexitat.

Apèndix H

Relevance Feedback

H.1 Cercar en el lloc adient segons la subjectivitat de l'usuari

Una de les més valuoses virtuts de les eines informàtiques és la seva capacitat per gestionar grans volums d'informació, ja siguin documents de text o d'imatges. Una de les fases més importants d'aquesta gestió és la definició dels criteris o preferències a través dels quals es recupera la informació. No obstant, en la presa d'aquesta decisió hi ha un factor molt difícil de definir: la subjectivitat de les persones. Totes les persones tenim una percepció pròpia de la realitat, la qual a més a més està condicionada per les nostres experiències i vessants de procedència. Així mateix, la complexitat i incertesa dels dominis reals contribueix a que un mateix aspecte pugui veure's des de diferents perspectives. Per tant, seria desitjable que el sistema pogués adaptar-se segons l'usuari que el fa servir, i aconseguir agilitzar el procés de recuperació. Dins d'aquest àmbit, les aplicacions basades en el Raonament basat en casos són un exemple de possible aplicació.

Les algorismes de *Relevance Feedback* són tècniques d'aprenentatge supervisat que introdueixen la subjectivitat de l'usuari en el procés de recuperació per tal d'ajustar la cerca segons les seves preferències. Tot i que van ser originàriament desenvolupades per la recuperació de texts de documents, el seu ús s'ha estès al camp de la cerca d'imatges per contingut. La seva aplicació permet reduir el forat existent entre el baix nivell de les imatges o les paraules, i els conceptes d'alt nivell definits per les persones. Per tant, la seva aplicació millora el rendiment de les cerques, tant en l'ajustament dels criteris de cerca, com el temps que necessita l'usuari en trobar el que busca. El funcionament d'aquests d'algorismes es pot descriure en els passos següents:

1. L'usuari fa una consulta a partir d'uns paràmetres, o bé, d'un exemple del qual vol trobar altres de similars.

2. El sistema retorna un conjunt de resultats.

3. L'usuari indica al sistema els exemples que considera com a bons (positius) i dolents (negatius), o bé, reajusta els paràmetres de cerca.

4. El sistema torna a repetir la consulta amb la informació rellevant retornada (*feedback*).

5. Es repeteixen els passos 3 i 4 fins que es consideren els resultats retornats com els desitjats.

Els algorismes de *Relevance Feedback* poden veure's com un problema d'aprenentatge, on un usuari proporciona exemples de *feedback* a partir dels primers resultats, i el sistema ha d'aprendre a refinar la consulta. Inicialment, gairebé tots els sistemes eren sistemes d'aprenentatge adaptats: arbres de decisió (MacArthur i C.E. Brodley, 2000), xarxes neuronals (Laaksonen et al., 1999), aprenentatge Bayesià (Vasconcelos i Lippman, 2000), etc. El problema amb el qual es trobaven

era que calien moltes dades per poder aprendre (com per exemple les xarxes neuronals o els arbres de decisió), i en aquest tipus de problemes es disposa de poques dades (exemples) amb moltes dimensions (atributs). Aquesta propietat del domini dificultava la seva aplicació per la dificultat d'entrenar-los.

Altres en canvi, afirmen que els algorismes de *Relevance Feedback* poden considerar-se un problema de classificació o de reconeixement de patrons. Sota aquesta consideració, els exemples positius i negatius poden considerar-se exemples d'entrenament que entrenen el classificador. A partir d'això, el classificador ha de separar les dades en grups de dades rellevants i irrellevants. Aquest enfocament ha estat implementat per diversos autors amb classificadors lineals (Wu i Huang, 2000), basats en el veí més pròxim (Wu i Manjunath, 2001), classificador Bayesià (Su i Kathleen, 2000) o SVM (Tong i Chang, 2001) entre d'altres.

El domini d'aplicació d'aquests algorismes és molt ampli i variat, des de dominis mèdics, motors de cerca d'imatge o processament del llenguatge natural, on el que es vol és millorar la qualitat i precisió de les consultes dels usuaris. Pot afirmar-se que el gran ventall d'aplicacions on poden incorporar-se aquest tipus d'algorismes cada cop és més gran, i contínuament es proposen noves tècniques que intenten compensar les mancances d'altres sistemes per tal de crear un valor afegit que els desmarqui de la competència. Per tant, aquests tipus d'algorismes són molt positius i permeten personalitzar i millorar la qualitat de les eines de cerca d'informació.

La finalitat d'aquest apèndix és presentar les propietats i vessants més característiques de les estratègies de *Relevance Feedback*.

H.2 Fonaments de les estratègies de *Relevance Feedback*

H.2.1 Propietats característiques dels algorismes

Els trets característiques dels algorismes de *Relevance Feedback* es poden dividir en les propietats següents:

Dades amb característiques de baix nivell i/o semàntiques. Fa referència a la manera com es representen les dades.

- Característiques de baix nivell. Són aquelles que es poden quantificar (per exemple que una frase tingui quatre paraules, o els resultats de l'histograma d'una imatge)
- Característiques semàntiques. Són aquelles on hi ha una informació més enllà de valors de baix nivell (per exemple el context d'una frase o el tipus de contingut d'una imatge).

Tot i que la definició de conceptes permet una representació més potent de la realitat, la seva definició pot ser complexa perquè les dades han d'etiquetar-se o conceptualitza-se. A part de l'esforç que això suposa, la subjectivitat de la persona que ho fa pot afectar molt.

Amb o sense memòria. Fa referència a la capacitat per memoritzar els *feedback* proporcionats per tots els usuaris.

- Sense memòria. El sistema no guarda les interaccions entre l'usuari i el sistema. Davant d'una mateixa consulta l'usuari ha de realitzar un altre cop tot el procés de cerca.
- Amb memòria. El sistema actualitza les relacions entre les dades amb la informació de l'usuari per realitzar les cerques més ràpidament.

La dificultat en aquests sistemes és el manteniment de la consistència de les subjectivitats dels diferents usuaris.

***Feedbacks* positius i negatius.** Els resultats que el sistema retorna es poden puntuar de diferents maneres per part de l'usuari:

- Marcar només els resultats positius.
- Marcar els resultats positius i els negatius.
- Ponderant de manera negativa els diferents resultats negatius.
- Ponderant conjuntament els resultats positius i els negatius.

Els algorismes que tenen en compte els casos negatius són capaços d'oferir millors resultats, sempre i quan la seva identificació per part de l'expert sigui correcte (Müller et al., 2000).

Cerca categòrica, per objectiu o navegatòria. Fa referència al tipus de necessitat que l'aplicació ha de realitzar (Cox et al., 2000). Els tipus de cerca a realitzar es poden dividir en:

- Cerca categòrica (*Category Search*). L'usuari busca a partir d'uns conceptes que representen un tipus d'informació.
- Cerca per objectiu (*Target Search*). L'usuari busca un element molt específic. Els exemples positius que proporciona són els més propers al que busca i, per tant, han de conduir-lo cap el resultat desitjat. Aquest enfocament té diversos problemes ja que d'una banda sovint és difícil establir quins són els exemples més similars i, d'altra banda hi ha la problemàtica que molts cops si és similar en un aspecte no ho és en un altre.
- Cerca navegatòria (*Browsing Search*). L'usuari no té cap objectiu ni patró prefixat a buscar, vol explorar per veure què troba. Són situacions en les que l'usuari pot canviar diversos cops de tipus d'objectiu durant la cerca. El seu criteri de cerca per exemple pot començar buscant imatges amb un color concret, i finalitzar buscant imatges amb un tipus de textura.

Consultes per exemples o per paraules clau. Representa la manera com l'usuari realitza les consultes sobre el sistema:

- Les consultes per exemples (*Query by example*). Es busca a partir d'un conjunt d'exemples a partir dels quals el sistema ha de buscar. Aquest tipus de cerca està normalment orientada en els sistemes CBIR (*Content-Base Image Retrieval*).
- Les consultes per paraules clau (*Query by keyword*). Es busca a partir d'un conjunt de paraules que representen uns conceptes.

Normalment els sistemes proporcionen només un dels dos enfocaments, encara que hi ha alguns que proporcionen ambdós mètodes per complementar les cerques (Chen et al., 2001).

La característica que més ha marcat l'evolució dels algorismes ha estat la definició de conceptes semàntics. Els apartats següents introdueixen breument alguns dels sistemes més representatius de les estratègies basades en característiques de baix nivell i en les caractarístiques semàntiques.

H.2.2 Cerques basades en propietats de baix nivell, els orígens

En sistemes basats en característiques de baix nivell, la cerca és fruït de l'aplicació d'una mètrica de distància sobre un conjunt d'exemples. A partir d'això, el sistema retorna una llista ordenada segons la similitud entre la instància inicial i les emmagatzemades. No obstant, aquest procés de cerca pot afrontar-se des de dos punts de vista:

- *Query point move.* L'objectiu del *feedback* es refinar la *query.* És un mètode que intenta estimar la separació entre els exemples bons i els dolents. Com a sistema característic d'aquest enfocament cal destacar el sistema MARS (Rui et al., 1997) de Yong Rui.

- *Re-weighting.* L'objectiu del *feedback* és refinar la manera de mesurar la similitud. Si un element està representat per diferents dimensions, amb aquest tipus de mètodes s'intenta donar més importància a les dimensions que influeixen més a l'hora de comparar dos elements. Un sistema pioner amb aquest enfocament va ser l'ImageRover (Sclaroff et al., 1997) de Stan Sclaroff.

A continuació es presenten aquests dos sistemes a detall d'exemple. A més a més, es presenta un tercer sistema basat en els Mapes autoorganitatius (Kohonen, 1984). Tot i que és una tècnica molt utilitzada, la seva aplicació no es recomanable per sistemes on s'hagin de memoritzar els *feedbacks* perquè el cost de reentrenament és costós.

H.2.2.1 El Sistema MARS

El sistema MARS (Rui et al., 1997) és un sistema de recuperació d'imatges que es va definir per estudiar l'impacte dels diferents tipus de representacions d'imatges. Proposa d'una banda una representació basada en la descomposició de colors, i d'altra banda una basada en matrius de correlacions construïda a partir d'escales de grisos. Per a cada representació es defineixen 10 i 8 característiques per imatge respectivament.

Es disposa de M imatges d'N característiques a la base de dades. A partir de la representació escollida, els N atributs de les imatges es converteixen en N pesos dels atributs de les imatges. Cada imatge i estarà representada per un vector D_i de pesos que representen la importància de cadascun dels termes. Aquesta conversió s'analitza de dues maneres diferents:

- Pesos basats en la freqüència. Es basa en la freqüència del terme en la imatge i en la resta d'imatges. Si és molt freqüent en una imatge vol dir que és un element important. Si té molta importància en totes les imatges, vol dir que no és diferenciador i per tant la seva importància no es tan gran.

- Mitjançant una normalització Gaussiana que representi els valors en el domini [-1..1].

El procés de recuperació és el següent:

1. L'usuari proposa una query Q formada per N elements que representen la importància que vol l'usuari de cadascun dels termes.

2. El sistema busca les imatges més similars. La similitud entre una imatge D_i i Q es calcula amb la distància del cosinus (Equació H.1)

$$sim(Q, D_i) = \frac{D_i Q}{||D_i||||Q||} \tag{H.1}$$

   On $||\_||$ representa la norma 2.

3. A partir dels resultats presentats, l'usuari marca els rellevants i els no rellevants.

4. Amb aquesta informació el sistema reajusta la *query* mitjançant la fórmula de Rocchio (Joachims, 1997)(Equació H.2)

$$Q' = \alpha Q + \beta(\frac{1}{N_{R'}} \sum_{i \in D'_R} D_i) - \gamma(\frac{1}{N_{N'}} \sum_{i \in D'_N} D_i) \tag{H.2}$$

   On:

- N'_R i N'_N representen el nombre d'elements rellevants i no rellevants respectivament.
- D'_R i D'_N representen el conjunt d'elements rellevants i no rellevants respectivament.
- α, β, γ són constants que mesuren la contribució del *feedback* respecte la *query* 'Q'.

Bàsicament aquest procés de *feedback* consisteix en reajustar la importància dels termes mitjançant mitges. Els pesos de la nova *query* Q' permetran tornar a realitzar la consulta i retornar els nous resultats.

5. Es repeteixen els passos anteriors de manera iterativa fins que l'usuari troba els resultats desitjats.

En aquest sistema, l'aplicació del *Relevance Feedback* té com a objectiu ajustar la importància dels termes que l'usuari busca per recuperar les imatges més similars segons els seus criteris, ja que sinó a priori és molt difícil determinar la importància d'aquests al tractar-se de conceptes de massa baix nivell.

H.2.2.2 El sistema ImageRover

ImageRover (Sclaroff et al., 1997) és un navegador d'imatges basat en el contingut, on un dels objectius fonamentals era proporcionar una eina de cerca d'imatges basada en un domini tan ampli i poc estructurat com és el WWW. L'eina disposa d'un conjunt de 38 processos ubicats en diferents màquines encarregades de recollir imatges d'Internet. Aquestes imatges són processades, i a partir de la informació obtinguda s'indexen i emmagatzemen en forma de vector. L'usuari busca la informació en el sistema a partir d'imatges d'exemples que proposa.

Degut a la gran quantitat d'imatges i l'elevada dimensionalitat d'aquestes, s'aplica una reducció mitjançant la tècnica de PCA (*Principal Component Analysis*) (McLachlan, 2004) i els resultats es guarden en forma d'un arbre que permet la seva indexació. D'aquesta manera, mitjançant un algorisme optimitzat es busca per l'arbre els veïns més pròxims a la imatge que s'ha proporcionat per l'usuari. L'usuari pot a més indicar el grau de velocitat i precisió en els resultats, fet que influirà molt en el temps d'execució. El procés de cerca en el sistema es pot esquematitzar en els següents passos:

1. L'usuari proporciona una imatge d'exemple sobre la que vol buscar exemples similars.

2. El sistema disposa de diferents mètriques basades en l'equació de Minkowski (Equació H.3). A partir d'aquesta funció de comparació retorna les imatges de la base de dades que més s'assemblen a la imatge d'exemple proporcionada per l'usuari. Inicialment s'estableix una equació per defecte, la qual anirà canviant segons els *feedbacks* de l'usuari.

$$\textit{distància}(x, y) = \sqrt[r]{\sum_{i=1}^{p} w_i \left| x_i - y_i \right|^r} \tag{H.3}$$

On:

- x, y són els casos a comparar.
- x_i , y_i és l'atribut i dels casos x i de y respectivament.
- w_i és la ponderació de l'atribut i.
- p és el número d'atributs del cas.
- r el valor que dóna el nom a la funció.

3. A partir dels resultats retornats l'usuari selecciona els més rellevants i torna a fer la consulta.

4. Amb aquesta informació, l'algorisme selecciona automàticament la mètrica de Minkowski apropiada que minimitza la distància mitja entre les imatges rellevants especificades per l'usuari. Amb aquesta funció seleccionada farà la següent cerca sobre la base de dades.

5. Es repeteixen els passos anteriors fins que l'usuari estigui satisfet amb els resultats.

Aquest procés de representació i indexació permet construir un sistema capaç d'agrupar coneixement sense cap estructura entenible per l'usuari a priori, ja que seran les característiques extretes mitjançant el PCA les que representaran el coneixement.

H.2.2.3 El sistema PicSOM

PicSOM (Laaksonen et al., 1999) és un sistema de recuperació d'imatges per contingut basat en Mapes autoorganitzatius (*Self-Organizing Mapping* - SOM). Aquest és un algorisme no supervisat de clustering basat en les xarxes neuronals que projecte l'espai original de les dades a un altre més reduït on les propietats més rellevants destaquen.

En aquest cas, les imatges estan caracteritzades mitjançant el color, textura, forma, però no disposen de cap característica semàntica, paraula clau o anotació. Per cadascuna d'aquestes característiques es defineix un vector d'atributs de 15, 40 i 40 elements respectivament.

El sistema crea internament una estructura en forma d'arbre basada en l'algorisme TS-SOM (*Tree StructuredSOM*) per tal de jerarquitzar la informació. La idea d'aquest algorisme es definir diferents nivells jeràrquics a partir dels nivells de l'arbre, on per cada nivell es defineix un mapa SOM. D'aquesta manera l'usuari navega a través d'aquesta jerarquia i segons les seves preferències anirà cap a una direcció de l'arbre o altra. En aquest cas concret, l'arbre està format per 3 nivells jeràrquics de 4x4, 16x16 i 64x64. El procés de cerca és el següent:

1. L'usuari realitza una consulta amb les seves preferències.

2. A partir d'aquestes preferències, el sistema aplica aquests *inputs* en els diferents mapes SOM, i retorna els resultats de les zones més densament mapejades.

3. L'usuari marca els resultats més rellevants i els envia al sistema.

4. Amb la informació de *feedback*, el sistema mapeja les imatges indicades, per detectar noves zones o reforçar-ne d'antigues. Es retornen les imatges de les zones noves i que estan més fortament mapejades.

5. Es repeteixen els passos anteriors fins que es satisfà la cerca de l'usuari.

En els experiments realitzats s'estudia el funcionament del sistema segons les característiques que es fan servir. Estudiant el comportament del sistema, es comprova que la utilització de les característiques de color, textura i forma de manera independent proporcionen millors resultats que fent-los servir tots. Aquest resultat es produeix perquè encara que s'augmenta la dimensionalitat de les imatges, no s'amplia el seu conjunt d'exemples.

H.2.3 Més enllà de les propietats, el context de la semàntica

Les principals limitacions dels algorismes que tracten només les propietats de baix nivell són les següents: (1) la representació és sovint incomprensible degut al seu baix nivell, (2) les característiques físiques són insuficients per representar clarament un concepte, i (3) és molt difícil comparar els alts conceptes semàntics mitjançant les característiques de baix nivell indicades en

les *query*. Resumint, hi ha massa separació entre els conceptes formats per les persones i les característiques de baix nivell, de tal manera que es fan necessàries informacions complementàries per tal de permetre una comparació més acurada. A partir d'aquesta premissa, els algorismes de cerca van començar a tenir en compte a part de propietats de baix nivell, paraules o anotacions que permetessin etiquetar o dotar d'un significat més conceptual, es va dotar als elements de semàntica.

Un dels primers sistemes en tenir en compte els conceptes semàntics va ser el PicHunter (Cox et al., 2000), el qual feia servir una aproximació Bayesiana per intentar predir el que l'usuari volia segons les seves accions. Això s'aconseguia mitjançant una distribució de probabilitat sobre totes les possibles imatges enlloc de refer la *query*. A més de basar-se en propietats, feia servir anotacions per tal de millorar la precisió dels resultats.

A detall d'exemple s'explicarà el sistema iFind (Zhang et al., 2000), el qual ha estat desenvolupat pel centre de recerca xinès de Microsoft. iFind és un entorn que fa servir informació semàntica i característiques de baix nivell a l'hora de recuperar imatges. Aquest entorn proporciona cerca basada en paraules clau, per imatges i per combinació d'ambdues anteriors. Les imatges en el sistema es representen amb característiques de baix nivell, paraules clau i opcionalment anotacions. La clau de l'èxit del sistema, és la integració de tot això i la seva actualització en funció de les accions realitzades per l'usuari.

De la mateixa manera que en l'ImageRover, iFind disposa de processos encarregats de recopilar informació d'Internet. Les característiques i paraules clau s'obtenen a partir de les imatges, les capçaleres URL, els texts al voltant de les imatges, els texts associats al camp HTML 'ALT' i els links. El fet de fer servir característiques semàntiques permet disposar de més informació qualitativa a l'hora de realitzar els processos de cerca. No obstant, aquesta informació semàntica pot tenir implicacions negatives ja que aquesta informació pot ser invàlida, incorrecte i/o inconsistent.

L'aplicació de tècniques de *Relevance Feedback* en aquest cas tenen com a objectiu millorar la representació i relació de les imatges amb els conceptes. Es distingeix un model d'espai de documents i un model d'espai d'usuari. El model d'espai de documents representa la informació recopilada pel sistema sobre les característiques físiques. El model d'espai d'usuari es construeix mitjançant les accions realitzades per l'usuari durant els processos de cerca. Aquest model es construeix mitjançant probabilitats Bayesianes que relacionen la probabilitat que una imatge tingui certes paraules com a paraules claus. L'objectiu és que l'espai del model d'usuari influeixi al model d'espai de documents per minimitzar l'impacte de les inconsistències produïdes per la baixa fiabilitat en el procés d'assignació de les anotacions en les imatges.

L'entorn disposa d'una xarxa semàntica que relaciona imatges amb anotacions, una mesura de similitud que integra característiques semàntiques i d'imatge, i un algorisme d'aprenentatge artificial que iterativament actualitza la xarxa semàntica per millorar el rendiment del sistema.

La xarxa semàntica disposa de links que uneixen una imatge amb diferents conceptes. Cadascun d'aquests links disposa d'un pes que indica la importància de la paraula clau (concepte) respecte la imatge. Si l'usuari fa servir paraules clau noves, el sistema pot incorporar-les per tal d'augmentar el seu vocabulari i millorar la representació de la informació.

La mesura de la similitud es calcula mitjançant una variant de la fórmula de Rocchio explicada anteriorment, en la qual es tenen en compte les paraules clau que hi ha en comú (Equació H.4).

$$G_j = log(1+\pi_j)D_j + \beta\left(\frac{1}{N_R}\sum_{k\in N_R}\left[(1+\frac{I_1}{A_1})S_{jk}\right]\right) - \gamma\left(\frac{1}{N_N}\sum_{k\in N_N}\left[(1+\frac{I_2}{A_2})S_{jk}\right]\right) \qquad \text{(H.4)}$$

On:

- N_R i N_N representen el nombre de documents rellevants i no rellevants respectivament.

- β, γ són constants que mesuren la contribució del *feedback* respecte la *query* original.

- I_1 i I_2 són el nombre de paraules clau sense repetir en comú entre la imatge j i totes les imatges positives i negatives del *feedback* respectivament.

- A_1 i A_2 són el nombre total de paraules clau sense repetir de totes les imatges positives i negatives del *feedback* respectivament.

- S_{ij} representa la distància Euclidiana entre les característiques de baix nivell de la imatge i i j.

- $log(1+\pi_j)D_j$ és un coeficient que compensa l'ambigüitat dels texts, exposat en (Rui i Huang, 1999). El seu càlcul es realitza a partir de la diferència dels models d'espai d'usuari i de documents.

A partir d'una consulta inicial, s'aplica la mètrica de l'Equació H.4 per retornar els resultats més similars, els quals són marcats com rellevants o no rellevants per l'usuari. A partir del *feedback* enviat al sistema, aquest actualitza les relacions entre els conceptes i les imatges, i torna a buscar els més similars. Amb l'aplicació del *feedback* de l'usuari s'aconsegueix compensar les diferències entre l'espai de models d'usuari i de documents, i d'aquesta manera millora la precisió en les consultes posteriors.

Bibliografia

Aamodt, A. (1991). A knowledge-intensive approach to problem solving and sustained learning, ph.d. dissertation. University of Trondheim, Norwegian Institute of Technology.

Aamodt, A. i Plaza, E. (1994). Case-based reasoning: Foundations issues, methodological variations, and system approaches. *AI Communications*, 7:39–59.

Aha, D. i Kibler, D. (1991). Instance-based learning algorithms. *Machine Learning*, 6:37–66.

Ahluwalia, M. i Bull, L. (1999). Coevolving functions in genetic programming: Classification using k-nearest-neighbour. *In Proceedings of the genetic and evolutionary computation conference*, pages 947–952.

Althoff, K. (1989). Knowledge acquisition in the domain of cbc machine centres: the moltke approach. *EKAW-89*, pages 180–198.

ARFF (2007). Attribute relation file format. http://www.cs.waikato.ac.nz/ ml/weka/arff.html.

Armengol, E. i Plaza, E. (2000). Bottom-up induction of feature terms. *Machine Learning*, 41(1):259–294.

Armengol, E. i Plaza, E. (2006). Symbolic explanation of similarities in cbr. *Computing and informatics*, 25(2-3):153–171.

Ashley, K. (1991). Modelling legal arguments: Reasoning with cases and hypotheticals. *MIT Press*.

Asuncion, A. i Newman, D. (2007). UCI machine learning repository.

Atkeson, C., Moore, A., i Schaal, S. (1996). Locally weighted learning. *To appear in Artificial Intelligence Review*. http://www.cc.gatech.edu/fac/Chris.Atkeson/.

Bachelor, B. (1978). Pattern recognition: Ideas in practice. *New York: Plenum Press*, pages 71–72.

Bareiss, R. (1988). *PROTOS: a unified approach to concept representation, classification and learning*. PhD thesis, University of Texas at Austin, Dep. of Computer Sciences.

Basu, M. i Ho, T. (2006). *Data Complexity in Pattern Recognition*. Advanced Information and Knowledge Processing. Springer.

Bauer, H. i Villmann, T. (1997). Growing a hypercubical output space in a self-organizing feature map. *IEEE Trans. on Neural Networks*, 8(2):218–226.

Benabdeslem, K. (2006). Hybrid neural system for time series prediction. In *28th International conference on information technology interface*, pages 349–354. IMAC/IEEE.

Biberman, Y. (1994). A context similarity measure. *European Conference on Machine Learning*, pages 49–63.

Bishop, C. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

Brachman, R. i Schmolze, J. (1985). An overview of the kl-one knowledge representation system. *Cognitive Sci*, 9(2).

Bradley, A. (1997). The use of area under the ro curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159.

Branting, L. i Porter, B. (1991). Rules and precedents as complementary warrants. *Proceedings AAAI-91*.

Bridge, D. i Ferguson, A. (2002). Diverse product recommendations using an expressive language for case retrieval. In *6th European Conference on Case-Based Reasoning*, pages 43–57. Springer-Verlag.

Broomhead, D. i Lowe, D. (1988). Multi-variable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355.

Brown, M. (1994). *A Memory Model for Case Retrieval by Activation Passing*. PhD thesis, University of Manchester.

BuilderX (2007). Accelerate mission-critical c++ development. http://www.borland.com/downloads/download_cbuilderx.html.

Cabelli, S. (1988). Analogy - from a unified perspective. *In D.H. Helman (ed.), Analogical reasoning. Kluwer Academic*, pages 65–103.

Cabena, P., Hadjnian, P., Stadler, R., Verhees, J., i Zanasi, A. (1997). *Discovering Data Mining from Concept to Implementation*. Prentice Hall.

Camps, J., Garrell, J., Golobardes, E., i Vernet, D. (2003). Diseño de funciones de similitud para el razonamiento basado en casos usando programación genética: estudio con problemas sintéticos. *II Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*, pages 409–416.

Carbonell, J. (1986). A theory of reconstrucive problem solving and expertise acquisition. *Machine Learning - An Arficial Intelligence Approach*, 2:371–392.

Cardeñosa, G. (2001). *Breast Imaging Companion - Second editiond*. Lippincott Williams and Wilkins.

Carolyn, M., Gada, K., Martin, L., Keith, P., i Chris, S. (2000). An investigation of machine learning based prediction systems. *The Journal of Systems and Software*, 53:23–29.

Carpenter, G. i Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Comput. Vision Graph. Image Process.*, 37(1):54–115.

Chang, P. i Lai, C. (2005). A hybrid system combining self-organizing maps with case-based reasoning in wholesaler's new-release book forecasting. *Expert Syst. Appl.*, 29(1):183–192.

Cheeseman, P. i Stutz, J. (1996). Bayesian classification (autoclass): Theory and results. *In Advances in Knowledge Discovery and Data Mining*, 17:153–180.

Chen, Z., Wenyin, L., Hu, C., Li, M., i Zhang, H. (2001). ifind: a web image search engine. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, page 450. ACM Press.

Cordón, O. i Herrera, E. (2003). Special issue on soft computing applications to intelligent information retrieval on the internet. *International Journal of Approximate Reasoning*, 34:2–3.

Corral, G., Armengol, E., Fornells, A., i Golobardes, E. (2007). Data security analysis using unsupervised learning and explanations. In Corchado, E., Corchado, J., i Abraham, A., editors, *Innovations in Hybrid Intelligent Systems*, volume 44. Springer-Verlag. En impremta.

Corral, G., Cadenas, X., Zaballos, A., i Cadenas, M. (2005a). A distributed security system for wlans. In *1st. IEEE International Conference on Wireless Internet*.

Corral, G., Fornells, A., Golobardes, E., i Abella, J. (2006). Cohesion factors: improving the clustering capabilities of consensus. In *7th International Conference on Intelligent Data Engineering and Automated Learning*, volume 4224 of *LNCS*, pages 488–495. Springer-Verlag.

Corral, G., Golobardes, E., Andreu, O., Serra, I., Maluquer, E., i Martínez, A. (2005b). Application of clustering techniques in a network security testing system. *Artificial Intelligence Research and Devolopment*, 131:157–164.

Corral, G., Zaballos, A., Cadenas, X., i Grané, A. (2005c). A distributed vulnerability detection system for an intranet. In *Proceedings of the 39th IEEE International Carnahan Conference on Security Technology (ICCST'05)*, pages 291–295.

Corral, G., Zaballos, A., Cadenas, X., i Grané, A. (2005d). A distributed security system for an intranet. In *39th IEEE International Carnahan Conference on Security Technology*.

Cover, T. i Hart, P. (1967). Nearest neighbor pattern classification. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 13:21–27.

Cox, I., Minka, T., i Papathomas, T. (2000). The bayesian image retrieval system, pichunter: Theory, implementation, and psychophysical experiments. *IEEE Transaction on Image Processing – special issue on digital libraries*, 9:20–37.

Crestani, F. i Pasi, G. (2000). *Soft Computing in Information Retrieval*. Physica-Verlag.

Csillaghy, A., Hinterberger, H., i Benz, A. (2000). Content-based image retrieval in astronomy. In *Information Retrieval*, volume 3, pages 229–241.

Cygwin (2007). A linux environment for windows. http://www.cygwin.com/.

Dasarathy, V. (1991). Nearest neighbor (nn) norms: Nn pattern classification techniques. *CA: IEEE Computer Society Press*.

Dawkins, J. i Hale, J. (2004). A systematic approach to multi-stage network attack analysis. *Second IEEE International Information Assurance Workshop*.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.

Diaz, B. i Calero, P. (2001). A declarative similarity framework for knowledge intensive cbr. In Aha, D. i Watson, I., editors, *4th International Conference on Case-Based Reasoning*, pages 158–172. Springer-Verlag.

Diaz, B. i Calero, P. (2003). Adaptation guided retrieval based on formal concept analysis. In *5th International Conference on Case-Based Reasoning*, pages 131–145. Springer-Verlag.

Diday, E. (1974). Recent progress in distance and similarity measures in pattern recognition. *Second International Joint Conference on Pattern Recognition*, pages 534–539.

Domingos, F. (1997). Control-sensitive feature selection for lazy learners. *Artif. Intell. Rev.*, 11(1-5):227–253.

Dougherty, J., Kohavi, R., i Sahami, M. (1995). Supervised and unsupervised discretization of continuos features. *In Procedings of the Twelfth International Conference on Machine Learning*, pages 194–203.

Doyle, D., Tsymbal, A., i Cunningham, P. ("2003"). A review of explanation and explanation in case-based reasoning. In *Technical report TCD-CS-2003-41*. Department of computer Science. Trinity college, Dublin.

Dudani, S. (1976). The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics*, 6:325–327.

E. Morales (2006). Búsqueda, optimización y aprendizaje. http://dns1.mor.itesm.mx/ emorales/Cursos/Busqueda04/principal.html.

Eclipse (2007). Eclipse - an open development platform. http://www.eclispe.org.

Essam, A. i Ahmed, S. (2001). Applying neural networks in casebased reasoning adaptation for cost assessment of steel buildings. In *Proceeding of the Euro-international symposium on computational intelligence*, pages 130–137.

Evett, M. i Fernandez, T. (1998). Numeric mutation improves the discovery of numeric constants in genetic programming. In R., J., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M. H., Goldberg, D. E., Iba, H., i Riolo, R., editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 66–71, University of Wisconsin, Madison, Wisconsin, USA. Morgan Kaufmann.

Farrel, R. (1987). Intelligent case selection and presentation. In *Proceedings of the tenth International Joint Conference on Artificial Intelligence*, pages 174–176.

Fischer, E., Lehman, E., Newman, I., Raskhodnikova, S., Rubinfeld, R., i Samorodnitsky, A. (2002). Monotonicity testing over general poset domains. In *STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 474–483, New York, NY, USA. ACM Press.

Fornells, A. (2001). Tfc: Anàlisi, disseny i implementació d'anàlisi de components principals (principal component analysis - pca). Enginyeria i Arquitectura La Salle - URL.

Fornells, A., Armengol, E., i Golobardes, E. (2007a). Explanation of a clustered case memory organization. In *Artificial Intelligence Research and Development*, volume 160, pages 153–160. IOS Press.

Fornells, A., Camps, J., Golobardes, E., i Garrell, J. (2005a). Comparison of strategies based on evolutionary computation for the design of similarity functions. In *Artificial Intelligence Research and Development*, volume 131, pages 231–238. IOS Press.

Fornells, A., Camps, J., Golobardes, E., i Garrell, J. (2005b). Incorporación de conocimiento en forma de restricciones sobre algoritmos evolutivos para la búsqueda de funciones de similitud. In *IV Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*, pages 397–404. Thomson.

Fornells, A. i Golobardes, E. (2007). Case-base maintenance in an associative memory organized by a self-organizing map. In Corchado, E., Corchado, J., i Abraham, A., editors, *Innovations in Hybrid Intelligent Systems*, volume 44. Springer-Verlag. En impremta.

Fornells, A., Golobardes, E., Bernadó, E., i Martí, J. (2006a). Decision support system for breast cancer diagnosis by a meta-learning approach based on grammar evolution. In *8th International Conference on Enterprise Information Systems*, volume 233, pages 222–239. INSTICC Press.

Fornells, A., Golobardes, E., Martorell, J., Garrell, J., Bernadó, E., i Macià, N. (2007b). Measuring the applicability of self-organizing maps in a case-based reasoning system. In *3rd Iberian Conference on Pattern Recognition and Image Analysis*, volume 4478 of *LNCS*, pages 532–539. Springer-Verlag.

Fornells, A., Golobardes, E., Martorell, J., Garrell, J., Bernadó, E., i Macià, N. (2007c). A methodology for analyzing the case retrieval from a clustered case memory. In *7th International Conference on Case-Based Reasoning*, volume 4626 of *LNAI*, pages 122–136. Springer-Verlag. Nominat al premi al millor article del congrés.

Fornells, A., Golobardes, E., Martorell, J., Garrell, J., i Vilasís, X. (2007d). Management of relations between cases and patterns from som for helping experts in breast cancer diagnosis. *International Journal of Neural Systems*. En procés de revisió amb els editors.

Fornells, A., Golobardes, E., Vernet, D., i Corral, G. (2006b). Unsupervised case memory organization: Analysing computational time and soft computing capabilities. In *8th European Conference on Case-Based Reasoning*, volume 4106 of *LNAI*, pages 241–255. Springer-Verlag.

Fornells, A., Golobardes, E., Vilasís, X., i Martí, J. (2006c). Integration of strategies based on relevance feedback into a tool for retrieval of mammographic images. In *7th International Conference on Intelligent Data Engineering and Automated Learning*, volume 4224 of *LNCS*, pages 116–124. Springer-Verlag.

Fritzke, B. (1994). Growing cell structures - a self organizing network for unsupervized learning. *Neural Networks*, 7(9):1441–1460.

Fritzke, B. (1995). Growing grid - a self organizing network with constant neighborhood range and adaptation strength. *Neural Processing Letters*, 5(2):9–13.

Fritzke, B. (1996). Growing self-organizing networks, why? In *ESANN'96: European Symposium on Artificial Neural Networks*, pages 61–72.

Gamagami, P. (1996). *Atlas of mammography. New Early Signs in Breast Cancer*. Blackwell Science.

Garrell, J., Golobardes, E., Bernadó, E., i Llorà, X. (1998). Automatic classification of mammary biopsy images with machine learning techniques. In *Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems*, volume 3 of *Artificial Intelligence*, pages 411–418. ICSS Academic Press.

Gnedin, A. (2000). A note on sequential selection from permutations. *Comb. Probab. Comput.*, 9(1):13–17.

Goldberg, D. E. (1989). *Genetic Algorithm in Search, Optimization, and Machine Learning.* Addison-Wesley.

Golobardes, E. (1998). *Aportacions al raonament basat en casos per resoldre problemes de classificació.* PhD thesis, Enginyeria i Arquitectura La Salle.

Golobardes, E. i Bernadó, E. (2005). Apunts del Curs de Doctorat Aprenentatge artificial i representació del coneixement, Universitat Ramon Llull. http://www.salleurl.edu/∼ elisabet/Assignatures/AA/temari_ml.html.

Golobardes, E., Nieto, M., Salamó, M., J.Camps, Calzada, G., Martí, J., i Vernet, D. (2001). Generació de funcions de similitud mitjançant la programació genètica pel raonament basat en casos. *Proceedings of the 4th Catalonian Conference on AI*, 25:100–107.

Gutierrez, R. (2004). Course of introduction to pattern recognition. Wright State University.

Hall, R. (1989). Computational approaches to analogical reasoning; a comparative analysis. *Artificial Intelligence*, 39:39–120.

Hammond, K. (1989). Case-based planning. Academic Press.

Han, K. i Myaeng, S. (1996). Image organization and retrieval with automatically constructed feature vectors. *SIGIR Forum*, special issue:157–165.

Hanks, S. i Weld, D. (1992). Systematic adaptation for case-based planning. In *In Proceedings of 1st International Conference on AI Planning Systems*, pages 96 – 105.

Hart, P. (1968). The condensed nearest neighbor rule. *Institute of Electrical and Electronics Engineers Transactions on Information Theory*, 14:515–516.

Hartigan, J. i Wong, M. (1979). A k-means clustering algorithm. In *Applied Statistics*, pages 28:100–108.

Heath, M., Bowyer, K., Kopans, D., Moore, R., i Kegelmeyer, P. (2000). The digital database for screening mammography. *International Workshop on Digital Mammography.*

Hecht-Nielsen, R. (1987). Counterpropagation networks. *Applied Optics*, 26:4979–4984.

Heinrich, T. i Kolodner, J. (1991). The roles of adaptation in case-based design. In *In Proceedings of the AAAI Worksop on Case-based Reasoning.*

Herlocker, J., Konstan, J., Borchers, A., i Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 230–237, New York, NY, USA. ACM Press.

Hinrichs, T. (1992). Strategies for adaptation and recovery in a design problem solver. *In Hammond (ed.): Proceedings Second Workshop on case-based reasoning, Pensacola Beach, Florida, Morgan-Kauffman.*

Ho, T. i Basu, M. (2002). Complexity measures of supervised classification problems. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24(3):289–300.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press.

Hongkyu, J. i Ingoo, H. (1996). Integration of case-based forecasting, neural network, and discriminant analysis for bankruptcy prediction. *Expert Systems with Applications*, 11(4):415–422.

Jain, A. i Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(2):153–158.

Jarmulak, J., Craw, S., i Crowe, R. (2000). Genetic algorithms to optimise cbr retrieval. In *Proceedings of the 5th European Workshop on Advances in Case-Based Reasoning*, pages 136–147, London, UK. Springer-Verlag.

JDOM (2007). API JDOM homepage. http://www.jdom.org/.

Jha, G., Hui, S., i Foo, S. (1999). A hybrid case-based reasoning and neural network approach to online intelligent fault diagnosis. In *Proceeding of the third international ICSC symposia on intelligent industrial automation and soft computing*, pages 376–381.

Joachims, T. (1997). A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In Fisher, D. H., editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 143–151, Nashville, US. Morgan Kaufmann Publishers, San Francisco, US.

Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., i Wu, A. (2000). The analysis of a simple k-means clustering algorithm. In *Symposium on Computational Geometry*, pages 100–109.

Kaski, S., Honkela, T., Lagus, K., i Kohonen, T. (1998a). Websom: Self-organizing maps of document collections. *Neurocomputing*, 21(1):101–117.

Kaski, S., Kangas, J., i Kohonen, T. (1998b). Bibliography of Self-Organizing Map (SOM) Papers: 1981-1997. http://www.cis.hut.fi/research/refs/.

Keane, M. (1988). Where's the beef? the absence of pragmatic factors in pragmatic theories of analogy. *Proceedings ECAI*, pages 327–332.

Kelly, J. i Davis, L. (1991). Hybridizing the genetic algorithms and the k nearest neighbors. *Proceedings of the 4th international conference on genetic algorithms*, pages 337–383.

Kim, K. i Han, I. (2001). The cluster-indexing method for case-based reasoning using self-organizing maps and learning vector quantization for bond rating cases. *Expert Systems with Applications*, 21:147–156.

Kitano, H. (1993). Challenges for massive parallelism. *IJCAI-93, Proceedings of the Thirteenth International Conference on Artificial Intelligence, Chambery, France*, pages 813–834.

Kohavi, R. i John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69.

Kohonen, T. (1984). *Self-Organizing and Associative Memory*, volume 8 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg. 3rd ed. 1989.

Kohonen, T. (1990). The self-organizing map. *In Proceedings of the IEEE*, 78:1464–1480.

Kohonen, T. (2000). *Self-Organizing Maps*. Springer, 3rd edition.

Kolodner, J. (1983). Reconstructive memory, a computer model. *Cognitive Science*, 7:281–328.

Kolodner, J. (1993). *Case-Based Reasoning*. Morgan Kaufmann Publishers, Inc.

Kolodner, J., Simpson, R., i Sycara, K. (1985). A process model of case-based reasoning in problem solving. *IJCAI-85*.

Kopans, D. (1998). *Breast Imaging - Second editiond*. Lippincott Williams and Wilkins.

Koton, P. (1989). Using experience in learning and problem solving. ph.d. thesis. *Computer Science Dept. MIT*.

Koza, J. (1992). *Genetic Programming. Programing of computers by means of natural selection*. MIT Press.

Kuncheva, L. (1995). Editing for the k-nearest neighbours rule by a genetic algorithm. *Pattern Recognition Letters, Special Issue on Genetic Algorithms*, 16:809–814.

Laaksonen, J., Koskela, M., i Oja, E. (1999). Picsom: Self-organizing maps for content-based image retrieval. In *Proceedings of International Joint Conference on NN*.

Lebowitz, M. (1985). Categorizing numeric information for generalization. *CognitiveScience*, 9:285–308.

Lechevallier, Y., Verde, R., i de Carvalho, F. (2006). Symbolic clustering of large datasets. In *Data Science and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 193–201. Springer Berlin Heidelberg.

Lenz, M., Burkhard, H., i Brückner, S. (1996). Applying case retrieval nets to diagnostic tasks in technical domains. In *Proceedings of the Third European Workshop on Advances in Case-Based Reasoning*, pages 219–233. Springer-Verlag.

LexYacc (2007). Building parsers in c++ by means of lex and yacc. http://www.codeproject.com/cpp/lexyacc2.asp.

López, B. i Plaza, E. (1990). Case-based learning of strategic knowledge. *Machine Learning-EWSML-91*, pages 398–411.

MacArthur, S. i C.E. Brodley, C. S. (2000). Relevance feedback decision trees in content-based image retrieval. *IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 68–72.

Macià, N., Bernadó, E., Fornells, A., Golobardes, E., Martorell, J., i Garrell, J. (2007). Revisión sobre métricas de complejidad en el modelado de clústers de un sistema cbr. In *V Taller nacional de minería de datos y aprendizaje*.

Malek, M. i Amy, B. (2007). A pre-processing model for integrating cbr and prototype-based neural networks. In *Connectionism-symbolic Integration*. Erlbaum.

Martí, J., Español, J., Golobardes, E., Freixenet, J., García, R., i Salamó, M. (2000). Classification of microcalcifications in digital mammograms using case-based reasonig. *International Workshop on Digital Mammography*.

Martorell, J. (2007). *Definició d'una metodologia experimental per a l'anàlisi de resultats en sistemes d'aprenentatge artificial*. PhD thesis, Enginyeria i Arquitectura La Salle.

McCarthy, K., Reilly, J., McGinty, L., i Smyth, B. (2004). Thinking positively - explanatory feedback for conversational recommender systems. In *Proceedings of the ECCBR 2004 Workshops. TR 142-04*, pages 115–124. Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, Madrid, Spain.

McLachlan, G. (2004). *Discriminant Analysis and Statistical Pattern Recognition*. Wiley-Interscience.

McSherry, D. (2002). Diversity-conscious retrieval. In *6th European Conference on Case-Based Reasoning*, pages 219–333. Springer-Verlag.

McSherry, D. (2003). Similarity and compromise. In *5th International Conference on Case-Based Reasoning*, pages 209–305. Springer-Verlag.

McSherry, D. (2005). Explanation in recommender systems. *Artif. Intell. Rev.*, 24(2):179–197.

Michalski, R., Stepp, R., i Diday, E. (1981). A recent advance in data analysis: Clustering objects into classes characterized by conjunctive concepts. *Progress in Pattern Recognition, Vol. 1, Laveen N. Kanal and Azriel Rosenfeld (Eds.). New York: North-Holland*, pages 33–56.

Minsky, M. (1975). A framework for representing knowledge. In (ed.), P. H. W., editor, *The Psychology of Computer Vision*. McGraw-Hill, New York.

Mitchell, T. (1980). The need for biases in learning generalizations. *In J. W. Shavlik & T. G. Dietterich (Eds.), Readings in Machine Learning. San Mateo, CA: Morgan Kaufmann*, pages 184–191.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill.

Montana, D. (1993). Strongly typed genetic programming. Technical Report 7866, Cambridge, MA 02138, USA.

Mougouie, B., Richter, M., i Bergman, R. (2003). Diversity-conscious retrieval from generalized cases: A branch and bound algorithm. In *5th International Conference on Case-Based Reasoning*, pages 319–331. Springer-Verlag.

Mujica, L., Vehí, J., i Rodellar, J. (2005). A hybrid system combining self organizing maps with case based reasoning in structural assessment. In *Artificial Intelligence Research and Development*, volume 131, pages 173–180. IOS Press.

Müller, H., Müller, W., Marchand-Maillet, S., i Pun, T. (2000). Strategies for positive and negative relevance feedback in image retrieval. *International Conference on Pattern Recognition*, 1:1043–1046.

Nadler, M. i Smith, E. P. (1993). Pattern recognition engineering. *New York: Wiley*, pages 293–294.

Navinchandra, D. (1991). *Exploration and Innovation in Design*. Springer-Verlag.

Netbeans (2007). Netbeans: Java ide open source. http://www.netbeans.org/.

Nicholson, R., Bridge, D., i Wilson, N. (2006). Decision diagrams: Fast and flexible support for case retrieval and recommendation. In *8th European Conference on Case-Based Reasoning*, volume 4106 of *LNAI*, pages 136–150. Springer-Verlag.

Nieto, M. (2001). Pfc: Estudio y análisis comparativo de diferentes políticas de mantenimiento de la memoria de casos en el razonamiento basado en casos. Enginyeria i Arquitectura La Salle - URL.

Nievergelt, J. (2000). Exhaustive search, combinatorial optimization and enumeration: Exploring the potential of raw computing power. In *SOFSEM '00: Proceedings of the 27th Conference on Current Trends in Theory and Practice of Informatics*, pages 18–35, London, UK. Springer-Verlag.

Nosofsky, R. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General*, 115:39–57.

Oja, M., Kaski, S., i Kohonen, T. (2003). Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001. http://www.cis.hut.fi/research/refs/.

Oliver, A., Freixenet, J., Bosch, A., Raba, D., i Zwiggelaar, R. (2005a). Automatic classification of breast tissue. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 431–438.

Oliver, A., Freixenet, J., i Zwiggelaar, R. (2005b). Automatic classification of breast density. *IEEE International Conference on Image Processing*, 2:1258–1261.

Oltean, M. (2003). Evolving evolutionary algorithms for function optimization. In Chen, K., editor, *Proceedings of the 5th International Workshop on Frontiers in Evolutionary Algorithms*, pages 295–298, Research Triangle Park, Carolina.

O'Neill, M. i Ryan, C. (2000). Crossover in grammatical evolution: A smooth operator. In *Proceedings of the European Conference on Genetic Programming*, number 1802 in Lecture Notes in Computer Science, pages 149–162. Springer-Verlag.

openMosix (2007). Open source linux cluster project. http://openmosix.sourceforge.net/.

Pawlac, Z. (1991). *Theoretical Aspects of Reasoning about Data*. Springer.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. Morgan-Kauffman.

Pelleg, D. i Moore, A. (2000). X-means: Extending K-means with efficient estimation of the number of clusters. In *Proceedings of the 17th International Conference of Machine Learning*, pages 727–734. Morgan Kaufmann Publishers Inc.

Plaza, E. i Arcos, J. (2002). Constructive adaptation. In *6th European Conference on Case-Based Reasoning*, pages 306–320. Springer-Verlag.

Plaza, E. i López de Mantarás, R. (1990). A case-based apprentice that learns from fuzzy examples. *Methodologies for Intelligent Systems*, 5:420–427.

Porter, B. (1986). Protos: An experiment in knowledge acquisition for heuristic classification tasks. *Proceedings First International Meeting on Advances in Learning, Les Arcs, France*, pages 159–174.

Porter, B., Bareiss, R., i Holte, R. (1990). Concept learning and heuristic classification in weak theory domains. *Artificial Intelligence*, 45:229–263.

Provost, F., Fawcett, T., i Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 445–453.

Rajasekaran, S. (2000). On simulated annealing and nested annealing. *J. of Global Optimization*, 16(1):43–56.

Raymer, M., Punch, W., Goodman, E., i Kuhn, L. (1996). Genetic programming for improved data mining: An application to the biochemistry of protein interactions. *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 375–380.

Riesbeck, C. K. i Schank, R. C. (1989). *Inside Case-Based Reasoning*. Lawrence Erlbaum Associates, Cambridge, MA.

Rissland, E. i Skalak, D. (1991). Cabaret: Rule interpretation in a hybrid architecture. *International J. of Man-Machine Studies 34*, pages 839–887.

Rissland, E. L., Skalak, D. B., i Friedman, M. (1993). Case retrieval through multiple indexing and heuristic search. In *International Joint Conferences on Artificial Intelligence*, pages 902–908.

Rui, Y. i Huang, T. (1999). A novel relevance feedback technique in image retrieval. In *ACM Multimedia (2)*, pages 67–70.

Rui, Y., Huang, T., i Mehrotra, S. (1997). Content-Based image retrieval with relevance feedback in MARS. In *In Proceedings of the IEEE International Conference on Image Processing*, pages 815–818.

Rumelhart, D. i McClelland, J. (1986). Parallel distributed processing. *MIT Press*, pages 318–362.

Ryan, C., Collins, J., i O'Neill, M. (1998). Grammatical evolution: Evolving programs for an arbitrary language. In Banzhaf, W. i Poli, R., editors, *Proceedings of the First European Workshop on Genetic Programming*, volume 1391, pages 83–95. Springer-Verlag.

Salamó, M. i Golobardes, E. (2001). Analysing rough sets weighting methdos for case-based reasoning systems. In *Proceedings IX Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA'01)*, page 73.

Salamó, M. i Golobardes, E. (2002). Deleting and building sort out reduction techniques for case base maintenance. *Proceedings of the European Conference on Case-Based Reasoning*, 1:365–379.

Salamó, M. i Golobardes, E. (2003a). Hybrid deletion policies for case base maintenance. In *Proceedings of the Florida Artificial Intelligence Research Society (FLAIRS'03)*, volume 1, pages 150–155.

Salamó, M. i Golobardes, E. (2003b). Unifying weighting and case reduction methods based on rough sets to improve retrieval. *Case-Based Reasoning Research and Development*, 2689:494–508.

Salamó, M. i Golobardes, E. (2004a). Dynamic case base maintenance for a case-based reasoning system. *IX Ibero-American Conference on Artificial Intelligence Advances in Artificial Intelligence (IBERAMIA'04)*, 3315:93–103.

Salamó, M. i Golobardes, E. (2004b). Dynamic experience update for a case-based reasoning. In *Frontiers in Artificial Intelligence and Applications*, volume 109, pages 158–164.

Salamó, M. i Golobardes, E. (2004c). Global, local and mixed case base maintenance techniques. *Frontiers in Artificial Intelligence and Applications*, 113:127–134.

Salamó, M., Golobardes, E., Vernet, D., i Nieto, M. (2000). Weighting methods for a case-based classifier system. In *Proceedings of the IEEE Learning'00*.

Salzberg, S. (1991). A nearest hyperrectangle learning method. *Machine Learning*, 6:277–309.

Samuels, T. H. (1998). *Illustrated Breast Imaging Reporting and Data System BIRADS*. American College of Radiology Publications, 3rd edition.

Schaaf, J. (1995). Fish and Sink - an anytime-algorithm to retrieve adequate cases. In *Proceedings of the First International Conference on Case-Based Reasoning Research and Development*, volume 1010, pages 538–547. Springer-Verlag.

Schaffer, C. (1994). A conservation law for generalization performance. *In Proceedings of the Eleventh International Conference on Machine Learning (ML'94), Morgan Kaufmann*.

Schank, Y. (1982). Dynamic memory: A theory of learning in computers and people. *Cambridge University Press*.

Schlimmer, J. (1987). Learning and representation change. *In Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI'87)*, 2:511–535.

Sclaroff, S., Taycher, L., i LaCascia, M. (1997). Imagerover: A content-based image browser for the world wide web. Technical Report 5, Boston.

Sharma, S. i Sleeman, D. (1988). Refiner; a case-based differential diagnosis aide for knowledge acquisition and knowledge refinement. *In: EWSL 88; Proceedings of the Third European Working Session on Learning, Pitman*, pages 201–210.

Shawe-Taylor, J. i Williamson, R. (1997). A PAC analysis of a bayesian estimator. In *COLT '97: Proceedings of the tenth annual conference on Computational learning theory*, pages 2–9, New York, NY, USA. ACM Press.

Sheskin, D. (1997). *Handbook of Parametric and Nonparametric Statistical Procedures*. CRC Press.

Siedlecki, W. i Sklansky, J. (1998). Automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*.

Smith, D. (1984). Random trees and the analysis of branch and bound procedures. *J. ACM*, 31(1):163–188.

Smith, F. (1968). Pattern classifier design by linear programming. *IEEE Transactions on Computers*, 17:367–372.

Smyth, B. i Mckenna, E. (1998). An efficient and effective procedure for updating a competence model for case-based reasoners. *Proceedings of the 11th European Conference on Machine Learning*, pages 357–368.

Spencer, G. (1994). Automatic generation of programs for crawling and walking. In E. Kenneth, J. K., editor, *Advances in Genetic Programming*, pages 335–353. MIT Press.

Stanfill, C. i Waltz, D. (1986). Toward memory-based reasoning. *Communications of the ACM*, 29:1213–1228.

Steele, G. L. (1990). *Common Lisp Language 2nd Edition*. Digital Press. http://www-2.cs.cmu.edu/Groups/AI/html/cltl/cltl2.html.

Strube, G. i Janetzko, D. (1990). Episodishes wissen und fallbasierte schliessen: Aufgabe fur die wissendsdiagnostik und die wissenspsychologie. *Schweizerische Zeitschrift fur Psychologie*, pages 211–221.

Su, Y. i Kathleen, C. (2000). Using multivariate rank sum tests to evaluate effectiveness of computer applications in teaching business statistics. *Journal of Applied Statistics*, 27(3):337–341.

Suckling, J., Parker, J., i Dance, D. (1994). The mammographic image analysis society digital mammogram database. In et al., A. G., editor, *Proceedings of 2nd International Workshop on Digital Mammography*, pages 211–221.

Sycara, K. (1988). Using case-based reasoning for plan adaptation and repair. *In Kolodner (ed.): Case-Based Reasoning. Proceedings from a Workshop, Clearwater Beach, Florida, Morgan-Kauffman Publ.*

Talavera, L. i Gaudioso, E. (2001). Unsupervised discretization methods for model-based clustering. *Proceedings of the 4th Catalonian Conference on Artificial Intelligence*, pages 154–161.

Together (2007). Together technologies simplify and accelerate the success of your applications. http://www.borland.com/together/.

Tong, S. i Chang, E. (2001). Support vector machine active learning for image retrieval. *ACM Multimedia 2001, Ottawa, Canada.*

Tsz-Chiu, A., Muñoz, H., i Nau, D. (2002). On the complexity of plan adaptaion by derivational analogy in a universal classical planning framework. In *6th European Conference on Case-Based Reasoning*, pages 13–277. Springer-Verlag.

Tversky, A. (1977). Features of similarity. *Psychological Review*, 4:327–352.

Vallespí, C. (2002). Tfc: Eina d'ajuda al diagnòstic automàtic de càncer de mama utilitzant raonament basat en casos. Enginyeria i Arquitectura La Salle - URL.

Vasconcelos, N. M. i Lippman, A. B. (2000). Bayesian representations and learning mechanisms for content-based image retrieval. In Yeung, M. M., Yeo, B.-L., i Bouman, C. A., editors, *Storage and Retrieval for Media Databases 2000*, volume 3972, pages 43–54.

Veloso, M. i Carbonell, J. (1993a). Derivational analogy in prodigy: Automating case acquisition, storage, and utilization. *Machine Learning*, 3(10):249–278.

Veloso, M. i Carbonell, J. (1993b). Toward scaling up machine learning: A case study with derivational analogy in prodigy. *Machine Learning Methods for Planning*, pages 233–272.

Venkatamaran, S., Krishnan, R., i Rao, K. (1993). A rule-rule-case based system for image analysis. *First European Workshop on Case-based Reasoning, Posters and Presentations, University of Kaiserslautern*, pages 410–415.

Ventura, D. i Martinez, T. (1995). An empirical comparison of discretization methods. *In Proceedings of the Tenth International Symposium on Computer and Information Sciences*, pages 443–450.

Verbeek, J., Vlassis, N., i Kröse, B. (2005). Self-organizing mixture models. *Neurocomputing*, 63:99–123.

Vernet, D. i Golobardes, E. (2003). An unsupervised learning approach for case-based classifier systems. *Expert Update. The Specialist Group on Artificial Intelligence*, 6(2):37–42.

WebJava (2007). Java technology: The source for developers. http://www.java.sun.com.

WebJavaCC (2007). Javacc homepage. http://javacc.dev.java.net/.

WebXML (2007). Applying xml and web services standards in industry. http://www.xml.org/.

Wess, S., Althoff, K., i Derwand, G. (1994). Using k-d trees to improve the retrieval step in case-based reasoning. In *Selected papers from the First European Workshop on Topics in Case-Based Reasoning*, volume 837, pages 167–181. Springer-Verlag.

Wettschereck, D., Aha, D., i Mohri, T. (1995). A review and comparative evaluation of feature weighting methods for lazy learning algorithms. Technical report, D.C.: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence.

Whigham, P. (1995). Grammatically-based genetic programming. In Rosca, J. P., editor, *Proceedings of the Workshop on GP: From Theory to Real-World Applications*, pages 33–41, Tahoe City, California, USA.

White, R. (1992). Competitive hebbian learning: Algorithms and demonstrations. *Neural Networks*, 5(2):261–275.

Wilke, W. i Bergmann, R. (1998). Techniques and knowledge used for adaptation during case-based problem solving. In *IEA/AIE '98: Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial In telligence and Expert Systems*, volume 2, pages 497–506. Springer-Verlag.

Wilke, W., Smyth, B., i Cunningham, P. (1998). Using configuration techniques for adaptation. In *Case-Based Reasoning Technology, From Foundations to Applications*, volume 1400, pages 139–168. Springer-Verlag.

Wilson, D. i Martinez, T. (1997). Improved heterogeneus distance functions. *Journal of Artificial Intelligence Research (JAIR)*, 6:1–34.

Witten, I. i Frank, E. (2000). *DataMining: Practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann Publishers.

Witten, I. i Frank, E. (2005). *DataMining: Practical machine learning tools and techniques with Java implementations*. 2nd Edition, Morgan Kaufmann Publishers.

Wolpert, D. (1993). On overfitting avoidance as bias. Technical report, The Santa Fe Institute.

Wu, P. i Manjunath, B. (2001). Adaptive nearest neighbor search for relevance feedback in large image databases. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 89–97. ACM Press.

Wu, Y. i Huang, Q. (2000). Discriminant em algorithm with application to image retrieval. *IEEE CVPR, South California*.

Yang, Q. i Wu, J. (2001). Enhancing the effectiveness of interactive cas-based reasoning with clustering and decision forests. *Applied Intelligence*, 14(1).

Yin, H. i Allinson, N. (2001). Self-organizing mixture networks for probability density estimation. *IEEE Transactions on Neuronal Networks*, 12(2).

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8:338–353.

Zenko, B., Dzeroski, S., i Struyf, J. (2005). Learning predictive clustering rules. In *Knowledge Discovery in Inductive Databases*, volume 3933 of *Lecture Notes in Computer Science*, pages 234–250. Springer-Verlag.

Zhang, H., Wenyin, L., i Hu, C. (2000). ifind: A system for semantics and feature based image retrieval over internet. In *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, pages 477–478. ACM Press.

Zhang, H. i Zhong, D. (1995). A scheme for visual feature based image indexing. In *Storage and Retrieval for Image and Video Databases III*, volume 2420.

Zhu, H. i Yang, K. (1999). Remembering to add: Competence-preserving case-addition policies for case base maintenance. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*.

Universitat Ramon Llull

Aquesta Tesi Doctoral ha estat defensada el dia ____ d _____ de 2007

al Centre _____

de la Universitat Ramon Llull

davant el Tribunal format pels Doctors sotasignants, havent obtingut la qualificació:

President/a

_____

Vocal

_____

Vocal

_____

Vocal

_____

Secretari/ària

_____

Doctorand/a

C. Claravall, 1-3
08022 Barcelona
Tel. 936 022 200
Fax 936 022 249
E-mail: urlsc@sec.url.es
www.url.es