El bagging en casos no supervisats. Implementació a GESCONDA per algoritmes de clustering

K. Gibert[a,d], L. J. Oliva[b], I. Pinyol[c], M. Sànchez-Marrè[b,d]

[a] Dep. Estadística i Investigació Operativa, UPC

[b]Departament de Llenguatges i Sistemes Informàtics, UPC

[c]Institut d'Investigació en Intel.ligència Artificial, CSIC

[d]Knowledge Engineering and Machine Learning group, UPC

Juliol 2006

Índex

Introducció

Els algorismes de *clustering* per entorns no supervisats que es basen en una inicialització aleatòria (p. Ex.: tria inicial de llavors en l'algorisme Kmeans), presenten un problema a l'hora d'obtenir solucions fiables.

Una solució per eliminar aquest factor d'aleatorietat seria emprar altres tècniques d'inicialització. Però com es veurà posteriorment en l'article, aquestes tècniques tenen una altre problemàtica, i és la de trobar solucions òptimes locals o solucions esbiaixades.

La solució que es proposa és la utilització de la tècnica de *bagging* que s'usa en entorns supervisats, i que a través de la unió de diversos resultats de classificació respecte unes mateixes dades, permet obtenir particions òptimes.

Així mateix, es va implementar tres formes de dur a terme el *bagging* segons la forma de seleccionar la classificació de referència a partir de la qual s'uneixen la resta de classificacions. Aquestes tres tècniques són: agafant la primera classificació, triant la que presenta una major inèrcia (relació variança entre-classes i intra- classes) i triant la que aporta una major informació (mitjançant el càlcul d'Informació Mútua de Shannon).

Finalment es van provar les tècniques d'inèrcia i informació mútua amb dades ambientals reals preses d'una depuradora d'aigües residuals, per tal de comprovar l'efectivitat dels resultats respecte al mètode tradicional.

Totes les implementacions i proves es van dur a terme sobre el Sistema Intel·ligent d'Anàlisi de Dades GESCONDA, el qual es descriurà en el pròxim apartat.

L'estudi finalitza amb una breu discussió dels resultats obtinguts i unes conclusions sobre el treball realitzat.

GESCONDA

El projecte d'investigació *Desenvolupament d'un Sistema Intel·ligent d"Anàlisi de Dades per la Gestió del Coneixement en Bases de Dades Ambientals* (DB)(TIC-2004-01368) està relacionat amb la construcció d'un Sistema Intel·ligent d'Anàlisi de Dades (SIAD) per donar suport a la pressa de decisions ambientals. Aquest projecte està finançat pel govern espanyol.

GESCONDA és el nom donat al SIAD desenvolupat dins del projecte, amb l'objectiu de realitzar Descobriment de Coneixement (DC) i anàlisi intel·ligent orientat específicament a bases de dades ambientals. De fet, el DC es un pas previ i obligatori per obtenir Sistemes Intel·ligents de Presa de Decisions Ambientals de confiança. Tot i que a la literatura existeixen altres eines de DC (WEKA, Intelligent Miner,etc.) cap d'ells integra, com el GESCONDA, mètodes estadístics i de IA, la possibilitat de gestionar explícitament el coneixement produït en Bases de Coneixement (BC) (en el sentit clàssic de la IA), tècniques mixtes que poden cooperar entre si per descobrir i extreure el coneixement contingut a les dades, anàlisi dinàmic de dades, etc... en una única eina, permetent la interacció entre tots els mètodes.

Amb la base d'experiències prèvies, GESCONDA va ser dissenyat amb una arquitectura multicapa de quatre nivells connectant l'usuari amb el sistema o procés ambiental. Aquests quatre nivells son els següents [Fig. 1]: *Data Filtering, Knowledge Discovery, Knowledge Management* i *Recommendations and Meta-Knowledge Management*.

A l'hora d'afegir les funcionalitats necessàries per implementar la tècnica de bagging es va utilitzar el mòdul *Clustering Techniques Agent*, de la capa *Knowledge Discovery*. A l'Annex 1 i 2 es troben per una banda les modificacions i funcionalitats fetes al sistema, i les classes modificades.

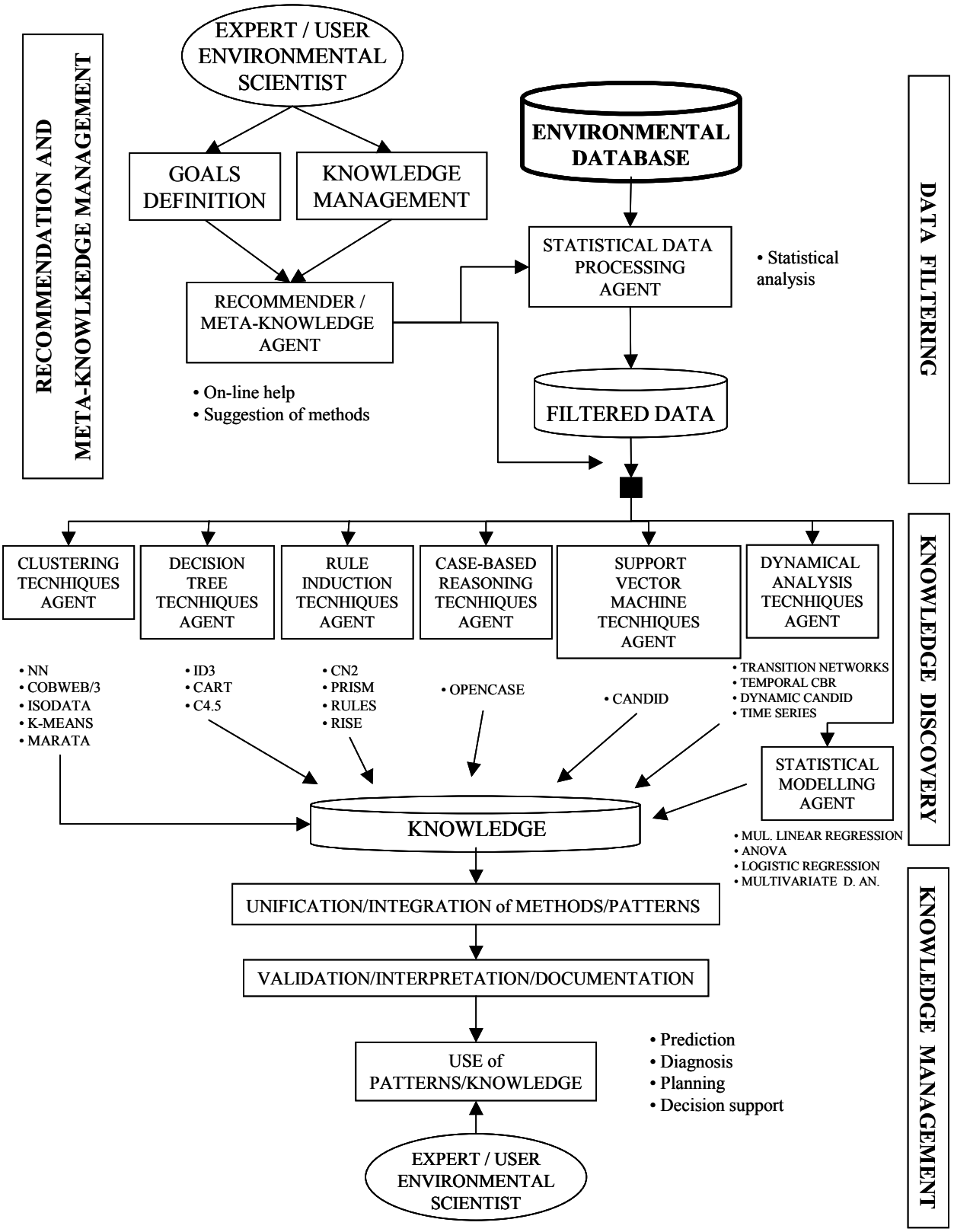


Fig. 1: Arquitectura de GESCONDA. Font: [x]

Algorismes de *clustering*

A continuació es descriuen els algorismes de *clustering* que es varen estudiar i emprar per aplicar les tècniques de *bagging*. Aquests algorismes van ser: el KMeans i el Nearest-Neighbour. També es descriu la tècnica de *bagging* i les diferents formes d'escollir la classificació de referència per tal d'agrupar les diferents classificacions obtingudes.

KMeans

L'algorisme KMeans és potser l'algorisme de clustering no supervisat més conegut i utilitzat en mineria de dades per la seva senzilla i intuïtiva estructura, així com l'eficiència en termes temporals i els bons resultats que acostuma a donar en certs entorns.

A part del conjunt d'instàncies rep com a entrada el número de classes que es volen obtenir i el increment mínim (que s'utilitza com a llindar d'aturada de l'algorisme). En algunes versions es substitueix aquest increment mínim per un nombre màxim d'iteracions.

Algorisme K-Means

Entrada: E: Conjunt d'instàncies
K: Enter (Nombre de classes)
e: Real (Increment mínim)

Sortida: P: Conjunt de prototips amb les corresponents instàncies

1.- Generar k prototips amb k instàncies escollides aleatòriament
2.- Assignar el conjunt d'instàncies al prototip més proper
3.- S1 := Suma del quadrat de les distàncies instància - prototips
4.- **fer**
5.- Recalcular prototips
6.- Reassignar les instàncies al prototip més proper
7.- S2: = S1
8.- S1:= Suma del quadrat de les distàncies instància - prototips
9.- **mentre** S1-S2 <= e

Fi Algorisme

Nearest Neighbours

Aquest algorisme, a diferència del K-Means, no té com a entrada el nombre de classes sinó que té un radi que especifica la distància màxima que hi pot haver entre una instància i el seu prototip. Així, el nombre de classes es converteix en variable de sortida.

Algorisme Nearest-Neighbour

Entrada: E: Conjunt d'instàncies
radi: Enter

Sortida: P: Conjunt de prototips amb les corresponents instàncies

1.- Generar 1 prototip amb una instància aleatòria de l'entrada
2.- **mentre** quedin instàncies sense assignar a cap prototip fer
3.- e = Escollir una instància no assignada
4.- d= Calcular la distància entre e i el prototip més proper p
5.- **si** d <= radi llavors
6.- Assignar la instancia e al prototip p
7.- Recalcular el prototip p
8.- **sinó**
9.- Generar un nou prototip amb la instància e
10.- **fi si**
11.- **fi mentre**

Fi Algorisme

Tècniques de Bagging

El bagging és una tècnica procedent de l'aprenentatge supervisat on es combinen els resultats obtinguts en diferents algorismes per a generar-ne una de consensuada. Com és ben sabut, existeixen una multitud d'algorismes d'aprenentatge supervisat i d'inducció de regles que donen maneres de classificar instàncies. Aquestes regles, en la majoria de casos, són diferents depenent de l'algorisme utilitzat, i conseqüentment, les classificacions que es realitzin podran ser també diferent. Així, per solucionar aquest biaix, s'utilitza el bagging, on en la fase

d'aprenentatge es contrasten les diferents classificacions que s'han obtingut d'una instància i s'assigna a la classe més freqüent.
En el present treball s'aplica aquesta idea als algorismes de clustering no supervisats. De la mateixa manera, existeixen diferents algorismes que particionen els conjunts d'instàncies de formes diferents. Fins i tot algorismes com el K-Means o el Nearest-Neighbour donen particions diferents al ser executats més d'una vegada amb exactament les mateixes entrades.

Especialment en aquest últim cas és on creiem que utilitzar tècniques de pseudobagging a diferents execucions del K-Means sobre el mateix conjunt de dades pot donar molt més bons resultats que quedar-se únicament amb la primera classificació obtinguda. Per aquest factor d'aleatorietat, una classificació pot ser bona o dolenta. Sense l'ajuda de cap expert no es pot saber quina és millor que l'altre. Ara bé, amb diferents execucions, per aquest mateix factor d'aleatorietat, algunes classificacions seran bones i d'altres dolentes. Combinant-les utilitzant bagging podrem obtenir classificacions més o menys consensuades que ens asseguraran un equilibri entre les bones i les dolentes, disminuint així el risc d'obtenir classificacions dolentes.

El pseudobagging que hem implementat està restringit a classificacions que s'hagin realitzat emprant prototips de classe, i segueix el següent esquema:

Sigui E = (E_1,...$E_{n)}$ el conjunt d'instàncies formades per k atributs E_i = (E_{i1},...E_{ik}).Sigui C=(C_1...C_r) el conjunt de classificacions obtingudes després d'executar r algorismes de clustering sobre les instàncies, totes elles amb exactament el mateix número de classes. L'algorisme és el següent:

Algorisme Bagging

Entrada: E: Conjunt d'instàncies
C: Conjunt de classificacions
Sortida: P: Conjunt de prototips amb les corresponents instàncies

1.- C_a = Seleccionar classificació de referència
2.- (C'_1,...,C'_r) = trobar correspondències de classes amb C_a
3.- **Per cada** instància *e* ∈ E
4.- Classificar *e* segons les classificacions (C'_1,...,C'_r)
5.- Assignar a *e* la classificació mes freqüent
6.- **Fi per cada**

Fi Algorisme

Els passos 1 i 2 són exclusius del bagging per algorismes no supervisats. Cada classificació, encara que utilitzi el mateix número de classes, pot tenir nomenclatura diferent, o, encara que s'utilitzi la mateixa, per exemple utilitzant enters {1..ξ} on ξ és el número de classes, la classe 1 d'una classificació pot no tenir res a veure amb la classe 1 d'una altre. Així, s'han de buscar les correspondències.

Per a fer-ho, primer es selecciona una classificació que ens servirà de referència, que serà la base per a calcular les correspondències per a cada classe. La idea és que la classificació de referència C_a quedarà com a definitiva, i per cada classe *t* d'una altra classificació C_f s'assignarà a la classe *t´* de C_a que més s'assembli. És a dir, el fet que *t* s'assigni a *t'* implicarà que el prototip de la classe *t* tindrà distància mínima amb el prototip de la classe *t* `. El criteri de distància hauria de ser el mateix que s'hagi utilitzat per a produir les classificacions C.

Així, el punt crític és com seleccionar la classificació de referència. Per fer-ho, hem optat per tres mètodes diferents: Primer, Inèrcia i Informació Mútua.

Primer

Aquest mètode simplement selecciona la primera classificació que s'obté d'entrada, és a dir, C_1. La problemàtica d'aquest mètode és que una classificació dolenta a C_1 pot donar classificacions dolentes al final del procés de bagging.

Inèrcia

El que anomenem inèrcia és al quocient entre la variança entre-classe (S^2_ξ) i la variança intra-classes (S^2_p), és a dir, com de separats estan les classes i com de compactes es cada una de les classes, respectivament. És lògic pensar que ens interessa una variança entre classes alta, i una variança intra classes baixa. Així, es pot utilitzar el càlcul de la inèrcia per seleccionar la classificació de referència amb l'índex d'inèrcia superior. Els càlculs matemàtics es descriuen a continuació. (Cal dir que aquest càlcul només es pot realitzar amb variables continues.)

Per calcula la variança intra-classe, es necessita calcular el centroide de la classe (1):

$$\bar{x}_{cj} = \frac{\sum_{\forall i \in c} x_{ij}}{n_c} \qquad (1)$$

On n_c és el número de centroides de la classe C i $\bar{x}_c = (\bar{x}_{c1}, \ldots, \bar{x}_{ck})$, és el seu centroide.

Igualment, per la variança entre-classes, cal tenir calculat per una banda el centroide de totes les instàncies (2):

$$\bar{x}_j = \frac{\sum_{i=1}^{n} x_{ij}}{n} \qquad (2)$$

On $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_k)$ és el centroide de totes les instàncies i n és el número d'instàncies.

I per una altre banda, és necessari calcular la distància euclidiana entre dos punts pertanyents a un clúster (3):

$$d(x_i, x_j)^2 = \sum_{z=1}^{k} (x_{iz} - x_{jz})^2 \qquad (3)$$

On k és el número d'atributs i $x_i = (x_{i1}, \ldots, x_{ik})$ és el centroide de l'element i-èssim.

Variança intra-classe

La variança intra-classe es calcula segons la següent equació (4):

$$S_p^2 = \frac{\sum_{\forall c} (n_c - 1) S_c^2}{n - \xi} \qquad (4)$$

On ξ és el número de classes de la partició i n és el número d'instàncies.

I on S_c^2 respon a la següent formula (5):

$$S_c^2 = \frac{\sum_{\forall i \in c} d(x_i, \bar{x}_c)^2}{n_c - 1} \qquad (5)$$

On ξ és el número de classes de la partició i n_c és el número de centroides de la classe C.

Variança entre-classe

La variança entre-classe té la següent forma (6):

$$S_\xi^2 = \frac{\sum_{\forall c} d(\bar{x}_c, \bar{x})^2}{n - \xi} \qquad (6)$$

On ξ és el número de classes de la partició i n és el número d'instàncies.

Índex d'inèrcia a maximitzar

L'índex d'inèrcia que emprarem per comparar quina classificació és millor respecte de la seva inèrcia respon a la següent forma (7):

$$F = \frac{S_\xi^2}{S_p^2} \qquad (7)$$

Informació mútua

Aquest mètode es basa en el càlcul d'informació mútua de Shanon, és a dir, en seleccionar la classificació amb el valor d'entropia més elevat. Aquest mètode doncs, només es pot utilitzar per variables categòriques, i per tant, qualsevol variable continua que aparegui s'haurà de discretitzar.

La formula d'informació mútua respon a la següent forma (8):

$$I(X,Y) = \sum_{x_i} \sum_{y_i} \Pr(x_i, y_i) \log \frac{\Pr(x_i, y_i)}{\Pr(x_i)\Pr(y_i)} \qquad (8)$$

L'equació (8) es composa de tres probabilitats:

$$\Pr(x_i, y_i) = \frac{n^o \, aparicions \; X = x_i \; i \; Y = y_i \; simultànies}{n^o \, observacions \; útils \; en \; X \; i \; Y \; útils \; tots \; dos} \qquad (9)$$

$$\Pr(x_i) = \frac{n^o \, aparicions \; x_i}{n^o \, observacions \; útils \; en \; X} \qquad (10)$$

$$\Pr(y_i) = \frac{n^o \, aparicions \; y_i}{n^o \, observacions \; útils \; en \; Y} \qquad (11)$$

Índex d'informació mútua a maximitzar

En concret aplicarem diferents atributs X_i contra la variable de classificació Y, per veure fins a quin punt és de bona una classificació respecte de les altres. L'índex que utilitzarem és la mateixa però per un conjunt d'atributs (12):

$$I(X_1,...,X_k,Y) = \sum_{x_1}...\sum_{x_k}\sum_{y_i} \Pr(x_1,...,x_k,y_i)\log\frac{\Pr(x_1,...,x_k,y_i)}{\Pr(x_i,...,x_k)\Pr(y_i)} \quad (12)$$

Objectius de l'estudi

L'objectiu del treball és estudiar l'efecte de la utilització de la tècnica de bagging descrita anteriorment per a l'anàlisi de dades en sistemes reals complexos. Com ja hem comentat, algorismes com el KMeans o el Nearest-Neighbour tenen una component d'aleatorietat a l'hora de seleccionar les llavors inicials que poden donar lloc a classificacions no desitjades.

En el cas simple d'instàncies de dues variables amb les quals volem dues classes ens podem trobar amb el següent [Fig 2]:

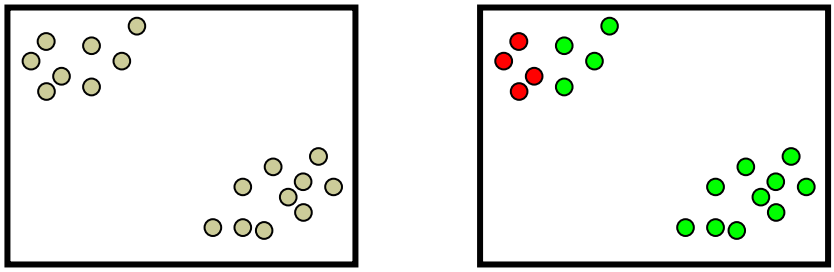


***Fig 2**: Classificació obtinguda al aplicar KMeans a unes dades on la selecció de les llavors inicials a generat una mala classificació*

Quan lo desitjable seria [Fig 3]:

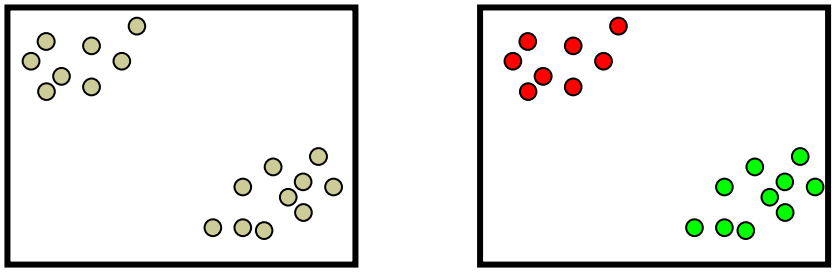


***Fig 3**: Classificació desitjada*

Ara bé, en entorn no supervisats no existeix una informació de referència que permeti comprovar la bondat d'una classificació. Així, per contrastar les classificacions obtingudes utilitzares els estadístics descrits anteriorment: Inèrcia i Informació mútua.

Cas d'estudi

Les dades amb les que s'han realitzat els experiments s'han extret d'una planta depuradora d'aigües residuals, on a diari s'han pres mesures que caracteritzen bàsicament el caudal d'entrada, l'estat de la mescla després del primer decantador, el caudal de sortida i l'estat de la mescla en el reactor biològic. A continuació es detallen les variables que s'han utilitzar per a l'anàlisi:

1. **Variables d'entrada**

   - Q-E: Caudal d'entrada (metres cúbics d'aigua per dia)
   - FE-E: Pretractament amb ferro(mg de ferro per litre d'aigua)
   - PH-E: pH (unitats de pH)
   - SS-E: Sòlids en suspensió (mg de sòlids per litre d'aigua)
   - SSV-E: Sòlids volàtils en suspensió (mg de sòlids per litre d'aigua)
   - DQO-E: Fracció de matèria orgànica degradable per acció d'agents químics oxidants sota condicions d'acidesa (mg d'òxid per litre d'aigua)
   - DBO-E: Fracció de matèria orgànica biodegradable en aigua residual (mg d'oxigen per litre d'aigua

2. **Variables després de la decantació**

   - PH-D: pH (unitats de pH)
   - SS-D: Sòlids en suspensió (mg de sòlids per litre d'aigua)
   - SSV-D: Sòlids volàtils en suspensió (mg de sòlids per litre d'aigua)
   - DQO-D: Fracció de matèria orgànica degradable per acció d'agents químics oxidants sota condicions d'acidesa (mg d'òxid per litre d'aigua)

- DBO-D: Fracció de matèria orgànica biodegradable en aigua residual (mg d'oxigen per litre d'aigua

3. Variables de sortida

- PH-S: pH (unitats de pH)
- SS-S: Sòlids en suspensió (mg de sòlids per litre d'aigua)
- SSV-S: Sòlids volàtils en suspensió (mg de sòlids per litre d'aigua)
- DQO-S: Fracció de matèria orgànica degradable per acció d'agents químics oxidants sota condicions d'acidesa (mg d'òxid per litre d'aigua)
- DBO-S: Fracció de matèria orgànica biodegradable en aigua residual (mg d'oxigen per litre d'aigua

4. Variables de tractament biològic

- V30-B: Anàlisis volumètric 30; Qualitat de sedimentació de la mescla(ml per litre d'aigua)
- MLSS-B: Sòlids en suspensió del licor mescla (mg de sòlids per litre d'aigua)
- MLVSS-B: Sòlids volàtils en suspensió del licor mescla (mg de sòlids per litre d'aigua)
- MCRT-B: Edat cel·lular (dies)
- QB-B: Caudal del reactor biològic (metres cúbics d'aigua per dia)

5. Altres Variables

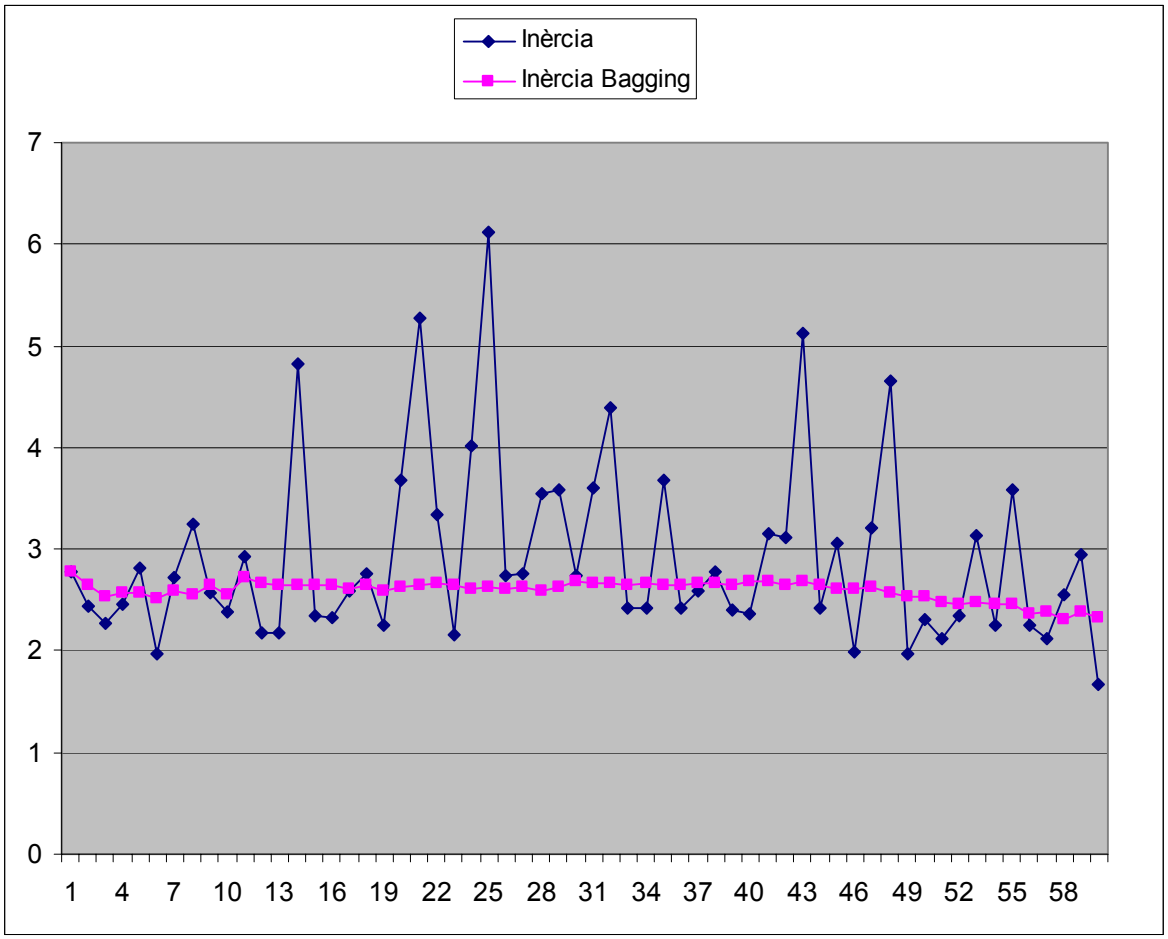
- QR-G: Caudal de recirculació (metres cúbics d'aigua per dia)
- QP-G: Caudal de la purga (metres cúbics d'aigua per dia)
- QA-G: Afluència d'aire (metres cúbics d'aire per dia)

Fase experimental

L'experiment ha consistit en l'execució de l'algorisme Kmeans 60 cops, sobre un conjunt de dades (descrites en l'anterior apartat) de 396 instàncies. Cada execució s'ha realitzat amb els mateixos paràmetres: partició en 4 classes, amb un increment mínim del 0.05 i una llavor inicial de 1150713012125.

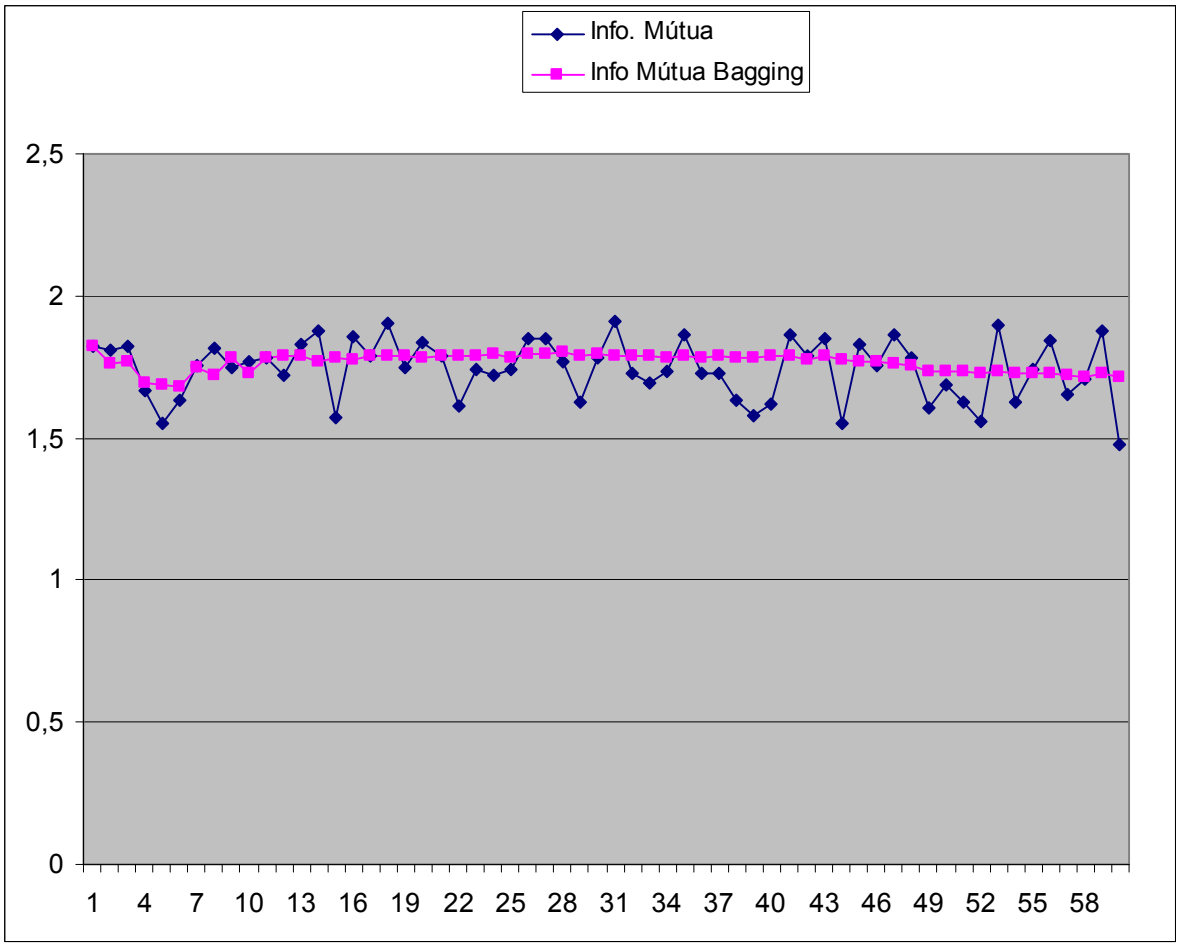
Posteriorment s'ha aplicat *bagging* de forma seqüencial: primer amb dues classificacions, després amb tres, i així successivament fins arribar a les 60 execucions. Per cada execució de Kmeans s'han aplicat les fórmules d'inèrcia i informació mútua; també s'han aplicat per cada execució incremental de *bagging*.

A continuació es mostra la gràfica de les inèrcies de les classificacions obtingudes:



Gràfica 1: Inèrcia d'execucions KMeans i de *bagging* incremental. **Font**: Autor.

I també per les execucions amb informació mútua:



Gràfica 2: Informació mútua d'execucions KMeans i de *bagging* incremental. **Font**: Autor.

Discussió de resultats

Com s'observa en les gràfiques, el gràfic dels estadístics en les 60 execucions mostren un alt grau de variabilitat amb execucions que varien de 3.75 a 1.5 en la inèrcia i entre 1.25 i 1.80 en la informació mútua de forma completament aleatòria i sense cap estabilitat.

Al fer *bagging* s'observa com a mesura que s'afegeixen classificacions, tan els valors de la Inèrcia i la Informació Mútua s'estabilitzen aproximadament a un terme mig de la seva variabilitat.

També s'observa que la tècnica de Informació mútua sembla que s'estabilitza més aviat que no pas la Inèrcia, tot i que les diferències no són força significatives.

Conclusions i treball futur

En aquest treball s'ha vist com la utilització de tècniques de bagging sobre classificacions generades a partir d'algorismes aleatoris no supervisats ajuda a estabilitzar la quantitat de informació de les noves classificacions. És ben sabut que en aquests entorns no supervisats és difícil conèixer la bondat de les classificacions. No obstant, el combinar diferents solucions generades amb certs graus d'aleatorietat acostuma a ser garantia que les noves solucions no seran tan dolentes com les que es podrien obtindre, però al mateix temps no seran tan bones. El que si garantirem utilitzant tècniques de bagging serà l'estabilitat en quant a quantia d'informació, trencant així l'absoluta incertesa d'una única execució.

ANNEX 1: Noves Utilitats a GESCONDA

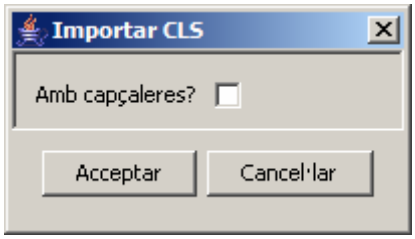
Exportar/Importar CLS

Menú → Arxiu → Exportar CLS...

Exporta la classificació seleccionada al camp 'Partició' a un fitxer amb format .cls . Si la opció 'Id. Per defecte' està activada utilitza com a identificador d'instància l'atribut seleccionat al camp 'Identificador'. Si no està seleccionada, utilitzarà un identificador seqüencial començant per i_1 fins a i_n.

La opció *Salvar les capçaleres*, serveix per guardar el nom de la columna seleccionada a *Partició*, juntament amb els valors de classificació.

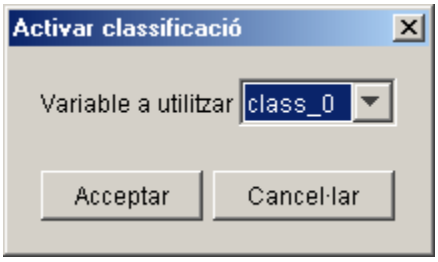
Menú → Arxiu → Importar CLS...



Importa un fitxer .cls, tot indicant si el fitxer conté una capçalera, i per tant el programa ha de d'agafar-la i emprar-la per denominar la columna que crearà, o bé no conté capçalera i el programa crearà amb la classificació una columna anomenada *class*.

Activar Classificació

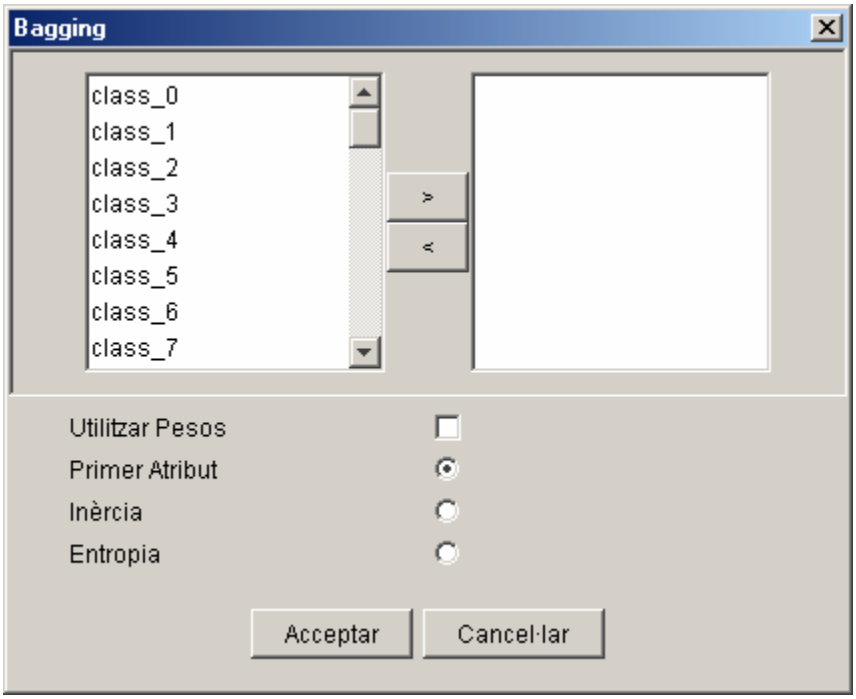
Menú → Validació → Activar Classificació



Recupera la classificació seleccionada al camp 'Variable a utilitzar'

Bagging

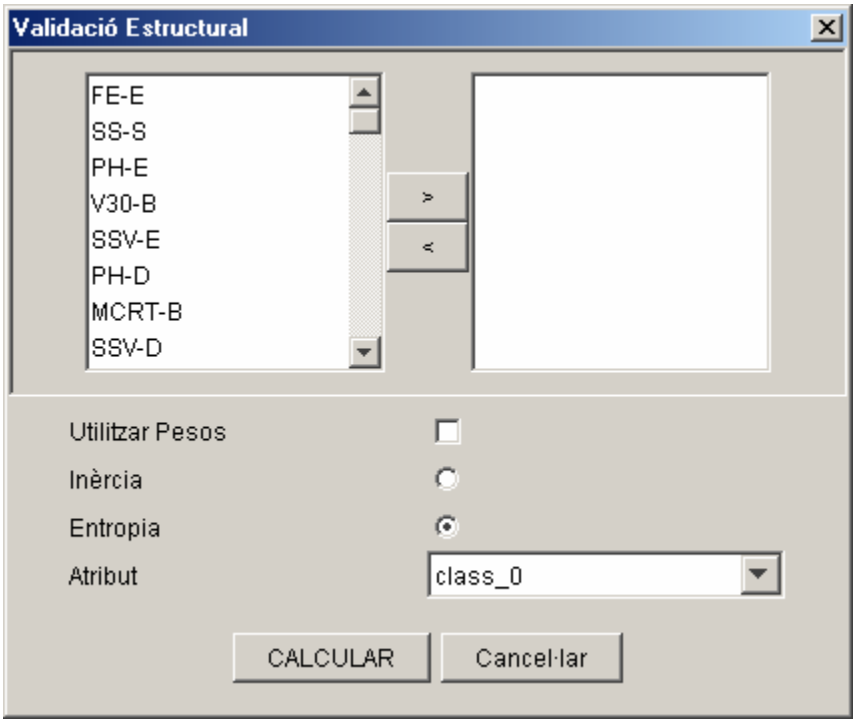
Menú → Algorismes → Bagging



Permet realitzar el bagging de les classificacions seleccionades al llistat dret. Permet escollir el mètode de selecció de la classificació de referència (Primer, Inèrcia, Entropia). Si la opció 'Utilitzar Pesos' està desactiva aquests no s'utilitzen per al càlcul de la Inèrcia.

Validació Estructural

Menú → Validació → Validació Estructural



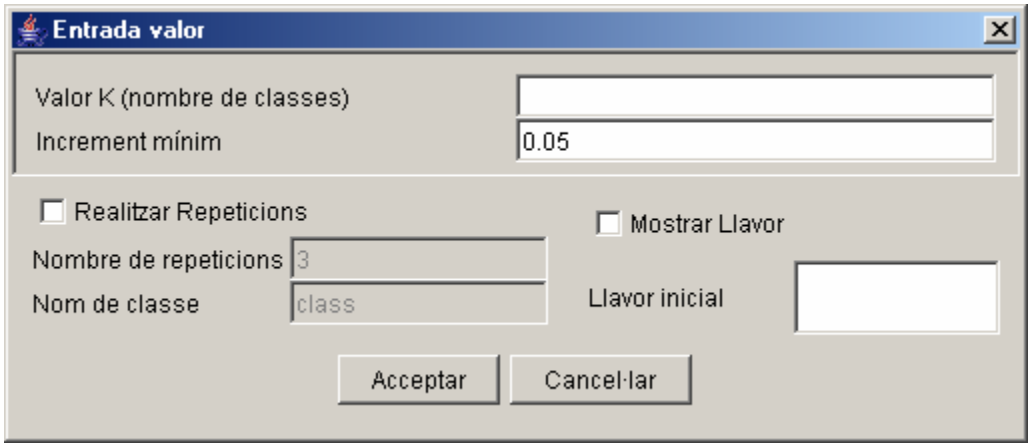
Permet calcular el valor de la Inèrcia o Entropia (segons selecció) de la classificació seleccionada al camp 'Atribut' sobre els camps seleccionats a la llista dreta. En cas de la Inèrcia, els camps seleccionats han de ser continus i s'utilitzaran els pesos d'aquests segons el camp 'Utilitzar Pesos'. En cas de l'entropia, tots els atributs continus es discretitzaran.

Randomització de l'algorisme Nearest-Neigbhour

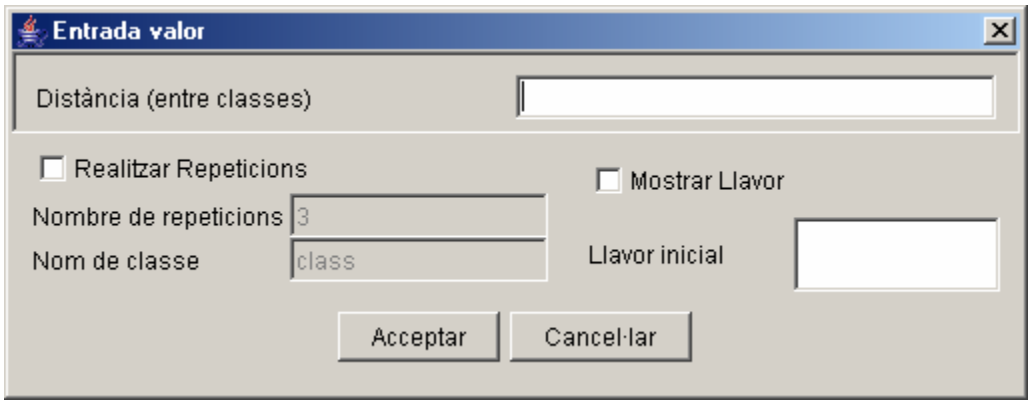
Fins ara, aquest algorisme utilitzava com a primer element a tractar la primera instància. Ara el primer element s'escull aleatòriament, convertint-se doncs en un algorisme no determinista.

Tractament de llavors aleatòries

Menú → Algorismes → Kmeans



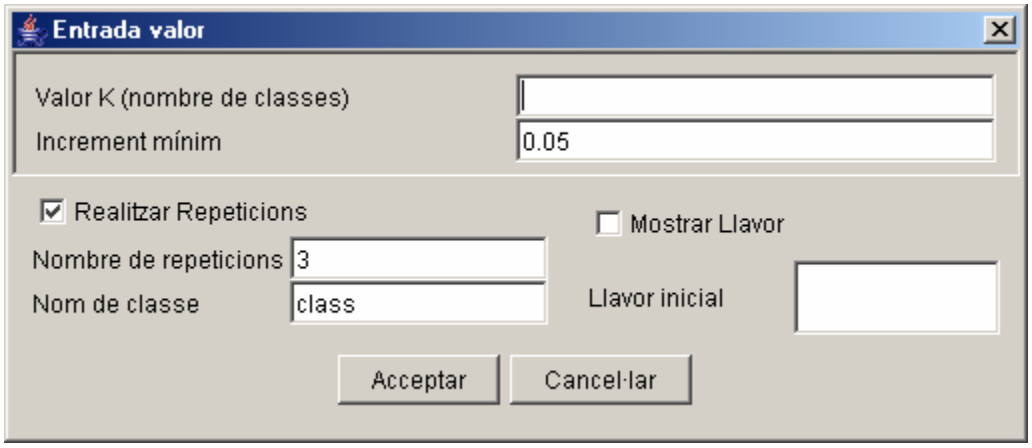
Menú → Algorismes → Nearest-Neighbour



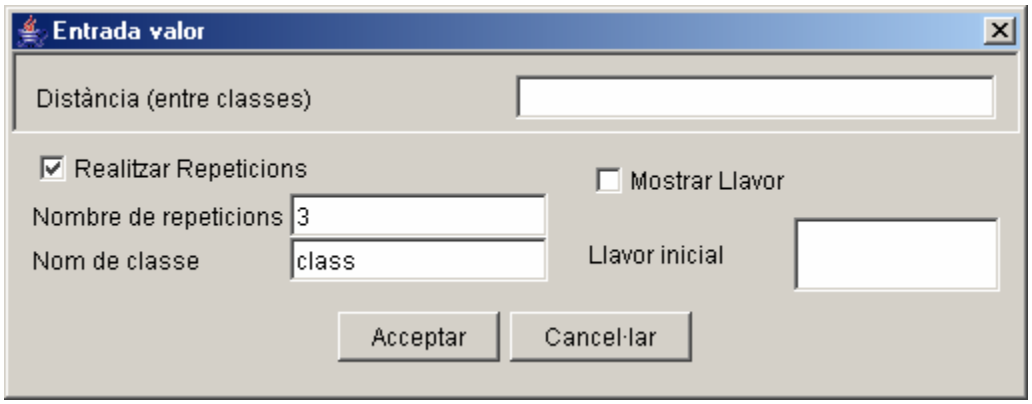
Es pot obtenir la llavor utilitzada per cadascun dels algorismes marcant la opció 'Mostrar Llavor'. Posteriorment aquesta es pot reutilitzar omplint el camp 'Llavor Inicial'. El valor de la llavor es mostra en quadre de diàleg mentre s'executa l'algorisme.

Execucions múltiples automàtiques de KMeans i Nearest-Neighbour

Menú → Algorismes → Kmeans



Menú → Algorismes → Nearest-Neighbour



Si la opció 'Realitzar Repeticions' està marcada es permet escollir el nombre de repeticions que s'executà l'algorisme. Les respectives classificacions s'afegiran com a columnes en les dades amb un nom assignat seqüencialment a partir del nom de la classe escrit.

Directori de treball

A l'hora d'obrir o guardar fitxers .gps, importar o exportar fitxers .cls, el sistema empra com a directori estàndard de treball l'indicat a un fitxer anomenat .directoryPath.

En aquest fitxer s'emmagatzema l'últim directori que ha emprat l'usuari a l'hora d'obrir/guardar fitxers o exportar/importar classificacions. Si el fitxer no existeix, el crea, i si per defecte utilitza el directori './dades', que en cas de no existir agafa el directori per defecte del sistema operatiu.

Luís Javier Oliva Felipe
Isaac Pinyol Catadau

ANNEX 2: Classes canviades

Diagrama de classes mòdul d'anàlisis

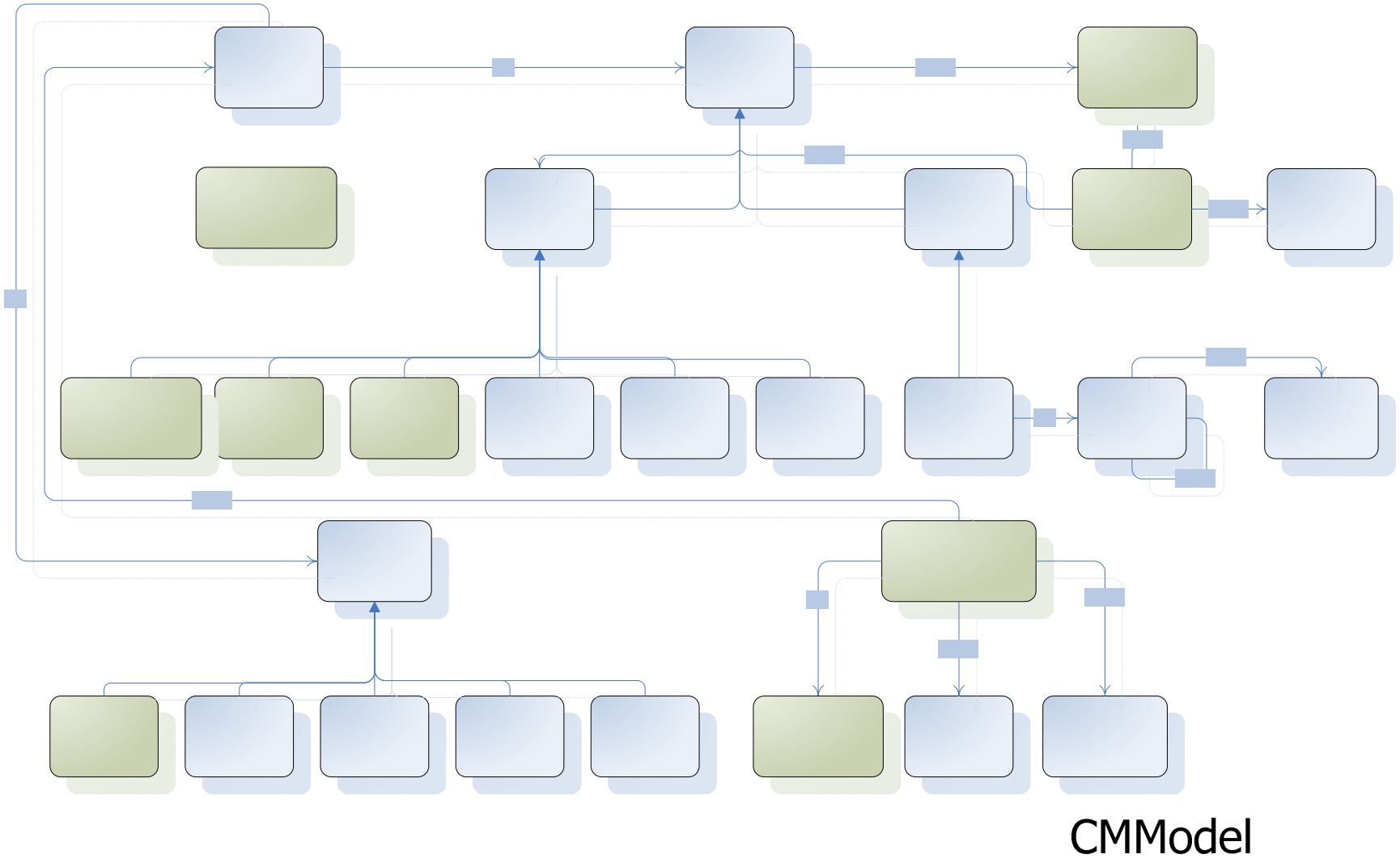


Diagrama de classes mòdul visualització

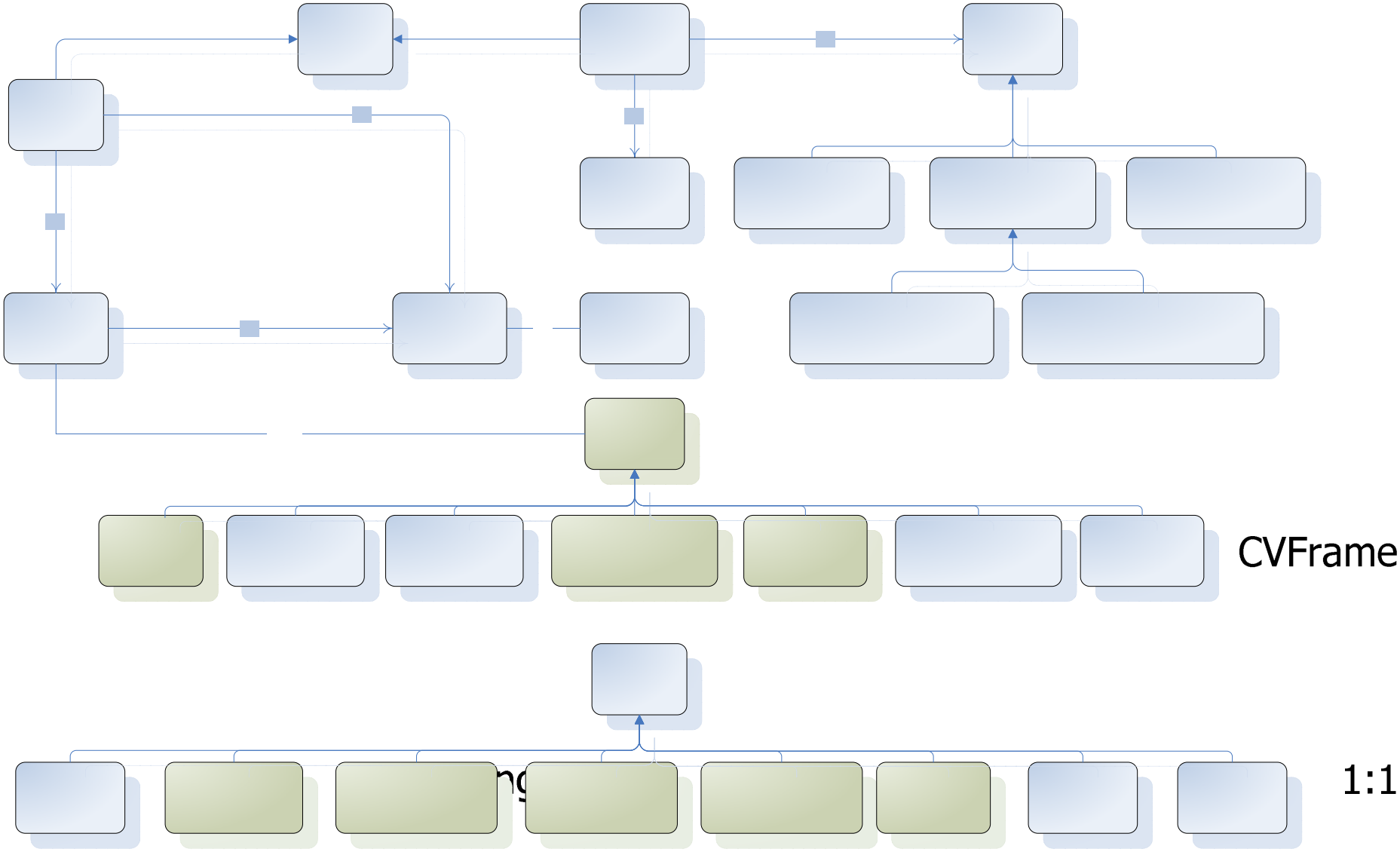
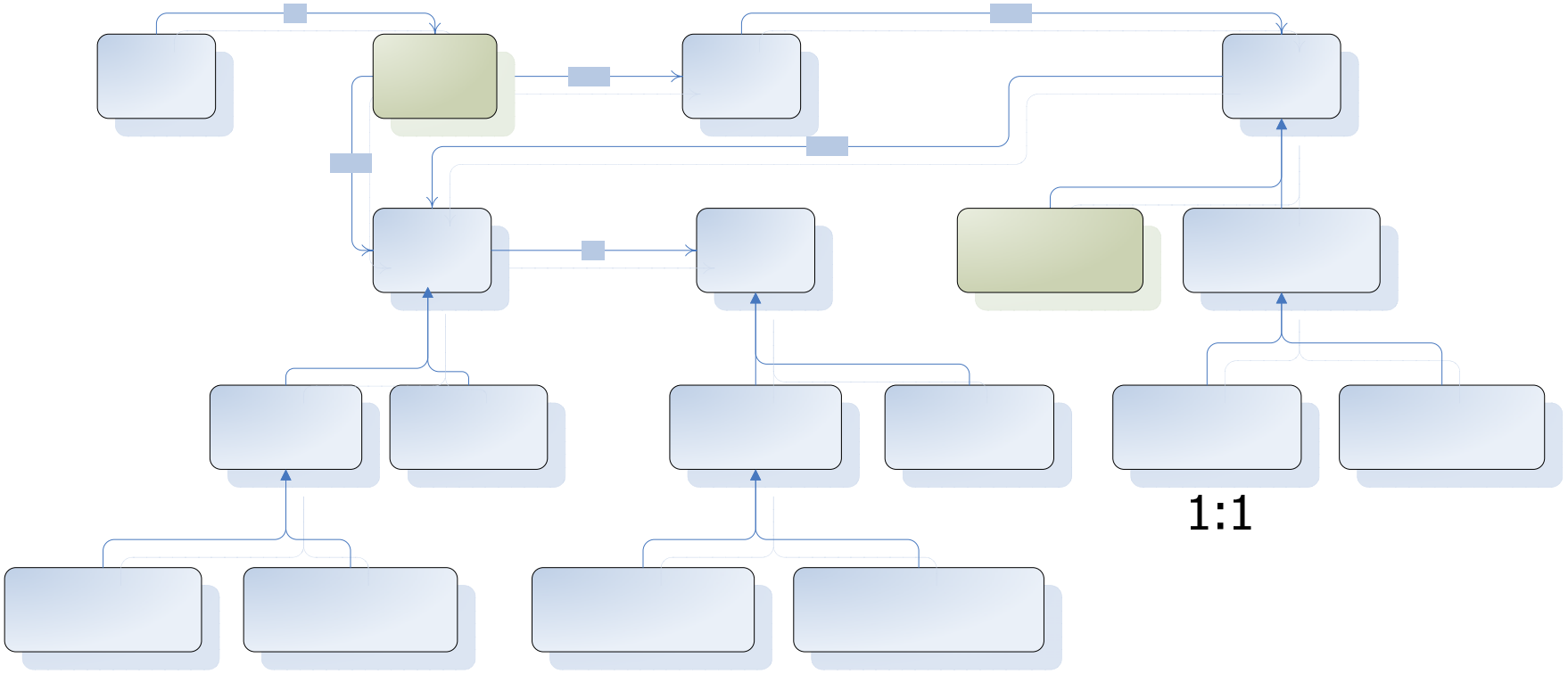


Diagrama classes mòdul dades



Referències

[1] Huh, M.Y. (2006) "Incremental Subset Selection for Complex Data", International Workshop on Visualization and Variable Selection

[2] Bühlmann, P (2003) "Bagging, subagging and bragging for improving some prediction algorithms". Recent Advances and Trends in Nonparametric Statistics (Eds. Akritas, M.G., Politis, D.N.), Elsevier

[3] Dudoit, S., Fridlyand, J (2003) "Bagging to improve the accuracy of a clustering procedure", *Bioinformatics*, Vol. 19 No 9, pp. 1090-1099

[4] Bauer, E., Kohavi, R. (1999) "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants", Machine *Learning*, Vol. 36 (1/2), pp. 105-139