



Modélisation et apprentissage des préférences par réseaux de neurones pour l'aide à la décision multicritère

Kary Främling

► **To cite this version:**

Kary Främling. Modélisation et apprentissage des préférences par réseaux de neurones pour l'aide à la décision multicritère. Sciences de l'environnement. INSA de Lyon, 1996. Français. <NNT : 1996ISAL0025>. <tel-00825854>

HAL Id: tel-00825854

<https://tel.archives-ouvertes.fr/tel-00825854>

Submitted on 24 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

présentée

DEVANT L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON

pour obtenir

LE GRADE DE DOCTEUR

FORMATION DOCTORALE: INGÉNIERIE INFORMATIQUE

PAR

Kary FRÄMLING

(Ingénieur en Informatique - Diplomingenjör i Datateknik)

Modélisation et apprentissage des préférences par réseaux de neurones pour l'aide à la décision multicritère

Soutenue le 15 Mars 1996 devant la commission d'Examen

Messieurs	Michel INSTALLÉ	Rapporteur
	Roman SLOWINSKI	Rapporteur
	Markku SYRJÄNEN	Examineur
	Bernard AMY	Examineur
	Joël FAVREL	Examineur
	Philippe VINCKE	Examineur
	Philippe BEAUNE	Examineur
	Didier GRAILLOT	Examineur
	Albert MATHON	Directeur de thèse

À mes parents

À mon épouse Marie et à mon fils Niels

Remerciements

Je remercie tout le monde ...

Cette petite phrase était le seul contenu de cette page pendant plus d'un an, écrite juste pour éviter une page vide, mais tellement vraie dans sa simplicité. Chacune des personnes que j'ai connues pendant ce travail de thèse a contribué à rendre mon séjour en France encore plus riche et intéressant que je ne l'aurais songé au moment de prendre la grande décision de quitter mon pays natal pour quelques années (à voir combien ...). Il est malheureusement impossible de remercier tous les anciens collègues et amis qui sont partis et ceux qui sont arrivés pour faire des thèses, des stages, etc. Je tiens malgré tout à remercier quelques personnes en particulier.

Didier GRAILLOT, Directeur de Recherche à l'École Nationale Supérieure des Mines de Saint-Etienne et Directeur adjoint du centre SIMADE (Sciences Informatiques, Modélisation, Aide à la Décision et Environnement), est grandement remercié pour les nombreuses discussions qui ont été importantes pour la réussite de cette thèse. Je le remercie également pour l'initiation à d'autres activités moins scientifiques, qui ont été bénéfiques pour mon intégration dans l'équipe et pour augmenter ma motivation au travail.

Je tiens également à remercier Albert MATHON, Professeur à l'École Nationale Supérieure des Mines de Saint-Etienne, pour m'avoir accueilli dans son département et dans son équipe doctorale de l'INSA de Lyon.

Mes plus vifs remerciements vont également à Michel INSTALLÉ, Professeur à l'Université Catholique de Louvain en Belgique, de m'avoir accueilli pour un stage de trois mois dans le Laboratoire d'Automatique, Dynamique et Analyse des Systèmes. Ces trois mois ont été particulièrement efficaces et intéressants grâce à la bonne ambiance de travail à Louvain-la-Neuve. Les contacts obtenus avec d'autres personnes travaillant dans le domaine d'aide à la décision multicritère, notamment le Professeur Roman SLOWINSKI et le Professeur Philippe VINCKE, ont été particulièrement bénéfiques pour ce travail de thèse. Je les remercie pour l'intérêt qu'ils y ont porté.

Je tiens à exprimer ma gratitude à l'égard du Professeur Markku SYRJÄNEN de l'Université de Technologie à Helsinki (Finlande), que j'ai vu régulièrement pendant mes séjours en Finlande pour discuter de ce travail de thèse et lui donner des nouvelles de la France.

D'autres personnes avec qui j'ai eu des discussions très intéressantes et que je remercie sont le Docteur Philippe BEAUNE pour les discussions souvent philosophiques mais utiles et le Docteur Bernard AMY pour son invitation à présenter mon travail à son équipe à l'INP de Grenoble. Ces discussions et ces contacts ont été particulièrement bénéfiques par la valeur des commentaires et des questions, pointues et variées, véritables détecteurs de « trous de raisonnement ». J'ai également apprécié mes rencontres avec le Professeur Joël FAVREL.

Je remercie toutes ces personnes d'avoir également bien voulu participer au Jury de thèse.

Sophie PERREARD, qui effectue une thèse sur les explications pédagogiques, m'a beaucoup simplifié le travail bibliographique sur les explications et je la remercie pour les discussions fructueuses que nous avons échangées.

Enfin, un grand merci à tous les « supporters » qui ont contribué à maintenir mon moral au plus haut. Il y a bien sûr mon épouse Marie que j'ai rencontré pendant ce séjour à Saint-Etienne et qui m'a encouragé tout particulièrement pour la rédaction de ce manuscrit, la partie la plus dure ayant été allégée par sa bonne humeur et sa compréhension. Mes amis George CRAWLEY, Nathalie BAINIER et Gilles DURANT, avec qui de nombreux projets ont vu le jour, à chaque fois pour le plus grand bien de chacun. Je ne peux non plus oublier mes parents, qui ont su garder leur moral pendant mon séjour en France.

Pour finir, je remercie tous les membres de SIMADE (et tout particulièrement de l'ancien département STRAD) pour le bon accueil et la bonne ambiance. Je souhaite bon courage aux thésards n'ayant pas encore franchi l'étape de rédaction et de soutenance.

Saint-Étienne, Février 1996.

Résumé

La modélisation des préférences des décideurs pour des problèmes de choix est l'objet principal de ce travail de thèse. Afin d'obtenir des modèles plus réalistes, des fonctions de préférence contextuelle sont présentées dans ce mémoire de thèse. La complexité d'expression de telles fonctions est réduite par le recours à des techniques d'apprentissage automatique à partir de réseaux de neurones.

L'identification des fonctions de préférence est réalisée par des réseaux de neurones, fondés sur les principes de la régression non-linéaire, à partir d'exemples de décision. Une telle technique n'est pas toujours applicable en pratique à cause du nombre élevé d'exemples nécessaire. Le réseau neuronal INKA, développé dans le cadre de ce travail, effectue la régression avec un nombre d'exemples relativement petit. INKA offre également des temps d'apprentissage courts par rapport à d'autres techniques, ce qui est nécessaire pour une acquisition interactive de la fonction de préférence.

INKA est utilisé dans le système interactif d'aide à la décision (SIAD) présenté, qui est un des premiers à mettre en œuvre un apprentissage automatique d'une fonction de préférence globale. La visualisation de la fonction apprise et les indicateurs de précision et de sensibilité permettent au décideur d'estimer le moment d'arrêter la recherche de solutions. Ceci est particulièrement utile pour apprendre les préférences des décideurs « abstraits » (acteurs sociaux, systèmes naturels, ...), qui ne peuvent pas utiliser directement le système.

Les fonctionnalités d'explication développées dans ce travail permettent de justifier les recommandations des SIAD, ce qui a constitué un des grands défis du domaine de l'aide à la décision multicritère et des réseaux neuronaux. Il est donc possible d'expliquer, de comprendre et d'analyser les préférences même pour des décideurs abstraits. L'intérêt de telles explications est de faciliter la prise de décisions négociées dans le cadre de projets d'aménagement complexes ou pour améliorer des produits dont les ventes dépendent des préférences des clients.

Mots clés

Aide décision, réseau neuronal, justification, système expert, décision collective, préférence, apprentissage.

Title

Learning and Explaining Preferences with Neural Networks for Multiple Criteria Decision Making

Abstract

The goal of this Ph.D. work is to improve multiple criteria decision making by the use of « context-dependent » preference models. Such models are more realistic than the ones used before, but it is usually difficult for the decider to express them. This is the reason for using machine learning by neural networks for identifying the preference model.

A neural net « learns » the preference model by the principles of non-linear regression, where the decision model is expressed by examples of decisions. The INKA neural network developed reduces the number of examples necessary, which is essential for the practical use of this technique. Learning times are also sufficiently short to make possible the interactive acquisition of the preference model.

The interactive decision support system developed using INKA is one of the first to use machine learning to identify a global preference model. The visualisation of the learnt model and the indicators of precision and sensibility developed help the decider to decide when to stop the interactive procedure. This is especially important for learning the preferences of « abstract deciders » (a group of people, consumers, nature, ...), who can not interact with the system.

Explaining the results is still a great problem both for decision support systems and for neural networks. The methods developed here make it possible to reduce or eliminate this problem. It is therefore possible to explain, understand and analyse even the preferences of « abstract deciders ». This information may then be used for improving group decision making or for improving product sales which depend on consumer preferences.

Keywords

Decision aid, neural network, justifications, expert system, social decision, preference, learning.

Sommaire

INTRODUCTION GENERALE	19
PARTIE I: APPRENTISSAGE DES FONCTIONS DE PREFERENCE PAR RESEAUX DE NEURONES DANS UN SYSTEME MULTICRITERE D'AIDE A LA DECISION	25
1. LES SYSTEMES D'AIDE A LA DECISION	26
1.1 ÉLÉMENTS TAXINOMIQUES	27
1.2 L'ÉTAT DE L'ART DES SYSTEMES D'AIDE A LA DECISION	29
1.3 FONCTIONS DE PREFERENCE CONTEXTUELLE	31
2. FONCTIONS DE PREFERENCE CONTEXTUELLE PAR COMBINAISON D'ANALYSE MULTICRITERE ET D'UNE BASE DE REGLES (STOCDECH)	32
2.1 ANALYSE MULTICRITERE PAR SYSTEMES EXPERTS	33
2.2 COMBINAISON DE METHODES D'ANALYSE MULTICRITERE ET D'UNE BASE DE REGLES	34
2.3 LA LOGIQUE FLOUE COMME OUTIL D'EXPRESSION DES FONCTIONS DE PREFERENCE CONTEXTUELLE	38
3. APPRENTISSAGE AUTOMATIQUE DES FONCTIONS DE PREFERENCE	39
3.1 METHODES D'AIDE A LA DECISION MULTICRITERE UTILISANT L'APPRENTISSAGE SYMBOLIQUE AUTOMATIQUE	41
3.2 LES RESEAUX DE NEURONES	43
3.2.1 Le neurone artificiel	45
3.2.2 Réseau neuronal	46
3.2.3 Apprentissage automatique par réseaux de neurones	48
4. DESCRIPTION DU RESEAU NEURONAL DEVELOPPE (INKA)	49
4.1 CARACTERISTIQUES TECHNIQUES DE INKA	50
4.2 ALGORITHME D'APPRENTISSAGE	53
5. APPRENTISSAGE D'UNE FONCTION DE PREFERENCE EN TROIS DIMENSIONS	56
6. SELECTION ET SAISIE D'EXEMPLES D'APPRENTISSAGE	59
6.1 SELECTION DES EXEMPLES D'APPRENTISSAGE PAR CARTE AUTO-ORGANISATRICE DE KOHONEN	60
6.1.1 Modèle de connectivité	60

Chapitre 1: Les systèmes d'aide à la décision

6.1.2 Apprentissage par compétition	61
6.1.3 Conservation de la topologie	62
6.2 SAISIE DE PREFERENCES	65
6.2.1 Analyse de sensibilité et de robustesse	67
6.2.2 Réduction du nombre de critères de sélection	69
7. EXEMPLE D'APPLICATION: CHOIX DE VOITURE	70
7.1 CRITERES DE SELECTION	70
7.2 MODELE DE PREFERENCE	72
7.3 SELECTION ET SAISIE DES EXEMPLES D'APPRENTISSAGE	72
7.4 ÉVALUATION DES RESULTATS OBTENUS PAR INKA	74
8. CONCLUSION	78
PARTIE II: METHODOLOGIE INTERACTIVE D'AIDE A LA DECISION UTILISANT DES RESEAUX DE NEURONES	81
9. INTRODUCTION AUX SYSTEMES INTERACTIFS D'AIDE A LA DECISION	82
9.1 METHODES INTERACTIVES DE RECHERCHE DE LA SOLUTION	84
9.2 METHODES POUR L'IDENTIFICATION INTERACTIVE D'UN MODELE DE PREFERENCE	85
10. SIAD UTILISANT L'APPRENTISSAGE AUTOMATIQUE PAR RESEAUX DE NEURONES	86
10.1 DETERMINATION DU NOMBRE D'EXEMPLES OPTIMAL A L'AIDE DE MODELES COUT / BENEFICE	88
10.2 DETERMINATION DU NOMBRE OPTIMAL D'EXEMPLES DE PREFERENCE POUR L'APPRENTISSAGE NEURONAL	89
11. ANALYSE DE LA FONCTION DE PREFERENCE APPRISE	90
11.1 ANALYSE GRAPHIQUE DE LA FONCTION DE PREFERENCE	91
11.2 TRAITEMENT DES COEFFICIENTS DE CERTITUDE	94
11.2.1 Coefficients de certitude dans INKA	94
11.2.2 Analyse graphique des coefficients de certitude de INKA	96
11.2.3 Coefficients de certitude d'un ensemble d'actions	97
12. CORRECTION DE LA FONCTION DE PREFERENCE PAR PSEUDO-EXEMPLES	97
12.1 PSEUDO-EXEMPLES	98
12.1.1 Saisie de pseudo-exemples	100
12.1.2 Transformations des valeurs de préférence	100
12.2 APPLICATION AU PROBLEME DU CHOIX DE VOITURE	101
13. INDICATEURS DE PRECISION	106
13.1 VALIDATION DES RESULTATS D'APPRENTISSAGE POUR UN FAIBLE NOMBRE D'EXEMPLES	107
13.2 CALCUL DE LA DIFFERENCE ENTRE DEUX CLASSEMENTS	109

Chapitre 1: Les systèmes d'aide à la décision

19.4 EXPLICATIONS PAR SIMILARITE	152
19.5 NIVEAU DE CREDIBILITE DANS LES EXPLICATIONS	153
20. CONSTRUCTION D'UNE EXPLICATION	154
20.1 PARAMETRES DETERMINANT LE CONTENU DES EXPLICATIONS	155
20.2 EXPLICATIONS DE TYPE « BASE DE REGLES »	157
20.3 EXPLICATIONS DE TYPE « ANALYSE MULTICRITERE »	158
20.4 EXPLICATIONS « MIXTES »	160
21. EXPLICATIONS POUR L'APPLICATION « CHOIX DE VOITURE »	162
21.1 EXPLICATIONS POUR DES FONCTIONS DE PREFERENCE CONTEXTUELLE	162
21.2 NOMBRE D'EXEMPLES D'APPRENTISSAGE NECESSAIRE	165
21.3 EXPLICATIONS PAR SIMILARITE	168
22. MODELES CONCEPTUELS POUR UN DIALOGUE EXPLICATIF	170
22.1 RESEAUX SEMANTIQUES	171
22.1.1 Diagnostic et explication par réseau sémantique	172
22.1.2 Diagnostic par réseau neuronal et explication par un réseau sémantique associé	173
22.2 DIALOGUES	176
22.2.1 Types de questions	176
22.2.2 Modèles d'utilisateur	178
23. PERSPECTIVES: DES EXPLICATIONS POUR LES DECISIONS NEGOCIEES	179
23.1 OBJECTIFS DU PROJET	180
23.2 METHODOLOGIE PROPOSEE	181
23.3 IMPLEMENTATION INFORMATIQUE	183
23.3.1 Rôle d'ADELAIDE dans la méthodologie SPREAD	183
23.3.2 Architecture informatique	184
23.3.3 Bases de données	185
23.3.4 Acquisition des paramètres des méthodes d'aide à la décision	185
23.3.5 Présentation et analyse des résultats	186
24. CONCLUSION	186
CONCLUSION GENERALE	189
REFERENCES BIBLIOGRAPHIQUES	193
ANNEXE A: PRESENTATION DETAILLEE DES APPLICATIONS UTILISEES	211

ANNEXE B: THE INTERPOLATING, NORMALISING AND KERNEL ALLOCATING NETWORK FOR FAST FUNCTION APPROXIMATION	229
ANNEXE C: PRESENTATION DE L'OUTIL STOCDECH	253
ANNEXE D: TABLEAUX DES DONNEES UTILISEES ET DES RESULTATS DES DIFFERENTES METHODES	269
LISTE DES SYMBOLES	287
LISTE DES FIGURES	291
GLOSSAIRE	295
INDEX	303

Introduction générale

Notre monde qui devient toujours plus complexe pose parfois des problèmes qui dépassent la capacité intellectuelle de l'être humain à prendre de bonnes décisions. Ces problèmes peuvent apparaître dans des systèmes créés par

Chapitre 1: Les systèmes d'aide à la décision

l'homme lui-même comme dans les systèmes de production. Les problèmes les plus difficiles se posent néanmoins dans les systèmes naturels, qui ne sont ni créés ni vraiment contrôlables par les êtres humains. Cette insuffisance de l'être humain et la nécessité de prendre de plus en plus de décisions dans ces domaines demandent le développement d'outils qui peuvent aider le décideur à comprendre ces systèmes et à prendre des décisions les concernant. C'est la raison pour laquelle ce travail s'intéresse aux systèmes d'aide à la décision multicritère.

Aide à la décision - mais quel type de décisions?

L'aide à la décision est un concept très large, qui inclut des méthodes assez variées comme la simulation, la recherche opérationnelle, l'analyse coût / bénéfice, ... Le choix de la méthode la plus appropriée dépend bien évidemment du problème à résoudre. S'il s'agit de la planification de la production d'une usine, il est peu probable que l'on utilisera les mêmes méthodes que pour le choix d'une automobile.

Les méthodes *d'aide* à la décision ne remplacent jamais le décideur. Ce sont des méthodes destinées à aider le décideur pour structurer, interpréter et comprendre l'information quand la quantité ou la complexité de celle-ci dépasse sa capacité décisionnelle. Le décideur ne doit jamais oublier sa part de responsabilité dans l'analyse des résultats de ces méthodes.

Parmi les méthodes d'aide à la décision on fait souvent la distinction entre *l'aide à la décision continue* et *l'aide à la décision discrète* [Pomerol & Barba-Romero, 1993]. L'aide à la décision continue consiste à trouver des valeurs d'entrée qui optimisent une valeur de sortie (valeur *objective*) en respectant des contraintes explicites. Un exemple de ce type de méthodes est la programmation linéaire.

Dans ce travail de thèse seront traitées les méthodes d'aide à la décision discrètes parce que les solutions possibles sont prédéfinies et qu'il s'agit d'identifier parmi ces solutions celle ou celles qui sont les meilleures. Les contraintes sont, dans ce cas, plutôt implicites qu'explicites car elles sont définies par l'ensemble des solutions possibles.

La notion *multicritère* est souvent ajoutée au concept d'aide à la décision et exprime le fait que de multiples entrées peuvent avoir une influence sur la décision. Dans le contexte d'aide à la décision discrète on appelle ces entrées des *critères de sélection*.

Dans le cadre de l'aide à la décision multicritère, il est le plus souvent faux de dire que l'on cherche à trouver une solution optimale et unique. Il est fort possible qu'une telle solution existe en théorie, mais cette solution n'est pratiquement jamais présente dans notre ensemble de solutions possibles. Il sera ainsi nécessaire de trouver *le meilleur compromis* en prenant en compte tous les critères de sélection. Ce meilleur compromis n'est pas unique, mais il change en fonction des *préférences du décideur*, pour qui certains critères sont plus importants que d'autres.

Même pour un problème aussi simple que celui du choix d'une voiture, le problème devient vite complexe s'il faut en choisir une qui convienne à tous les membres d'une famille. Il ne suffit plus de prendre en compte les préférences d'une seule personne, mais celles de plusieurs. Nous nous trouvons alors dans une situation de discussion entre les décideurs, l'objectif consistant à trouver une *action consensuelle*. Les *systèmes d'aide à la décision à décideur multiple* SADDM¹ sont souvent les outils le plus adaptés à de telles situations de décision.

Pour des problèmes d'une échelle encore plus grande, il devient difficile d'utiliser des SADDM. Trouver le meilleur compromis entre tous les décideurs dans le cadre d'un problème de choix de sites de stockage de déchets industriels, par exemple, est très difficile à cause du grand nombre d'acteurs impliqués. Dans ce processus de décision, les décideurs sont les autorités légalement compétentes pour prendre la décision et la mettre en œuvre. Ceux qui sont susceptibles de subir les conséquences des décisions prises sont également d'autres acteurs. Interviennent aussi un certain nombre d'acteurs plus abstraits, comme la nature, l'économie, l'industrie, le développement technologique, les associations variées, etc., dont les préférences sont mal connues et difficiles à estimer au moment de la prise de décision.

Les expériences de cas réels ont clairement montré qu'il est nécessaire de prendre en compte les préférences d'un nombre d'acteurs plus grand que dans le passé pour prendre de tels types

¹ En anglais: *multiple decider decision making systems*, MDDM.

de décision. Sinon, dans un système démocratique, il devient impossible de les mettre en œuvre. La prise en compte des préférences d'un tel nombre de décideurs est néanmoins très difficile avec les SAD existants. Ce sont souvent des préférences uniquement issues de résultats de sondages ou d'enquêtes, de données statistiques, etc. Le problème de choix des sites de stockage de déchets industriels ultimes dans la région Rhône-Alpes en France permet de bien illustrer ce type de problèmes. Ce travail s'intéresse ainsi à la modélisation et à l'identification des préférences de tels acteurs. Les modèles de préférence identifiés peuvent ensuite faciliter le processus de négociation, qui peut se faire à l'aide d'un SADDM.

Contribution de ce travail au domaine de l'aide à la décision multicritère

Le domaine de l'aide à la décision multicritère est très vaste et il existe une grande gamme de méthodes différentes, utilisant des pondérations, des règles, des coefficients de certitude, ... Malheureusement, ces méthodes présentent des inconvénients qui les rendent difficilement applicables en pratique. Elles sont d'habitude soit trop simples pour permettre une expression réaliste des préférences, soit trop compliquées pour qu'elles soient facilement compréhensibles et utilisables par les décideurs. Le SAD StocDech développé au cours de ce travail illustre certaines difficultés de ces méthodes.

L'inconvénient de la plupart de ces méthodes est qu'elles ne permettent pas de traiter les préférences contextuelles des décideurs. Pour le faire, il faut que les importances des critères de sélection et l'utilité de leurs valeurs puisse changer en fonction de valeurs des critères. Ceci nécessite d'utiliser de nouveaux modèles de préférence. Le modèle retenu est celui de réseaux de neurones, car il permet de traiter de telles préférences et de les identifier à partir d'exemples sur le comportement du décideur.

Le premier objectif est donc de montrer comment apprendre le modèle de préférence à l'aide d'un réseau de neurones et sous quelles conditions il est possible de le faire. Les conditions concernent avant tout le nombre et le type d'exemples nécessaires. Les méthodes de sélection et de saisie d'exemples et le réseau neuronal développés pour diminuer ce nombre sont présentés dans la première partie. Les performances de ce Système d'Aide à la Décision (SAD) neuronal sont illustrées par une application de choix de voiture.

La deuxième partie présente l'extension de ce SAD neuronal vers un Système Interactif d'Aide à la Décision (SIAD) neuronal, qui permet au décideur de modifier ses préférences au cours du processus de décision. Pour pouvoir le faire, il faut lui offrir les moyens de comprendre et d'évaluer le modèle de préférence appris et les résultats du SIAD. Il faut également être facile pour le décideur de modifier ce modèle de préférence. La visualisation du modèle de préférence offerte par l'approche neuronale rend ceci possible.

Tout SIAD doit permettre au décideur de continuer la recherche de sa solution aussi longtemps qu'il l'estime nécessaire. Dans la plupart de cas, sa satisfaction est un compromis entre le coût pour continuer la recherche et le risque de se tromper, qui correspond au point d'arrêt de la recherche. L'évaluation de ce point d'arrêt peut être difficile si le décideur n'est pas informé par des indicateurs de coût et de risque. Une grande partie de la deuxième partie se concentre sur ce type d'indicateurs.

La troisième partie de ce mémoire de thèse traite de l'explication des résultats des SAD. Plus les problèmes à traiter deviennent difficiles à résoudre, plus la pertinence des explications devient importante. Malheureusement, cet aspect est encore très mal résolu dans les SAD existants. C'est également le cas des réseaux de neurones, qui présentent des grandes difficultés pour expliquer leurs résultats. Un réseau de neurones est donc a priori un mauvais choix comme modèle de préférence. Le premier objectif de la troisième partie est ainsi de présenter des méthodes d'explication des résultats des SAD et des réseaux de neurones. Un deuxième objectif consiste à utiliser les explications pour mieux comprendre les préférences apprises par le réseau neuronal.

Pour illustrer les fonctionnalités et les avantages de la méthodologie présentée dans le cadre de ce travail, trois problèmes de grandeur réelle sont traités. Les résultats obtenus montrent l'intérêt de cette méthodologie et les possibilités de l'utiliser dans d'autres applications.

Plan de la thèse

Ce mémoire de thèse se compose de trois parties principales comme l'indique le schéma ci-dessous. La première partie traite de l'utilisation de l'approche neuronale en tant que *système d'aide à la décision*, (SAD), sans aspect interactif. Elle débute par une introduction aux

Chapitre 1: Les systèmes d'aide à la décision

méthodes « traditionnelles » d'aide à la décision, et permet d'identifier le type de problèmes à résoudre. Les principes généraux de l'utilisation d'un réseau neuronal en tant que SAD sont ensuite expliqués, en rappelant les principes de base des réseaux neuronaux et du réseau neuronal INKA en particulier. Cette première partie s'achève par une application concernant le choix d'automobile pour montrer comment l'apprentissage des préférences d'un décideur est réalisé à partir d'un ensemble d'exemples prédéfini et d'un SAD neuronal.

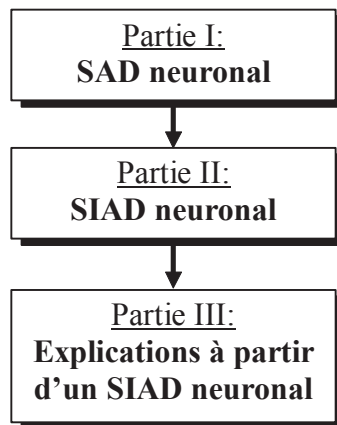


Schéma du plan.

La deuxième partie traite de l'intégration de l'approche neuronale dans un *système interactif d'aide à la décision* (SIAD). Une introduction aux SIAD existants est suivie d'une présentation de la méthode interactive par réseau neuronal. Les éléments nécessaires à l'interaction avec le décideur sont ensuite décrits et appliqués au problème de choix de voiture. Cette seconde partie se termine par l'application de la méthodologie complète à un problème de décision très important et d'actualité dans le domaine de la protection de l'environnement d'aujourd'hui: le choix de sites de stockage de déchets industriels ultimes. Cette seconde partie permet de savoir comment fonctionne l'acquisition interactive des préférences du décideur, pour obtenir la flexibilité et l'incrémentalité typique des SIAD.

La troisième partie est orientée vers les problèmes d'*explication* dans les systèmes d'aide à la décision. Cette partie commence par une introduction générale et par une description des facilités d'explication développées dans l'outil StocDech. L'obtention d'explications à partir des réseaux de neurones est comparée à d'autres techniques. Finalement, l'intégration et l'implémentation des méthodes est décrit, ce qui permet de savoir comment obtenir des explications sur une décision produite par une technique neuronale.

Partie I

Apprentissage des fonctions de préférence par réseaux de neurones dans un système multicritère d'aide à la décision

Cette partie présente la manière d'utiliser les réseaux neuronaux ainsi que leurs capacités d'apprentissage automatique comme moyen de construire des *systèmes d'aide à la décision* (SAD) multicritère. La problématique traitée dans cette partie est celle de l'acquisition de préférences des décideurs pour des problèmes de décision de complexité variée. La méthodologie de construction des SAD utilisant des réseaux de neurones développée ici est avant tout destinée au traitement de problèmes de grande échelle, comme par exemple le choix de sites de stockage de déchets. Pour ce genre de problème, il est difficile d'identifier les décideurs réels, il y a des milliers d'actions (sites de stockage) potentielles et plus de dix critères de sélection. Ceci n'empêche pas une utilisation pour des problèmes de taille plus réduite, comme le choix d'une voiture.

L'identification de préférences d'un grand nombre d'acteurs, comme toute une population, à partir de données statistiques est difficile. Il existe depuis longtemps toute une gamme de méthodes d'*analyse statistique* et d'*analyse de données*, qui ont déjà été appliquées à des problèmes de décisions environnementales [Devillers & Karcher, 1991]. Malgré les efforts, ces méthodes d'analyse n'ont pas encore pu être bien exploitées pour l'analyse multicritère. Les techniques d'analyse des données ne permettent pas directement d'identifier des modèles de préférence. Les résultats obtenus par ces méthodes sont également difficiles à utiliser dans les SAD.

Chapitre 1: Les systèmes d'aide à la décision

Il existe encore d'autres approches qui sont destinées à l'identification de préférences à partir d'exemples de situations de décision. Ces méthodes présentent néanmoins des restrictions trop sévères pour traiter les *fonctions des préférences contextuelles* évoquées au Chapitre 1.3. Une approche par système expert permet de mieux représenter ces fonctions de préférence contextuelle, mais ne résoud pas le problème d'acquisition de connaissances (ou, dans notre cas, de préférences). C'est la raison qui conduit au choix d'une approche par *apprentissage automatique* (au sens de l'intelligence artificielle).

Les *réseaux neuronaux* ou *réseaux de neurones* font partie des méthodes d'apprentissage automatique. Les raisons qui conduisent au choix des réseaux neuronaux au lieu d'autres techniques d'apprentissage automatique sont multiples. Un avantage des réseaux neuronaux est le traitement facile des critères de sélection quantitatifs et la continuité du modèle mathématique des préférences.

Cette première partie débute par une introduction générale aux SAD et une présentation des problèmes normalement traités par ces systèmes. Cette discussion sur l'état de l'art permet d'une part, de définir le vocabulaire utilisé dans ce document et d'autre part, d'identifier les problèmes à résoudre dans les SAD existants. L'état de l'art des méthodes d'apprentissage automatique existantes explique les raisons du choix des réseaux de neurones; le reste de cette partie développe l'approche neuronale utilisée. Les composants principaux de cette approche sont le réseau de neurones INKA et les méthodes neuronales qui facilitent la saisie des préférences du décideur. L'utilisation du SAD neuronal est illustrée par une application de choix de voiture. Cette application permet également de découvrir certaines insuffisances du SAD neuronal. Résoudre ces insuffisances fait l'objet de l'approche interactive de la Partie II.

1. Les systèmes d'aide à la décision

Il existe un très grand nombre de méthodes d'aide à la décision multicritère. La plupart d'entre elles sont rassemblées et expliquées par [Roy, 1985], [Schärlig, 1985], [Vincke, 1989], [Pomerol & Barba-Romero, 1993] et [Roy & Bouyssou, 1993]. Sans qu'il soit utile de reprendre le contenu de ces méthodes en détail, leurs caractéristiques générales suffisent pour montrer leurs insuffisances potentielles face à un certain nombre de problèmes réels. Il est nécessaire de définir d'abord le vocabulaire utilisé.

1.1 Éléments taxinomiques

Toute méthode d'aide à la décision multicritère a comme objectif de proposer les moyens de déterminer un *ordre de préférences* ou *classement* d'un *ensemble d'actions potentielles* pour un problème donné. La nature de cet ordre dépend du problème traité. Les quatre problématiques de référence introduites par [Roy, 1985] et reprises par [Maystre et al, 1994] sont décrites dans le Tableau 1-1.

Problématique	Objectif	Résultat	Procédure
α	Choix d'un sous-ensemble contenant les actions « les meilleures » ou, à défaut, « satisfaisantes »	Choix	Sélection
β	Tri par affectation des actions à des catégories prédéfinies	Tri	Affectation
γ	Rangement de classes d'équivalence, composées d'actions, ces classes étant ordonnées de façon complète ou partielle	Rangement	Classement
δ	Description, dans un langage approprié, des actions et de leurs conséquences	Description	Cognition

Tableau 1-1. Problématiques de référence.

La détermination du classement se fait à l'aide de l'ensemble des *critères de sélection*, C . Pour chaque *action potentielle* a_i et chaque critère de sélection C_j , une *valeur d'utilité*, $g_j(a_i)$ (lire: « évaluation de l'action a_i selon le critère g_j ») est calculée par la *fonction d'utilité* du critère. L'ensemble des évaluations peut être représenté par un tableau à double entrée, appelé la *matrice des évaluations*, dans laquelle chaque ligne représente une action et chaque colonne un critère.

La matrice des évaluations peut ensuite être exploitée afin de calculer un classement de l'ensemble A . Parmi les méthodes appartenant à la problématique α , la méthode d'optimum de Pareto est la plus simple. S'il est possible de trouver une seule solution optimale de Pareto, cette solution est toujours la meilleure. Le concept d'optimalité Parétienne est défini de la manière suivante [Dong, 1992]:

Définition: Une solution $a_i \in A$ est une solution Parétienne (aussi appelée solution efficace, ou solution non-dominée), s'il n'existe pas de solution réalisable a_j telle que:

1. $\forall c, g_c(a_j) \geq g_c(a_i)$, et
2. $\exists c', g_{c'}(a_j) > g_{c'}(a_i)$.

Pour des problèmes de décision réels, il est extrêmement rare de trouver une seule solution Parétienne, ce qui nécessite l'utilisation de calculs plus sophistiqués permettant d'effectuer l'arbitrage. Cet arbitrage peut être fait de façons très différentes qui, pour la plupart, demandent au décideur d'exprimer ses préférences de manière plus ou moins détaillée. Les méthodes les plus anciennes de ce genre sont les méthodes de Borda et de Condorcet [Pomerol & Barba-Romero, 1993], qui sont des méthodes développées déjà au 18^e siècle.

La plupart des méthodes de SAD utilisées en pratique demandent au décideur d'affecter une importance aux critères de sélection. Cette importance est traduite par un *poids* p_i , qui peut être soit ordinal, soit cardinal. Un poids ordinal définit un ordre d'importance entre les critères, où un critère peut être plus/moins/aussi important qu'un autre. Un poids cardinal est un chiffre, souvent un pourcentage, qui permet de mieux détailler les importances des critères, mais qui est plus difficile à exprimer par le décideur.

Le concept d'importance des critères de sélection est, en réalité, présent dans tous les SAD, même si elle n'est pas toujours exprimée par l'intermédiaire d'un poids. L'ordre des actions peut même être directement calculé par les valeurs d'utilité $g_j(a_i)$. Les poids et les valeurs d'utilité sont les données d'un *processus d'agrégation*, qui permet d'obtenir une *valeur de préférence* ou *note*² globale pour chaque action. La valeur de préférence peut être une valeur absolue ou une valeur relative par rapport à une ou plusieurs autres actions, ce qui permet ensuite de déterminer le classement.

L'utilisation de la plupart des méthodes d'aide à la décision nécessite l'identification d'un décideur capable d'exprimer ses préférences sous le format imposé par la méthode.

² Les concepts de *valeur de préférence* et de *note* sont souvent appelés *valeur d'utilité* dans la littérature, mais ce concept est ici réservé à la valeur obtenue de la fonction d'utilité d'un critère. Ces concepts sont appropriés pour des modèles de préférence qui utilisent une note unique par action, ce qui n'est pas le cas pour des relations valuées, par exemple.

L'identification du décideur n'est néanmoins pas toujours simple. Un décideur n'est qu'un acteur parmi d'autres dans le processus de décision, qui est normalement obligé de prendre en compte l'avis d'autres acteurs pour aboutir à une solution. Dans [Roy, 1985], un acteur est défini comme suit:

*Un individu ou un groupe d'individus est **acteur** d'un processus de décision si, par son système de valeurs, que ce soit au premier degré du fait des intentions de cet individu ou groupe d'individus ou au second degré par la manière dont il fait intervenir ceux d'autres individus, il influence directement ou indirectement la décision. De plus, pour qu'un groupe d'individus (corps constitué ou collectivité) soit identifié comme un seul et même acteur, il faut que, relativement au processus, les systèmes de valeurs, systèmes informationnels et réseaux relationnels des divers membres du groupe n'aient pas à être différenciés.*

Cette définition a été davantage détaillée par [Jacquet-Lagrèze, 1981] et [Maystre et al., 1994]. Dans les problèmes les plus simples, le décideur est une personne pour qui il est suffisant de prendre en compte uniquement ses propres préférences pour prendre la décision. Un tel décideur a rarement vraiment besoin d'un SAD, sauf dans des cas où le nombre d'actions potentielles devient trop grand. Au cas où ce décideur décide malgré tout d'utiliser un système d'aide à la décision, le problème principal devient le transfert de ses préférences vers le SAD retenu. Quel que soit ce SAD, il impose une certaine représentation des préférences du décideur. Malheureusement, les représentations actuellement utilisées sont souvent trop difficiles à comprendre pour le décideur ou trop simplistes pour bien remplir leur fonction, ce qui est discuté dans le chapitre suivant.

1.2 L'état de l'art des systèmes d'aide à la décision

Deux « écoles » existent dans le domaine et suivent des principes de base assez différents [Maystre et al, 1994]. La première est « l'école Américaine », qui utilise le plus souvent une *fonction d'utilité additive* qui permet de combiner les valeurs d'utilité $g_j(a_i)$ en une note globale pour l'action. La méthode la plus simple de cette catégorie est celle de la somme pondérée, où la note globale d'une action est la somme pondérée des notes de chaque critère de sélection multipliée par le poids du critère. D'autres méthodes de ce type sont MAUT, *Multi Attribute Utility Theory* [Keeney & Raiffa, 1976] et MHM, *Méthode de hiérarchisation multicritère* [Saaty, 1984].

L'école Européenne favorise plutôt des méthodes fondées sur des notions de surclassement entre les actions potentielles. Les méthodes les mieux connues sont les méthodes ELECTRE, *Elimination Et Choix Traduisant la Réalité* [Roy, 1968, 1978], [Roy & Bertier, 1971, 1973], [Roy & Hugonnard, 1982a, b], [Roy & Skalka, 1985], [Roy & Bouyssou, 1991], [Yu, 1992a, b], [Maystre et al, 1994] et PROMETHEE, *Preference Ranking Organisation METHOD for Enrichment Evaluations* [Brans et al, 1984]. Les relations de surclassement permettent de construire le *graphe de surclassement*, à partir duquel est construit le classement final des actions potentielles. Ces méthodes permettent aussi d'établir des notions d'indifférence et d'incomparabilité entre les actions, ce qui correspond souvent le mieux à la façon de penser du décideur. Malheureusement ces deux notions soulèvent souvent des problèmes d'interprétation du graphe de surclassement. De nombreuses modifications aux méthodes de base ont été proposées, notamment pour les méthodes ELECTRE, ce qui en explique le grand nombre de variantes. Ces modifications sont liées à l'utilisation de *pseudo-critères*, à l'introduction de plusieurs *seuils de surclassement* et à l'utilisation de *concepts flous*. Ces modifications peuvent éviter la définition de fonctions d'utilité pour les critères de sélection, tout en permettant d'améliorer les performances techniques de la méthode. La méthode devient néanmoins souvent encore plus difficilement accessible au décideur.

Il existe encore un grand nombre d'autres méthodes qui n'appartiennent ni à l'une ni à l'autre des deux « écoles », comme QUALIFLEX [Paelinck, 1978] et les méthodes utilisant des principes de coût et bénéfice [Raïffa, 1968, 1973] et [Schlaifer, 1959, 1969]. Toutes ces méthodes demandent, elles aussi, l'obtention a priori de poids et d'autres paramètres.

Il est généralement admis que l'expression a priori de préférences du décideur est très difficile, ce qui explique l'apparition de méthodes permettant une obtention progressive des préférences. Ce sont les Systèmes Interactifs d'Aide à la Décision, SIAD, discutés dans la Partie II. Avant d'aborder le sujet des SIAD, il est nécessaire d'aborder le problème qui consiste à établir des fonctions de préférence contextuelle. Dans le cas de telles fonctions de préférence, les poids et les valeurs d'utilité peuvent varier en fonction du contexte, qui est défini par la ou les action(s) traitée(s).

1.3 Fonctions de préférence contextuelle

Un grand inconvénient concernant pratiquement tous les SAD ainsi que la plupart des SIAD est issue du fait qu'ils sont fondés sur les deux *conditions de base* suivantes:

- L'**importance**
- L'**utilité** des valeurs

d'un critère ne peut pas **changer en fonction de valeurs des critères**. Un modèle de préférence qui n'est pas limité par ces deux conditions de base est un *modèle de préférence contextuel*. Dans le cas d'un modèle de préférence représenté comme une fonction mathématique, nous utilisons plutôt la notion de *fonction de préférence contextuelle* [Frâmling et al., 1995].

Définition: Une fonction de préférence contextuelle est un modèle des préférences du décideur où l'importance des critères de sélection et l'utilité des valeurs de critères de sélection peuvent varier en fonction des valeurs des critères de sélection. Le contexte est défini par les propriétés de l'action évaluée.

Le besoin de représenter des fonctions de préférence contextuelle est évoqué par exemple dans [Pomerol & Barba-Romero, 1993]. Il est facile de trouver des exemples de situations de décision qui nécessitent de telles fonctions de préférence. Les exemples sont choisis pour un problème de choix de voiture (voir l'Annexe A). Les deux premières règles illustrent des cas où la première condition de base pose des problèmes, la troisième règle illustre un cas où c'est la deuxième condition qui pose des problèmes:

« SI le prix de la voiture > 200.000,- francs français
ALORS diminuer l'importance des autres critères car je peux
difficilement payer plus »

« SI la consommation moyenne > 11L/100km ALORS diminuer
l'importance des autres critères car ça deviendra vite trop
cher »

« SI la puissance maximale dépasse 200 chevaux alors qu'il
n'y a pas de traction 4x4 ALORS réduire l'utilité de la

puissance maximale car je patinerai tout le temps et le contrôle de la voiture deviendra plus difficile »

Un décideur réel qui est confronté à un problème d'achat d'une voiture a le plus souvent des préférences de ce type, mais de telles préférences sont quasiment impossibles à modéliser avec les SAD existants. Ces trois règles seront utilisées sous différentes formes plus tard pour comparer les capacités d'expression des préférences des différents SAD.

Un deuxième exemple provenant du problème de choix de sites de stockage de déchets industriels ultimes (voir l'Annexe A) est illustré par les règles suivantes:

« SI il y a une connexion ferroviaire à proximité du site
ALORS diminuer l'importance des critères concernant l'accès
routier »

« SI il y a une exploitation de marnes à proximité du site
ALORS diminuer l'utilité d'avoir une faible perméabilité »

Ces deux règles illustrent bien que la différence entre l'utilité d'une valeur d'un critère et le poids d'un critère soit parfois assez subtile. Dans ces deux règles il serait possible d'interchanger les mots importance et utilité. L'importance est un concept plus global, qui touche toutes les valeurs possibles du critère de sélection. L'utilité est plus locale et concerne uniquement une valeur du critère de sélection. L'approche neuronale présentée dans ce travail permet l'utilisation de telles fonctions de préférence sans faire de séparation entre importance et utilité. Il est néanmoins possible de retrouver ces concepts par l'intermédiaire des méthodes d'obtention d'explications, présentées dans la Partie III.

2. Fonctions de préférence contextuelle par combinaison d'analyse multicritère et d'une base de règles (StocDech)

StocDech est un logiciel développé à l'École des Mines de Saint-Étienne dans le cadre de ce travail pour aider à résoudre le problème du choix de sites de stockage de déchets industriels ultimes dans la région Rhône-Alpes en France. Ce logiciel permet la définition de fonctions

de préférence contextuelle par une combinaison des méthodes de l'analyse multicritère et d'un « système expert » à base de règles de décision.

Deux méthodes d'analyse multicritère sont incluses dans StocDech - la méthode de la somme pondérée et la méthode ELECTRE I. La base de règles et la logique de résolution utilisées par StocDech sont celles de l'outil « Nexpert Object v.2.0 » [Neuron Data, 1991]. Les détails sur l'implémentation de ces méthodes se trouvent dans l'Annexe C. Le logiciel même est décrit dans les quatre volumes de documentation [Främling & Graillot, 1994], où se trouve également une description plus détaillée des méthodes utilisées.

Il y a plusieurs raisons pour inclure une base de règles dans l'outil. D'abord, il est parfois plus facile pour des experts et des scientifiques de formuler leurs connaissances sous la forme de règles que par les modèles de l'analyse multicritère classique. Ensuite, les règles de décision permettent normalement de mieux expliquer les résultats obtenus que ceux fournis par l'analyse multicritère.

2.1 Analyse multicritère par systèmes experts

Les bases de règles sont souvent appelées des systèmes experts, dont la définition suivante s'inspire de celles de [Royer, 1990] et [Ayel & Rousset, 1990]:

Définition: Un système expert est un logiciel ou une partie d'un logiciel destiné à assister l'homme dans des domaines où est reconnue une expertise humaine. Dans les systèmes experts à base de règles cette connaissance est stockée dans la base de règles, qui est interprétée par le moteur d'inférences utilisant la base de faits pour stocker les données propres au problème à traiter et les déductions faites par le système.

Les connaissances sont exprimées avec des formules de la logique de prédicats [Genesereth & Nilsson, 1987]. Ces clauses sont d'habitude appelées des *règles de production* qui contiennent une partie prémisses et une partie actions. Si les prémisses sont vraies le moteur d'inférence exécute les actions.

Les premiers systèmes experts de ce type ont été développés dans les années soixante-dix aux États-Unis [Buchanan & Shortliffe, 1984]. Les systèmes experts ont eu beaucoup de succès, mais sans jamais atteindre la popularité attendue. Cet effet est avant tout dû au fameux *goulot d'étranglement*, (voir par exemple [Dreyfus, 1991] ou [Beaune, 1992]), qui résulte de la

difficulté de l'expert à exprimer sa connaissance et du grand nombre de règles nécessaire pour la plupart des applications d'intérêt pratique. Il se produit même très vite une explosion combinatoire quand le nombre de critères de sélection devient grand et quand il y a beaucoup de valeurs possibles pour chaque critère. Il devient notamment difficile de s'assurer de la maintenance de la cohérence et de la complétude d'une telle base de règles [Ayel & Rousset, 1990]. Il existe néanmoins un grand nombre de systèmes de maintien de la cohérence [Léa Sombé, 1989], dont les plus connus sont probablement TMS (Thruth-Maintenance System) [Doyle, 1979] et ATMS (Assumption-based Thruth-Maintenance System) [De Kleer, 1984, 1986a, b].

Pour faciliter la formalisation de règles on utilise souvent une hiérarchie de critères et de sous-critères. Le choix d'un site de stockage de déchets, par exemple, peut se décomposer à l'aide d'une classification par l'accès routier, les nuisances vis-à-vis de la population locale, etc. Si elle est bien choisie, la décomposition simplifie suffisamment le problème pour rendre possible la définition des règles. Il y a une correspondance directe entre cette hiérarchie et l'arborescence de la méthode MHM [Saaty, 1981, 1984].

Un des avantages des règles de production est qu'il est très facile d'exprimer des connaissances certaines et précises. Les bases de règles permettent également en partie d'éviter la subjectivité introduite dans les poids des critères des méthodes d'analyse multicritère. Par contre, il devient vite très difficile de définir des règles qui couvrent toutes les valeurs possibles des critères de sélection. La base de règles présentée dans l'Annexe C, destinée à la sélection d'un site de stockage de déchets industriels ultimes en Suisse avec treize critères de sélection, contient déjà 45 règles, avec un niveau de précision qui demeure assez faible.

2.2 Combinaison de méthodes d'analyse multicritère et d'une base de règles

A priori, il semble avantageux de combiner les techniques d'analyse multicritère et de bases de règles. L'analyse multicritère permettrait dans ce cas de définir une information globale et de traiter les aspects subjectifs de la décision, c'est-à-dire les aspects où ce sont les préférences du décideur qui comptent. La base de règles permet plutôt de traiter les aspects plus objectifs, pour lesquels il existe une certaine expertise reconnue. La base de règles

servirait dans ce cas comme un « filtre » qui effectue un prétraitement de l'information, permettant ainsi de diminuer la quantité d'information à traiter par le décideur. StocDech permet cette combinaison d'une façon très simple, illustrée dans la Figure 2-1.

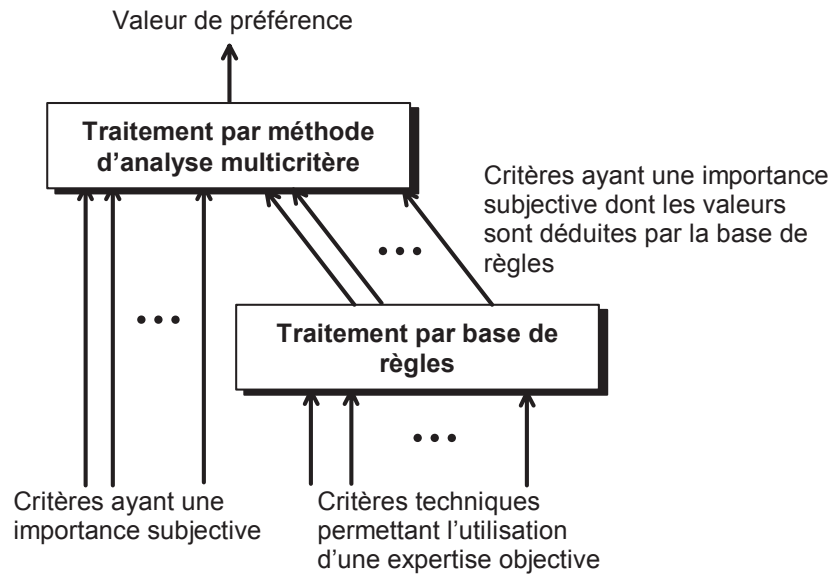


Figure 2-1. Combinaison de l'analyse multicritère et de la base de règles dans StocDech.

StocDech permet de définir trois types de critères de sélection:

- Des critères de type quantitatif.
- Des critères de type qualitatif (dont les critères booléens).
- Des critères déduits (et forcément qualitatifs dans le cas d'une base de règles).

La déclaration d'un critère de sélection comme étant un critère déduit signifie qu'il existe des règles dans la base de règles permettant de déduire la valeur de ce critère à partir des valeurs déjà fournies pour d'autres critères. Un exemple d'illustration serait l'utilisation des critères « puissance maximale », « poids » et « coefficient aérodynamique » pour déterminer la valeur du critère « vitesse maximale » d'une voiture. La valeur obtenue serait ensuite traitée par l'analyse multicritère en relation avec d'autres critères comme le prix, la consommation, le confort, etc.

Pour assurer la compatibilité des deux méthodes et des critères qu'elles utilisent ou qu'elles produisent, StocDech transforme d'abord toute valeur quantitative en valeur qualitative à l'aide d'*intervalles de valeurs*. Les valeurs qualitatives ainsi obtenues sont ensuite transformées en valeurs d'utilité, où la valeur la moins favorable obtient une valeur de un, la

deuxième une valeur de deux et ainsi de suite. Pour la méthode de la somme pondérée, les valeurs d'utilité sont encore normalisées par une division par la valeur la plus grande des critères considérés. Les valeurs de tous les critères subissent le même traitement de transformation de valeur quantitative en valeur qualitative puis en valeur d'utilité, sauf pour les critères qualitatifs où la première transformation n'est pas nécessaire. La base de règles utilise les valeurs qualitatives, les méthodes d'analyse multicritère les valeurs d'utilité.

Dans la plupart des cas il est plus logique d'appliquer la base de règles avant l'analyse multicritère. Il est néanmoins imaginable qu'une approche mixte pourrait être utile dans certains cas, permettant une chaîne de traitement type: analyse multicritère \Rightarrow base de règles \Rightarrow analyse multicritère \Rightarrow ... Pour les problèmes traités avec StocDech, ceci ne s'est pas révélé utile et StocDech utilise uniquement une séquence base de règles \Rightarrow analyse multicritère. Il est, bien sûr, également possible d'utiliser uniquement la base de règles.

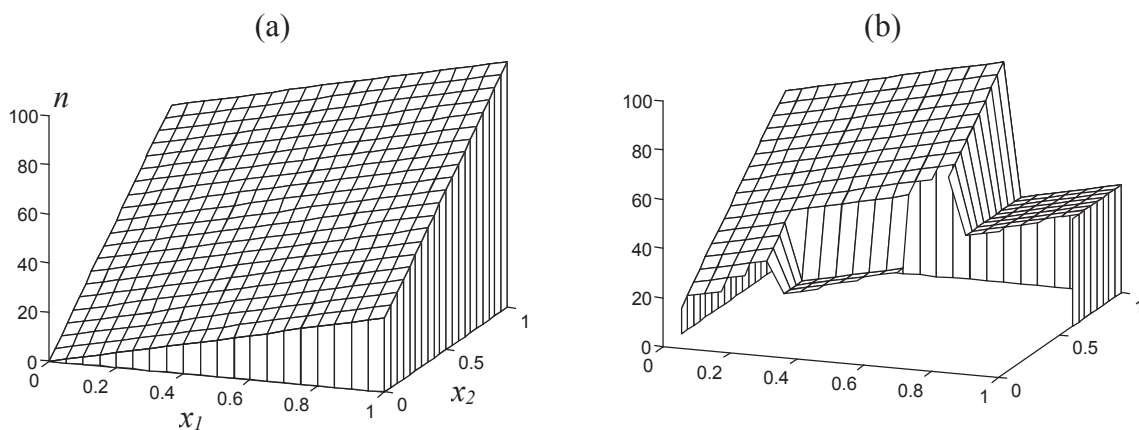


Figure 2-2. Exemple d'une fonction de préférence (a) par somme pondérée pour deux critères de sélection avec des poids $p_1 = 30$ et $p_2 = 70$, (b) par somme pondérée modifiée par deux règles.

Un exemple simple d'une fonction de préférence en trois dimensions permettra d'illustrer quelques avantages et inconvénients de l'approche par base de règles. Les deux premières dimensions sont celles des deux critères de sélection (x_1 et x_2) et la troisième celle de la valeur de préférence de l'action (n).

La Figure 2-2(a) montre une fonction de préférence obtenue par une somme pondérée simple, où les fonctions d'utilité sont linéaires pour les deux critères de sélection. La Figure 2-2(b) montre la fonction de préférence obtenue par la combinaison de la somme pondérée et les deux règles suivantes:

SI $x_1 > 0.7$ & $x_2 > 0.2$ ALORS $x_2 = 0.2$

SI $x_1 > 0.2$ & $x_1 < 0.6$ & $x_2 > 0.1$ & $x_2 < 0.5$ ALORS $x_2 = 0.1$

La première règle donne à peu près le même effet que la règle concernant le prix d'une voiture mentionnée dans le Chapitre 1.3. La deuxième règle correspond à une situation où certaines combinaisons de valeurs sont très défavorables à la décision. Les deux fonctions de préférence de la Figure 2-2 nous permettent de constater le suivant:

1. Il est impossible d'obtenir la fonction de préférence de la Figure 2-2(b) avec une méthode classique d'analyse multicritère. Raison: Il s'agit d'une fonction de préférence contextuelle, où il est nécessaire de pouvoir changer les poids et les valeurs d'utilité en fonction des valeurs d'autres critères.
2. Pour un grand nombre de situations de décision, il est nécessaire de pouvoir représenter de telles fonctions de préférence. Raison: Ceci a déjà été montré pour les problèmes du choix de voiture et du choix de sites de stockage de déchets.
3. Dans le cas d'une base de règles seule, il faut un grand nombre de règles pour définir toute la fonction de préférence et cette fonction ne sera pas continue. Raison: La valeur de sortie numérique changerait en escalier à cause de la non-continuité du traitement qualitatif des bases de règles classiques.
4. La combinaison de l'analyse multicritère et d'une base de règles pose, elle aussi, des problèmes de continuité pour la fonction de préférence. Raison: Même raison que pour le point précédent, mais il faut moins de règles puisqu'elles traitent une partie de l'espace d'entrée plus petite.

Les deux derniers points concernant les bases de règles sont liés aux limitations de la logique des prédicats. Les bases de règles sont bien adaptées pour traiter des problèmes avec uniquement des variables qualitatives. Par contre, elles présentent des difficultés importantes pour traiter les variables quantitatives car elles produisent des fonctions discontinues et des changements abrupts de la valeur de préférence dûs à de très faibles variations des valeurs de critères de sélection.

2.3 La logique floue comme outil d'expression des fonctions de préférence contextuelle

La logique floue est une technique qui permet de mieux traiter les variables quantitatives. Cette technique développée par [Zadeh, 1965] a produit un nombre d'applications impressionnant ces dernières années [Zadeh et al., 1975], [Harmon, 1993a], même si les premières idées venaient déjà de [Lukasiewicz, 1930]. L'un des domaines d'application où la logique floue a eu le plus de succès est le contrôle non-linéaire en temps réel [Catherin, 1991]. Ce sont des applications où les variables d'entrée et de sortie sont quantitatives.

La logique floue peut effectuer une transformation des valeurs quantitatives en valeurs qualitatives à l'aide des ensembles flous, contrairement aux bases de règles qui le font par intervalles de valeurs³. Un ensemble flou ne fait pas uniquement une transformation quantitatif-qualitatif, mais il associe également un *coefficient de certitude* à la valeur qualitative. Il est donc possible et raisonnable d'avoir plusieurs valeurs qualitatives qui soient vraies en même temps, mais avec des certitudes différentes. Ceci implique qu'il y a plusieurs chemins d'inférence à développer, qui produisent le plus souvent plusieurs résultats auxquels sont associés des coefficients de certitude. Une procédure dite de « défuzzification » permet de transformer ces valeurs qualitatives multiples en une seule valeur quantitative.

Les coefficients de certitude étaient utilisés déjà dans les premiers systèmes experts [Buchanan & Shortliffe, 1984]. MYCIN effectuait des analyses médicales et donnait des solutions auxquelles était associé un degré de vraisemblance, tout en restant mal adapté à traiter des variables quantitatives. L'introduction des ensembles flous permet non seulement de mieux traiter les variables quantitatives, mais aussi de définir n'importe quelle transformation de valeurs réelles de $\mathfrak{R}^N \rightarrow \mathfrak{R}^M$, où N est la dimension du vecteur d'entrée et M est la dimension du vecteur de sortie.

La logique floue a déjà été utilisée dans certaines applications d'analyse multicritère [Grabot, 1990]. L'inconvénient majeur de cette technique pour son application au domaine de l'analyse multicritère reste le grand nombre de règles et de paramètres à définir. Une combinaison de la logique floue et des méthodes d'analyse multicritère pourrait être une voie intéressante à

³ Une telle transformation est par exemple « SI $x > 10$ ET $x < 30$ ALORS x est Moyen ».

étudier, car cela permettrait certainement de diminuer largement le nombre de règles requises. Ce couplage n'est malheureusement pas sans problème, car il faudrait au moins trouver les moyens d'assurer un transfert continu d'une technique à l'autre.

La logique floue présente donc l'avantage de permettre l'expression de préférences contextuelles, mais le nombre de paramètres trop grand rend cette technique inaccessible pour le décideur. La non-linéarité souvent nécessaire pour permettre d'établir des fonctions de préférence contextuelle rend vite l'expression mathématique de cette fonction très difficile. Plus le nombre de critères de sélection devient grand, plus l'expression de cette fonction devient difficile. La seule solution qui semble réaliste pour résoudre ce problème est d'essayer d'identifier implicitement la fonction de préférence à partir d'exemples de préférences des décideurs et des acteurs du problème. L'identification des fonctions et l'extraction des connaissances à partir d'exemples s'appelle l'*apprentissage automatique* dans le domaine de l'intelligence artificielle.

Pour les bases de règles classiques il existe un grand nombre de techniques d'apprentissage automatique possibles. Pour construire des bases de règles utilisant la logique floue, c'est avant tout les techniques de réseaux neuronaux qui ont été utilisées [Pradenc, 1991], [Kosko, 1992], [Harmon, 1993b], même si certaines techniques d'apprentissage probabiliste sont aussi applicables. La logique floue et les réseaux neuronaux ont beaucoup de points en commun.

3. Apprentissage automatique des fonctions de préférence

L'apprentissage automatique dans le contexte informatique signifie la capacité d'un logiciel à améliorer son comportement à l'aide d'exemples. Cela peut par exemple signifier l'ajout, la suppression ou la modification des règles dans une base de règles, ou bien l'ajout ou la suppression des neurones et le changement des valeurs des poids d'un réseau de neurones. Les exemples sont considérés comme des mesures du système que l'on cherche à comprendre et à modéliser. Dans notre cas, ce sont des exemples de situations de décision, pour lesquelles le décideur fournit une réponse qui est considérée comme bonne. L'apprentissage automatique est souvent combiné avec des *méthodes d'acquisition de connaissances*, qui facilitent l'interaction avec l'expert ou le décideur (voir par exemple dans [Senoune, 1995]).

Tout apprentissage nécessite l'existence d'un *ensemble d'apprentissage*, où se trouvent les exemples *d'apprentissage* qui comportent dans notre cas au moins deux informations:

1. Un jeu de valeurs d'entrée, dans notre cas les valeurs des critères de sélection pour une action, potentielle ou non.
2. Un jeu de valeurs de sortie correspondant à la bonne réponse, qui dans notre cas est la bonne valeur de préférence de l'action.

Il s'agit donc d'un *apprentissage supervisé*, puisque la bonne réponse est fournie avec l'exemple. L'apprentissage non-supervisé sera discuté dans le Chapitre 6.1.

Dans le cas de l'apprentissage supervisé, le module d'apprentissage essaie de trouver les bonnes relations entre les valeurs d'entrée et les valeurs de sortie. Ces relations peuvent être des règles de décision ou des poids d'un réseau neuronal, par exemple. Le *système de résolution* résultant permet à la suite de reproduire les bonnes valeurs de sortie quand les valeurs d'entrée des exemples d'apprentissage sont présentées à nouveau.

En pratique, tous les systèmes d'apprentissage automatique essaient de trouver des relations qui permettent de déduire les bonnes réponses pour des valeurs d'entrée différentes de celles présentes dans l'ensemble d'apprentissage. Ceci s'appelle le processus de *généralisation*. Un système de résolution peut facilement apprendre les bonnes réponses pour tout l'ensemble d'apprentissage, mais en revanche donner des résultats complètement erronés pour tout autre jeu d'entrée s'il possède une mauvaise capacité de généralisation.

L'estimation des capacités de généralisation se fait le plus souvent à l'aide d'un *ensemble de validation*, qui contient des *exemples de validation* qui ont des valeurs d'entrée différentes des *exemples d'entraînement* même s'ils contiennent la même information. Les *ensembles d'entraînement* et de validation sont normalement deux sous-ensembles de l'ensemble d'apprentissage indépendants. D'autres méthodes d'estimation des capacités de généralisation existent, la partition de l'ensemble d'apprentissage y est moins stricte, permettant une meilleure utilisation des exemples disponibles.

3.1 Méthodes d'aide à la décision multicritère utilisant l'apprentissage symbolique automatique

Le concept d'*apprentissage symbolique automatique* signifie ici un apprentissage qui permet de trouver des liens de dépendance entre des symboles, souvent sous forme de règles de production. Cet apprentissage permet la construction de systèmes experts à partir d'exemples. Certaines méthodes sont également capables de traiter et de générer des probabilités ou coefficients de certitude, parmi lesquelles la méthode des arbres de décision probabilistes de Quinlan, *ID3* [Quinlan, 1986, 1987a, 1990] et *C4* [Quinlan, 1987b], qui est certainement la plus connue. D'autres méthodes sont présentées par exemple dans [Kodratoff & Michalski, 1990].

L'apprentissage symbolique utilise avant tout les méthodes de déduction et d'induction. Déduction signifie inférer des faits ou des règles plus spécifiques à partir de faits ou de règles généraux. Induction, par contre, signifie l'inférence de règles plus générales en utilisant des règles et faits spécifiques. En apprentissage, on utilise le plus souvent l'induction pour la généralisation. Un avantage majeur de l'apprentissage automatique est qu'il permet parfois de réduire les problèmes d'explosion combinatoire des systèmes experts.

Certaines autres méthodes d'apprentissage automatique sont aussi qualifiées de « symbolique ». Il s'agit notamment des algorithmes génétiques (en anglais: *genetic algorithms*) [Holland, 1975], [De Jong, 1975] et du raisonnement à base de cas (en anglais: *case-based learning*) [Kolodner, 1987]. Néanmoins l'utilité de ces méthodes pour notre problématique ne semble pas évidente. Les algorithmes génétiques sont une méthode permettant avant tout de faire de l'optimisation [Grefenstette, 1985, 1987]. Le raisonnement à base de cas souffre des mêmes difficultés à traiter des valeurs quantitatives que les bases de règles.

Dans la méthode PLEXIGLAS [Rommel, 1989] des méthodes d'apprentissage symbolique automatique sont utilisées pour automatiquement identifier des règles de surclassement de type ELECTRE. Il s'agit alors d'un problème de classification par règles de surclassement, ce qui diminue les problèmes de non-continuité dans le modèle de préférence.

Les travaux récents utilisant des ensembles approximatifs (en anglais: *rough sets*) et l'apprentissage automatique présentés dans [Pawlak & Slowinski, 1994a, b], [Slowinski, 1994] et [Slowinski & Stefanowski, 1994] développent encore plus cette approche par

Chapitre 3. Apprentissage automatique des fonctions de préférence

apprentissage automatique, tout en la rendant mieux adaptée aux problèmes réels. Les ensembles approximatifs étaient introduits par Z. Pawlak en début des années 80. Ils permettent de quelque sorte de construire des classes d'objets flous, où les classes construites correspondent à des décisions possibles. Il est, par exemple, possible de dire si l'action a_i est meilleure, aussi bonne ou pire que l'action a_j . Le niveau de précision de surclassement dépend de l'échelle utilisée pour indiquer la « force » du surclassement.

Le principe de base des ensembles approximatifs est de, à partir d'exemples d'apprentissage, trouver des règles qui expliquent l'appartenance des exemples à une classe ou à l'autre. Les règles peuvent être soit déterministes, soit non déterministes en fonction de leur applicabilité à chaque situation. Les règles non déterministes sont utilisées quand il y a un conflit dans l'appartenance aux classes entre deux ou plusieurs exemples d'apprentissage. La classification des actions potentielles dans un problème de décision se fait par ces mêmes règles. S'il y a une seule règle applicable à l'action, le résultat est sans ambiguïté. Par contre, s'il y a plusieurs règles applicables en même temps, ou s'il n'y en a aucune, il devient nécessaire d'introduire des notions de distance pour choisir la règle à utiliser. Cette technique souffre ainsi des mêmes problèmes que d'autres techniques utilisant des notions de distance.

Les avantages de cette technique sont d'être simple au niveau de la compréhension et du calcul, de la production de règles qui expliquent les préférences et de la capacité à traiter l'incertain et les données manquantes. Un inconvénient est le fait d'utiliser uniquement des classes comme résultat, ce qui diminue toujours un peu la précision théorique par rapport à des valeurs continues. Il s'agit également d'une technique adaptée essentiellement à traiter des données qualitatives plutôt que des données quantitatives. Il reste, bien sûr, toujours possible d'utiliser des ensembles flous ou d'autres techniques pour les traiter, mais dans ce cas l'utilisation de la technique devient plus compliquée.

L'apprentissage symbolique automatique est, en effet, souvent mieux adapté aux problèmes de classification qu'un apprentissage numérique. C'est le cas par exemple dans [Weiss & Kulikowski, 1991], où des méthodes d'apprentissage symbolique automatique sont comparées avec le type de réseaux de neurones actuellement le plus utilisé, celui de rétro-propagation [Rumelhart et al., 1986]. Les performances du réseau neuronal étaient pratiquement toujours moins bonnes. Il faut néanmoins constater que le réseau de rétro-propagation n'est pas toujours le réseau neuronal le plus adapté aux problèmes de classification simples. Ses capacités de régression non-linéaire deviennent souvent utiles uniquement pour des problèmes

assez complexes. Les résultats obtenus dépendent également beaucoup de la nature même des variables utilisées.

L'utilisation des réseaux de neurones plutôt que l'apprentissage symbolique automatique est justifiée s'il s'agit d'obtenir des fonctions de préférence contextuelle et continues. Dans le domaine du raisonnement symbolique, il semble que la logique floue soit la seule à offrir une solution satisfaisante à la condition de continuité. Le lien fort entre la logique floue et certains réseaux de neurones et les capacités d'apprentissage automatique des réseaux de neurones conduisent à choisir les réseaux de neurones comme technique.

Il y a déjà eu certains travaux sur l'utilisation des réseaux de neurones comme méthodes d'analyse multicritère. Des réseaux neuronaux multicouche à rétro-propagation étaient utilisés dans [Wang & Malakooti, 1992] pour apprendre la fonction de préférence des décideurs à partir d'exemples. Comme elle est montré dans l'Annexe B, ce type de réseau présente beaucoup de problèmes au niveau de l'apprentissage, spécialement quand le nombre de critères de sélection dépasse trois ou quatre. Les exemples utilisés dans [Wang & Malakooti, 1992] se limitent à des problèmes à deux critères de sélection. Ils se limitent également à des fonctions de préférence monotones, ce qui n'est pas toujours le cas des fonctions de préférence réelles, comme illustré par l'exemple de choix d'automobile. Il s'agit dans leur cas notamment d'un SAD non interactif, ce qui limite encore l'utilité de leur approche par rapport à celle présentée ici.

3.2 Les réseaux de neurones

L'étude des réseaux de neurones artificiels a débuté dans les années 40 [McCulloch & Pitts, 1943]. Le nom de réseaux de neurones vient du fait que cette technique s'inspire de la connaissance sur le fonctionnement du cerveau humain pour effectuer un apprentissage automatique. Jusqu'à la publication de l'ouvrage de [Minsky & Papert, 1969], beaucoup de travail avait déjà été effectué dans ce domaine, mais les insuffisances de l'époque soulevées par Minsky et Papert ont provoqué une perte d'intérêt quasi-total pour les réseaux de neurones. Un regain d'intérêt s'est manifesté au début des années 80, suite aux nouveaux modèles de réseaux neuronaux présentés dans [Hopfield, 1982], [Kohonen, 1984] et [Rumelhart et al., 1986]. Il existe maintenant un grand nombre d'ouvrages où les méthodes essentielles sont expliquées, comme [Decamp & Amy, 1988], [Hecht-Nielsen, 1990], [Blayo

et al., 1993] et [Gallant, 1993]. C'est la raison pour laquelle la présentation des réseaux de neurones restera ici assez courte et générale.

Les réseaux neuronaux ont été utilisés dans de nombreux domaines d'application: mémoires associatives, diagnostic, traitement du signal, reconnaissance d'images, reconnaissance de la parole, compression de données, contrôle, prévision, analyse de données, etc. Un réseau neuronal est essentiellement un outil mathématique qui permet d'identifier des fonctions mathématiques⁴. Dans ce travail, les réseaux neuronaux sont utilisés pour l'identification des fonctions de préférence des décideurs.

Dès que l'on évoque les problèmes d'approximation de fonctions numériques, les méthodes d'apprentissage symbolique automatique deviennent peu intéressantes à cause des difficultés de traiter des valeurs quantitatives de façon continue. Par contre, il existe toute une gamme de méthodes d'identification qui sont plus adaptées à cette tâche. La plupart d'entre elles sont fondées sur l'application d'une fonction « type », dont les paramètres sont optimisés pour « coller » le mieux possible aux exemples d'apprentissage. Les fonctions sinusoïdes, polynomiales, ... sont des fonctions « type ». Ces méthodes sont néanmoins difficiles à utiliser pour des problèmes de grande dimension et pour des fonctions très irrégulières.

D'autres méthodes, comme les *splines*, permettent également à traiter des fonctions irrégulières, mais les problèmes de grande dimension restent difficiles. De nombreuses expériences comparatives ont cependant montré que certains réseaux de neurones donnent de meilleures performances que les splines pour un grand nombre de problèmes [Platt, 1991], [Deco & Ebmeyer, 1993].

L'identification d'une fonction de préférence univoque suppose que le décideur soit rationnel et cohérent, même si le décideur humain ne peut probablement jamais l'être à 100%. Il est difficile de représenter un décideur qui un jour donne une réponse et le lendemain son contraire. Dans le cas d'un tel décideur, un réseau neuronal qui fonctionne par des principes de régression ne donnera ni la première réponse ni la deuxième, mais la moyenne des deux. Une fonction de préférence univoque nécessite également de respecter la notion de transitivité

⁴ Le domaine de la classification est ici considéré comme un cas spécial de l'approximation de fonctions.

(si $A > B$ et $B > C$ donc $A > C$). Un autre inconvénient est que ce modèle ne prend pas en compte la notion d'incomparabilité des méthodes Electre, par exemple. L'incomparabilité entre deux actions peut néanmoins être détectée par une carte auto-organisatrice, comme discuté dans l'Annexe A, Chapitre 3.2.

L'inconvénient majeur des réseaux de neurones est lié à leur aspect « boîte noire », car la fonction mathématique représentée par le réseau de neurones devient vite trop complexe pour l'analyser et la comprendre directement. C'est une des raisons exposées par exemple dans [Slowinski, 1994] pour utiliser une approche par apprentissage symbolique automatique plutôt qu'une approche neuronale. Les méthodes d'extraction d'explications présentées dans la Partie III permettent néanmoins, au moins en partie, de résoudre ce problème.

Il existe un grand nombre de types de réseaux de neurones et il est souvent difficile de choisir celui à utiliser. Pour le problème d'identification des fonctions de préférence, aucun des réseaux de neurones existants ne semblait bien adapté au cadre de ce travail à cause du nombre d'exemples d'apprentissage souvent très limité. C'est la raison principale pour laquelle nous avons développé un réseau neuronal particulier, INKA, décrit dans le Chapitre 4.

3.2.1 Le neurone artificiel

La structure d'un neurone artificiel est représentée dans la Figure 3-1. Les valeurs d'entrée (x_1, x_2, \dots, x_N) et les poids associés aux entrées (w_1, w_2, \dots, w_N) sont les variables de la *fonction d'activation* qui détermine la *valeur d'activation* du neurone. Cette valeur d'activation est ensuite passée comme argument à la *fonction de sortie*, qui détermine la *valeur de sortie* du neurone, y' . L'entrée supplémentaire y sert pour indiquer au neurone la *valeur de sortie attendue* pour que le neurone puisse corriger ses poids afin de s'approcher de cette valeur. Malheureusement le vocabulaire n'est pas encore bien établi dans le domaine des réseaux neuronaux et d'autres terminologies sont fréquemment utilisées.

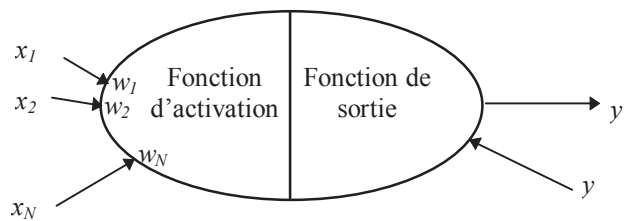


Figure 3-1. Schéma d'un neurone artificiel.

La fonction d'activation est le plus souvent la somme pondérée des valeurs d'entrée:

$$S = \sum_{i=1}^N w_i x_i \quad (1)$$

où S est la valeur d'activation, w_i est le poids de l'entrée i et x_i est la valeur de l'entrée i . La valeur d'activation devient l'argument de la fonction de sortie, qui est le plus souvent une des formes présentées ci-dessous. Une fonction de sortie fréquemment rencontrée est la simple multiplication par un, la valeur d'activation est alors directement la valeur de sortie. Le choix de la fonction de sortie dépend de l'application et du comportement souhaité. S'il faut avoir des sorties binaires c'est la première fonction que l'on choisit habituellement.

Une entrée spéciale est souvent introduite pour chaque neurone. Cette entrée, normalement appelée *biais* (en anglais: *bias*), sert à normaliser la fonction d'activation. La valeur de cette entrée est toujours égale à 1, ce qui permet une normalisation grâce à la valeur du poids correspondant.

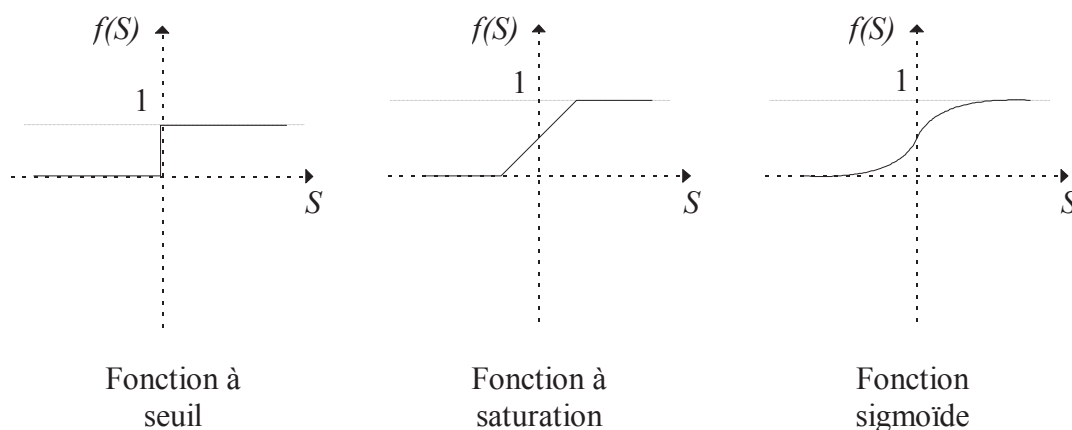


Figure 3-2. Exemples de fonctions de sortie.

3.2.2 Réseau neuronal

Un réseau de neurones se compose de neurones qui sont interconnectés de façon à ce que la sortie d'un neurone puisse être l'entrée d'un ou plusieurs autres neurones. Les huit composants principaux d'un réseau de neurones sont [Rumelhart et al., 1986]:

- Un *ensemble de neurones*.
- Un *état d'activation* pour chaque neurone (actif, inactif, ...).
- Une *fonction de sortie* pour chaque neurone ($f(S)$).
- Un *modèle de connectivité* entre les neurones (chaque neurone est connecté à tous les autres, par exemple).
- Une *règle de propagation* pour propager les valeurs d'entrée à travers le réseau vers les sorties.
- Une *règle d'activation* pour combiner les entrées d'un neurone (très souvent une somme pondérée), ici appelée une fonction d'activation.
- Une *règle d'apprentissage*.
- Un *environnement* d'opération (le type de données, de fonctionnement, ...).

Le comportement d'un réseau et les possibilités d'application dépendent complètement de ces huit facteurs et le changement d'un seul d'entre eux peut radicalement changer le comportement du réseau neuronal.

Les neurones et les connexions du réseau sont souvent illustrés comme dans la Figure 4-1. En fonction des connexions, les réseaux peuvent être à flux dirigé vers l'avant (en anglais: *feed-forward*) ou récurrents. Dans un réseau de neurones à flux dirigé vers l'avant, les connexions vont toujours des entrées vers les sorties, sans circuit ni boucle. Dans ce type de réseau neuronal, les neurones sont pratiquement toujours regroupés dans des *couches*. Une couche est un concept logique, pour lequel il est difficile de trouver de définition dans la littérature. Une proposition de définition serait la suivante, même s'il existe certainement des exceptions:

Définition: Une couche de neurones dans un réseau neuronal est un ensemble de neurones pour lesquels il est possible de simultanément déterminer leur valeur de sortie.

Trois types de couches sont souvent identifiés - la *couche d'entrée*, la *couche de sortie* et les *couches cachées*. La couche d'entrée reçoit toutes les valeurs d'entrée du réseau neuronal et la couche de sortie émet toutes les valeurs de sortie. Les couches d'entrée et de sortie sont même parfois confondues. Une couche cachée n'a pas de connexions vers l'extérieur et les valeurs d'entrée des neurones de cette couche sont toujours des valeurs de sortie d'autres neurones. Malheureusement il y a encore une grande confusion dans la terminologie du domaine et, notamment pour le réseau à rétro-propagation, la couche d'entrée est souvent omise. Un réseau à deux couches et un réseau à trois couches peuvent parfois signifier la même chose.

Dans le domaine de l'identification, ce sont avant tout les réseaux de neurones à flux dirigé vers l'avant qui sont utilisés. Dans le cadre de ce travail seuls les réseaux de neurones de ce type seront donc traités. Appartiennent à ce type de réseaux de neurones: la carte auto-organisatrice de Kohonen [Kohonen, 1984], le réseau de « counter-propagation » [Hecht-Nielsen, 1987], le « Group Method of Data Handling », ou GMDH [Ivakhnenko, 1968], le réseau à rétro-propagation [Werbos, 1974], [Rumelhart et al., 1986], les réseaux à fonctions radiales (en anglais: *radial basis functions*) développés au Chapitre 4, ... D'autres types de réseaux de neurones sont le réseau de Hopfield, le Brain State in a Box et le « Boltzmann Machine », qui sont décrits respectivement dans [Hopfield, 1982], [Anderson et al., 1977], [Kirkpatrick et al., 1983], [Rumelhart et al., 1986].

3.2.3 Apprentissage automatique par réseaux de neurones

L'apprentissage automatique des réseaux de neurones s'effectue traditionnellement uniquement par modification des poids du réseau. Par contre, le nombre de neurones est défini par l'utilisateur avant l'apprentissage et reste le même jusqu'à la fin de l'apprentissage. Il existe néanmoins un certain nombre d'algorithmes d'apprentissage qui apprennent par ajout et suppression de neurones [Jokinen, 1991], [Främling & Graillot, 1995]. Certains algorithmes combinent les deux approches, comme ceux qui sont décrits dans [Platt, 1991] et [Deco & Ebmeyer, 1993].

Il existe un grand nombre de *règles d'apprentissage*, dont celles de Hebb⁵, du perceptron [Rosenblatt, 1958] et de Widrow (ou de Widrow/Hoff) [Widrow & Hoff, 1960], qui ont d'ailleurs servi de base pour la plupart des règles de modification de poids développées par la suite. La règle d'apprentissage de Widrow est actuellement celle qui est la plus utilisée, elle est notamment utilisée dans le réseau de rétro-propagation. Ceci a provoqué le développement d'un grand nombre de versions améliorées de la règle d'origine, qui ont contribué à la notoriété des réseaux de rétro-propagation.

L'objectif de tout algorithme d'apprentissage est de minimiser une mesure d'erreur. La mesure la plus utilisée pour des problèmes d'approximation de fonctions est probablement celle de l'erreur des moindres carrés (en anglais: *Root Mean Square Error, RMSE*), ce qui revient à la minimisation de l'expression:

$$E = \sum_{i=1}^N (y'_i - y_i)^2 \quad (2)$$

où E est la variable à minimiser, y'_i est la sortie du réseau neuronal pour l'exemple i , y_i est la sortie attendue pour l'exemple i et N est le nombre d'exemples utilisés dans le calcul. N est normalement le nombre d'exemples de l'ensemble d'entraînement ou de l'ensemble de validation.

La minimisation de E se fait par des procédures d'optimisation des paramètres du réseau neuronal. Le principe du réseau de rétro-propagation est d'optimiser les poids du réseau neuronal par une technique de *descente du gradient*. Pour chaque exemple d'entraînement présenté, le gradient de l'erreur est rétro-propagé d'une couche à l'autre en partant de la couche de sortie, ainsi permettant aux poids d'être modifiés. Il est habituellement nécessaire de présenter chaque exemple au réseau de neurones plusieurs fois pour trouver les bons poids. La vitesse d'apprentissage est souvent indiquée en fonction de ce nombre, le nombre de *cycles d'apprentissage*, c'est-à-dire le nombre de passages de l'ensemble d'entraînement.

Les techniques de descente de gradient présentent l'inconvénient de produire des *minima locaux*, où toute tentative de changement des poids du réseau neuronal donne une

⁵ **HEBB, Donald.** *The Organization of Behaviour*. New-York: Wiley, 1949.

augmentation de E , même si les valeurs optimales (le *minimum global*) des poids ne sont pas encore trouvées. La résolution de ce problème constitue l'un des plus grands défis des réseaux de rétro-propagation. Les problèmes dus à l'existence de minima locaux ont donc conduit au développement de réseaux de neurones et d'algorithmes d'apprentissage permettant la modification de la structure interne du réseau neuronal même pendant l'apprentissage. Le réseau neuronal INKA possède cette caractéristique.

4. Description du réseau neuronal développé (INKA)

Le réseau de neurones développé dans le cadre de ce travail s'appelle INKA, qui vient de *Interpolating, Normalising and Kernel Allocating*. La partie *interpolation* de son nom vient du fait qu'il s'agit d'un réseau neuronal qui permet d'améliorer les capacités d'interpolation par rapport aux autres réseaux de neurones étudiés pendant ce travail. L'allocation de noyaux vient du fait que INKA utilise un algorithme d'apprentissage incrémental, qui détermine automatiquement un nombre de neurones cachés proche du nombre optimal. Il permet de résoudre un certain nombre de problèmes non résolus par d'autres réseaux de neurones, notamment un apprentissage avec peu d'exemples, tout en permettant de garder des capacités de généralisation. L'apprentissage est aussi beaucoup plus rapide que pour la plupart d'autres réseaux neuronaux.

Une description plus détaillée de INKA se trouve dans [Främling & Graillot, 1995] ainsi que dans l'Annexe B. Il y a également des comparaisons entre INKA et d'autres réseaux neuronaux, des explications sur l'utilisation des valeurs normalisées dans la couche cachée, des analyses des paramètres du réseau, ... Les comparaisons sont effectuées avec des réseaux de rétro-propagation et avec des réseaux de type RBF (*Radial Basis Function*). Les principes de base de INKA viennent des réseaux neuronaux de type RBF. Ce type de réseau neuronal présente plusieurs avantages par rapport aux réseaux de type rétro-propagation. En plus de la vitesse d'apprentissage, le traitement de minima locaux et une interprétation plus facile de l'information contenue dans le réseau neuronal représentent d'avantages supplémentaires.

4.1 Caractéristiques techniques de INKA

Les réseaux neuronaux de type RBF sont toujours en trois couches. La couche d'entrée sert uniquement à transmettre les valeurs d'entrée (valeurs des critères de sélection) aux neurones de la couche cachée. Le nombre de neurones dans la couche cachée est souvent déterminé par l'algorithme d'apprentissage. La couche de sortie contient le même nombre de neurones que des valeurs de sortie souhaitées.

Chapitre 5. Apprentissage d'une fonction de préférence en trois dimensions

La fonction d'activation de la couche cachée des réseaux RBF est pratiquement toujours une fonction de calcul de distance entre le vecteur \vec{x} et le vecteur \vec{w}_i (le *centroïde* du neurone). INKA fait un calcul de distance Euclidienne, mais d'autres mesures de distance sont possibles:

$$r_i^2 = (\vec{x} - \vec{w}_i)(\vec{x} - \vec{w}_i)^T \quad (3)$$

où r_i^2 est la valeur d'activation et T signifie la transposition du vecteur $\vec{x} - \vec{w}_i$. La fonction de sortie le plus couramment utilisée est la fonction *Gaussienne* (ou de distribution normale). De meilleures performances ont néanmoins été obtenues par INKA avec la fonction *IMQE* (Inverse Multiquadrics Equation) [Hardy, 1971], [Michelli, 1986], [Powell, 1987], [Gorinevsky & Connolly, 1994]:

$$h_i = \left(1 + \frac{r_i^2}{d_i^2}\right)^{-1/2} \quad (4)$$

où d_i^2 est un paramètre qui détermine la « largeur » de la fonction et h_i est la valeur de sortie du neurone caché.

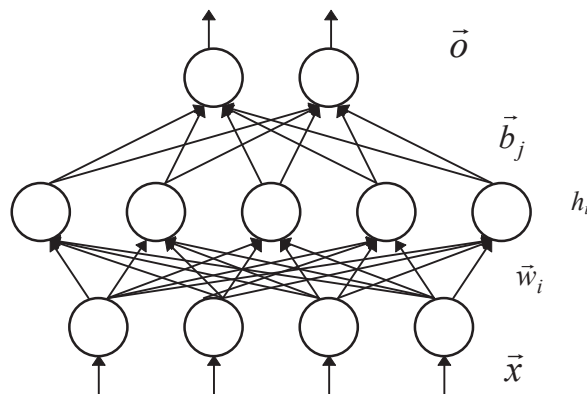


Figure 4-1. Structure du réseau de neurones INKA. \vec{x} est le vecteur d'entrée, \vec{w}_i le vecteur poids du neurone i , h_i est la sortie du neurone caché i , \vec{b}_j est le vecteur poids du neurone de sortie j et \vec{o} est le vecteur de sortie.

La fonction d'activation des neurones de la couche de sortie est normalement une somme pondérée (biaisée ou non⁶) des valeurs h_i . INKA, par contre, normalise d'abord les entrées de chaque neurone de sortie en les divisant par la somme des valeurs de sortie de tous les neurones connectés:

$$hn_{j,i} = h_i / \sum_{k=1}^N h_k \quad (5)$$

où $hn_{j,i}$ est la valeur normalisée de l'entrée i du neurone de sortie j et N est le nombre d'entrées du neurone j . Pour un réseau complètement connecté⁷, ce qui est le plus souvent le cas, $hn_{j,i}$ est le même pour chaque j . La notion hn_i sera donc employée par la suite.

Dans les neurones de sortie, la somme pondérée est ensuite appliquée aux valeurs normalisées:

$$o_j = \sum_{i=1}^N b_{j,i} hn_i \quad (6)$$

où $b_{j,i}$ est le poids du neurone de sortie j associé au neurone caché i . La valeur de sortie est la même que la valeur d'activation.

Dans l'approche de représentation des fonctions de préférence de l'aide à la décision multicritère retenue, il y a une seule sortie du réseau neuronal. Cette valeur de sortie correspond à la note proposée par INKA pour l'action dont les valeurs des critères sont fournies en entrée. La normalisation des valeurs d'entrée de la couche de sortie sert notamment à améliorer l'interpolation de cette fonction de préférence entre les valeurs de sortie connues pour les exemples d'apprentissage. Puisque la distance entre les vecteurs

⁶ Un biais est une entrée supplémentaire ayant une valeur constante de un. La valeur par défaut de la sortie est donc égal au poids de cette entrée, car c'est la valeur de sortie quand la valeur de toutes les autres entrées est égale à zéro.

⁷ Un réseau complètement connecté signifie ici que tous les neurones cachés sont connectés à tous les neurones de sortie. Au cas contraire, uniquement les h_i et h_k des neurones cachés connectés au neurone j sont prises en compte.

d'entrée de ces exemples varie beaucoup, sans la normalisation il serait très difficile de trouver la bonne valeur de d^2 . Un décideur réel n'a ni l'envie, ni les connaissances suffisantes pour faire une telle recherche.

Les critères de sélection utilisent le plus souvent des échelles de valeurs complètement différentes, qui doivent être normalisées pour un traitement par INKA. La normalisation effectuée consiste à ramener les valeurs de chaque critère dans l'intervalle $[0,1]$. La valeur 0 correspond à la valeur minimale et 1 à la valeur maximale de chaque critère pour l'ensemble d'actions potentielles. La valeur normalisée de l'action i pour le critère j est donc:

$$an_{i,j} = \frac{a_{i,j} - \min(C_j)}{\max(C_j) - \min(C_j)} \quad (7)$$

où $an_{i,j}$ est la valeur normalisée de l'action i pour le critère j , $a_{i,j}$ est la valeur de l'action i pour le critère j , $\min(C_j)$ est la valeur minimale du critère j pour l'ensemble d'actions et $\max(C_j)$ est la valeur maximale du critère j pour l'ensemble d'actions. Cette normalisation est utilisée dans tous les calculs présentés.

4.2 Algorithme d'apprentissage

Le principe de l'apprentissage est de partir avec une couche cachée vide, ce qui implique que les valeurs de sortie ont toutes des valeurs égales à zéro. L'entraînement consiste à identifier l'exemple de l'ensemble d'entraînement pour lequel l'erreur entre la valeur de sortie de INKA et la valeur attendue est la plus grande et ajouter un neurone pour traiter cet exemple.

L'apprentissage continue de cette façon jusqu'au moment où la condition d'arrêt, souvent une limite pour l'erreur des moindres carrés, est atteinte. La taille de la couche cachée est complètement déterminée par cet algorithme, sans intervention de la part de l'utilisateur. L'algorithme d'apprentissage est présenté dans la Figure 4-2.

Le calcul des poids de la couche de sortie de l'étape 5 se fait actuellement par une pseudo-inversion de matrice (d'autres possibilités existent⁸):

$$Y = HN * B \quad (8)$$

où Y est la matrice de valeurs de sortie attendues, HN la matrice de valeurs de sortie normalisées des neurones cachés et B la matrice de poids des neurones de sortie, où il y a une ligne par exemple d'apprentissage utilisé. L'utilisation de l'algorithme demande à l'utilisateur de donner des valeurs aux trois paramètres avant de commencer l'apprentissage. Il s'agit des paramètres d^2 , c et la limite pour la condition d'arrêt.

1. Sélectionner les valeurs des paramètres d_i^2 et c (la distance minimale entre deux centroïdes).
2. Initialiser l'ensemble d'exemples d'apprentissage restants R à tout l'ensemble d'entraînement.
3. Prendre l'exemple de R pour lequel la différence entre la sortie donnée par INKA et la sortie attendue est la plus grande, ajouter cet exemple à l'ensemble d'exemples utilisés U et l'enlever de R .
4. S'il n'existe pas encore de neurone caché avec un centroïde trop proche (distance $< c$) alors créer un nouveau neurone caché ayant \bar{w}_i égal à \bar{x} de l'exemple.
5. Recalculer les poids de la couche de sortie pour les exemples dans U .
6. Si la mesure d'erreur globale n'est pas suffisamment petite, revenir au pas 3.

Figure 4-2. Algorithme d'apprentissage de INKA.

Le choix d'une valeur pour d^2 est le plus difficile des trois. C'est aussi celle qui détermine le plus les capacités de INKA à approximer une fonction donnée. L'expérience a heureusement montré qu'il est assez facile de trouver une valeur satisfaisante. La valeur 1 a donc été utilisée dans tous les exemples de ce mémoire avec des résultats satisfaisants. Les essais de l'Annexe

⁸ Une autre méthode qui est souvent utilisée est la descente de gradient, mais cette méthode est plus longue.

Chapitre 5. Apprentissage d'une fonction de préférence en trois dimensions

B montrent aussi que les performances de INKA sont beaucoup moins sensibles à la valeur de ce paramètre que les RBF classiques.

Le paramètre c détermine la distance minimale permise entre les centroïdes des neurones cachés. Une valeur égale à zéro a été utilisée dans tous les calculs de ce document. Les seuls cas où il vaut mieux utiliser d'autres valeurs que zéro sont les cas où la fonction à apprendre est très irrégulière ou lorsque les exemples sont très bruités. De telles situations sont détectées en faisant le calcul (8), où la matrice HN devient singulière et son inversion impossible.

La condition d'arrêt actuellement utilisée est celle de l'erreur des moindres carrés (en anglais: *root mean square error, RMSE*) entre la sortie de INKA et la sortie attendue pour l'ensemble d'entraînement ou, plus souvent, pour l'ensemble de validation. Il est possible d'utiliser d'autres conditions d'arrêt, mais celle-là est celle qui est le plus couramment utilisée. Ce paramètre permet, avec le paramètre d^2 , de jouer sur le nombre de neurones cachés créés et les capacités de généralisation obtenues. Une limite trop grande peut conduire à une approximation trop peu précise. Une limite trop petite peut conduire à une approximation trop précise (*surapprentissage*), où tout le bruit présent dans les exemples est appris aussi, conduisant ainsi à une mauvaise généralisation. La détermination de cette limite est néanmoins facile pour un utilisateur qui connaît à peu près la complexité de la fonction à apprendre et le niveau de bruit dans les exemples d'apprentissage. Il est en général nécessaire de déterminer la valeur d'une telle limite pour pratiquement toutes les méthodes de régression non-linéaire existantes.

L'algorithme d'apprentissage a tendance à créer des neurones cachés pour les exemples d'apprentissage possédant des valeurs de sortie extrêmes en premier. Dans le cadre de l'analyse multicritère, cela signifie les actions ayant les valeurs de préférence les meilleures ou les pires en premier. Ceci implique que les non-linéarités ayant l'amplitude la plus grande sont apprises en premier, jusqu'au moment où il ne reste plus que du bruit. On peut alors dire que les *exemples significatifs* [Hardy, 1971], c'est-à-dire les exemples qui contiennent le plus d'information possible sur la fonction à apprendre, sont pris en compte en premier. Ce type de fonctionnement est très utile, notamment pour effectuer les calculs nécessaires à l'obtention des explications évoquées dans la Partie III.

Malgré cela, l'algorithme d'apprentissage ne donne pas toujours une solution optimale. Un algorithme d'optimisation permet dans ce cas d'améliorer le résultat en éliminant les neurones

cachés pour lesquels la mesure d'erreur globale de INKA n'augmente pas, ou très peu, même s'ils sont supprimés. Cet algorithme est présenté dans l'Annexe B.

Les deux algorithmes ressemblent beaucoup à certains algorithmes connus en statistique. L'algorithme d'apprentissage ressemble fort à l'algorithme *stepwise forward selection* [Boyce et al., 1974]. L'algorithme d'optimisation ressemble au *backward elimination*. C'est la première fois, à notre connaissance, que ces algorithmes sont utilisés pour l'apprentissage des réseaux de neurones. Les algorithmes ont été développés de manière indépendante dans le cadre de ce travail.

5. Apprentissage d'une fonction de préférence en trois dimensions

Pour donner une idée du fonctionnement de l'apprentissage et des capacités de INKA à apprendre des fonctions de préférence contextuelle, telles qu'elles ont été définies au Chapitre 1.3, nous allons d'abord étudier deux cas simples en trois dimensions. Les deux premières dimensions sont celles des deux critères de sélection x_1 et x_2 et la troisième celle de la valeur de préférence n . La fonction de préférence à apprendre est dans le premier cas une fonction linéaire et dans le deuxième cas une fonction non-linéaire.

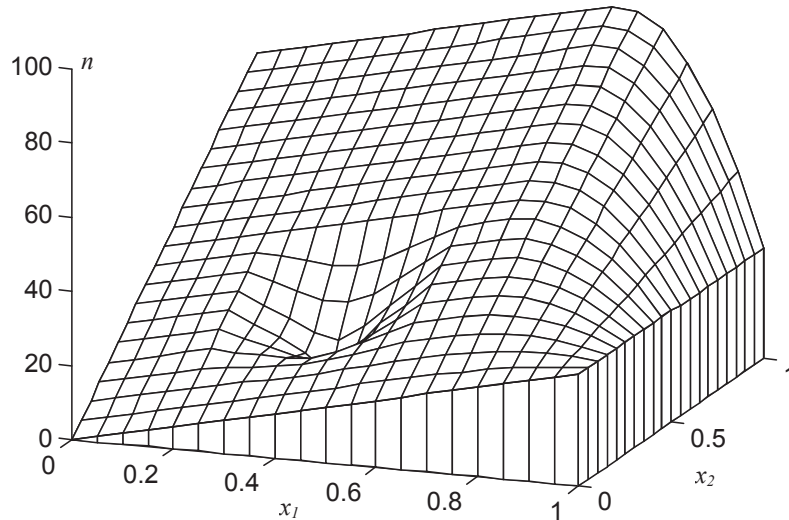


Figure 5-1. Exemple d'une fonction de préférence non-linéaire et continue pour deux critères de sélection, obtenue par une somme pondérée modifiée pour certaines parties.

La première fonction est une somme pondérée, où les poids des deux critères de sélection sont $p_1 = 30$ et $p_2 = 70$. Cette fonction est visualisée dans la Figure 2-2. Pour obtenir une fonction de préférence non-linéaire, la somme pondérée a été modifiée par deux règles, avec la fonction visualisée dans la Figure 5-1 comme résultat. Les deux règles sont:

$$\text{SI } x_1 > x1low \text{ ALORS } p_2 = p_2 - p_2 \times \left(\frac{x_1 - x1low}{1 - x1low} \right)^2$$

$$\text{SI } x_1 > x1med - x1rad \ \& \ x_1 < x1med + x1rad \ \&$$

$$x_2 > x2med - x2rad \ \& \ x_2 < x2med + x2rad \ \text{ALORS}$$

$$p_2 = p_2 - p_2 \times \left(1 - \frac{\sqrt{(x1med - x_1)^2 + (x2med - x_2)^2}}{\sqrt{x2rad^2 + x2rad^2}} \right)$$

où:

- $x1low$ est la limite de la « chute », valeur 0.7,
- $x1med$ est la valeur x_1 pour le fond du « trou », valeur 0.4,
- $x2med$ est la valeur x_2 pour le fond du « trou », valeur 0.3,
- $x1rad$ est l'étendue du « trou » pour x_1 , valeur 0.2,
- $x2rad$ est l'étendue du « trou » pour x_2 , valeur 0.2.

Ces deux règles correspondent à la définition d'une fonction de préférence contextuelle donnée dans le Chapitre 1.3. La première règle correspond plutôt au cas où le dépassement d'une valeur pour un critère change le poids d'un autre. La deuxième règle correspond plutôt à une mauvaise combinaison des valeurs des deux critères, malgré le fait que c'est le poids p_2 qui est modifié pour obtenir le « trou ». Il est très difficile pour un décideur réel d'exprimer de telles fonctions de préférence selon trois dimensions déjà, ce qui justifie bien une approche par apprentissage automatique.

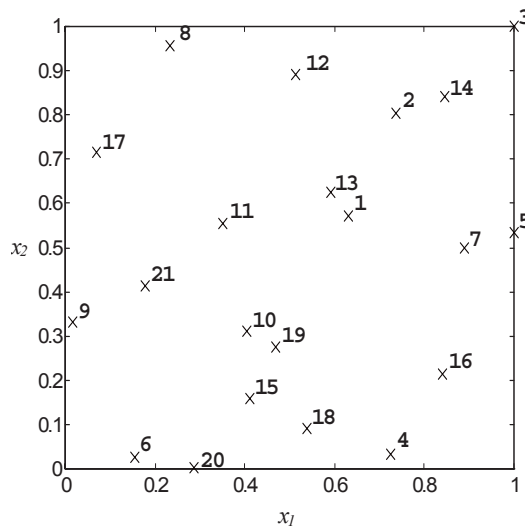


Figure 5-2. Position des 21 exemples dans le plan x_1, x_2 avec leur ordre d'apparition.

Les capacités d'apprentissage de INKA seront d'abord illustrées pour la fonction linéaire. Les exemples d'apprentissage sont choisis aléatoirement dans l'espace x_1, x_2 avec des valeurs de l'intervalle $[0,1]$, sauf le troisième exemple qui était ajouté manuellement pour le couple $(1,1)$. Les couples de valeurs (x_1, x_2) utilisés et leur ordre d'apparition dans l'ensemble d'apprentissage sont indiquées dans la Figure 5-2.

INKA apprend la pente générale de la fonction linéaire dès l'introduction de deux exemples et la fonction apprise est pratiquement parfaite après l'introduction de huit exemples, comme le montre la Figure 5-3. Si les deux exemples et le paramètre d^2 de INKA sont bien choisis, la

Chapitre 5. Apprentissage d'une fonction de préférence en trois dimensions

fonction apprise est directement la bonne, même si la définition d'une surface selon trois dimensions requiert normalement au moins trois exemples. La normalisation des valeurs de sortie de la couche cachée introduit néanmoins par défaut une condition supplémentaire. S'il y a uniquement deux neurones cachés avec les centroïdes \bar{w}_1 et \bar{w}_2 , et si les notes correspondantes sont n_1 et n_2 , alors la note n donnée par INKA pour toute valeur (x_1, x_2) ayant la même distance aux deux centroïdes est la moyenne des valeurs n_1 et n_2 . INKA n'utilise alors pas uniquement les valeurs n absolues, mais aussi l'information différentielle qu'elles donnent.

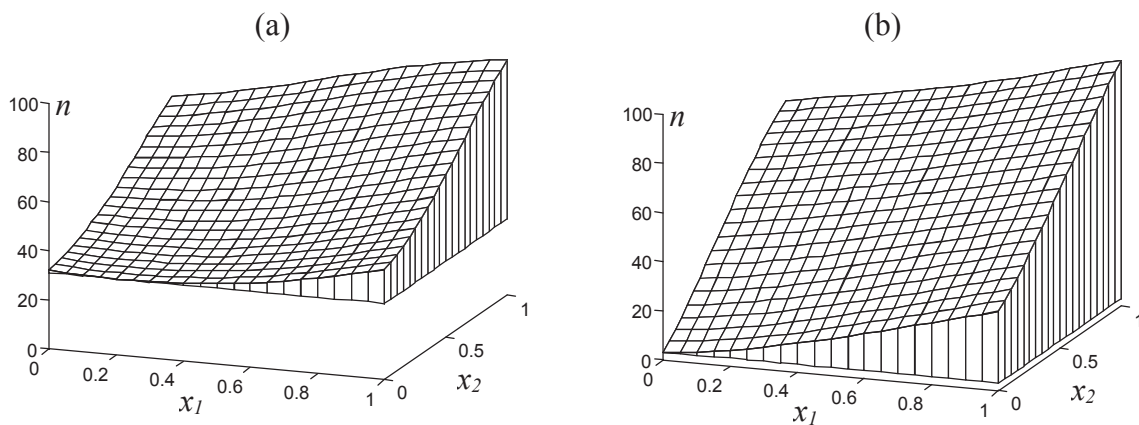


Figure 5-3. Apprentissage de la fonction de préférence linéaire de la Figure 2-2(a) par INKA. (a) Fonction apprise en utilisant uniquement les deux premiers exemples, (b) fonction apprise après l'utilisation des huit premiers exemples.

Dans le cas de la fonction non-linéaire, INKA apprend vite l'allure générale de la fonction, comme illustré dans la Figure 5-4. Il est néanmoins évident qu'il est nécessaire de présenter un nombre d'exemples plus grand que pour la fonction linéaire. La présence du « trou » est impossible à déterminer avant l'introduction d'un exemple qui se situe dans le trou. Une fois que le « trou » est appris par INKA, il est nécessaire d'introduire encore un grand nombre d'exemples pour bien apprendre ses limites, c'est-à-dire les bords du trou.

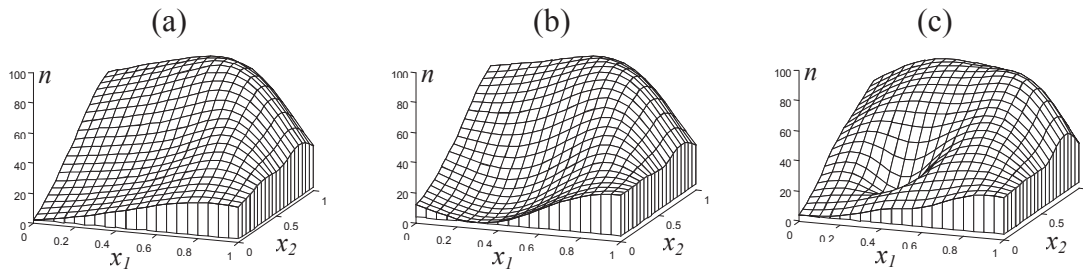


Figure 5-4. Apprentissage de la fonction de préférence non-linéaire de la Figure 5-1 par INKA. Les figures illustrent la fonction apprise (a) après l'introduction de 8 exemples, dont aucun venant du « trou », (b) après l'introduction du premier exemple se situant dans le trou (exemple numéro 10), et (c) le résultat final après 21 exemples.

Les résultats montrent que INKA est capable d'apprendre des fonctions de préférence assez complexes, même si cela demande parfois un grand nombre d'exemples. Le problème du nombre d'exemples sera traité en détail plus loin. Notre objectif étant seulement de déterminer un classement correct parmi les actions potentielles, il n'est pas nécessaire d'avoir une très grande précision d'approximation de la fonction de préférence. La complexité de la fonction utilisée ici est un peu exagérée, les vraies fonctions de préférence des décideurs peuvent être considérées comme étant plus régulières et plus lisses.

6. Sélection et saisie d'exemples d'apprentissage

Pour bien apprendre la fonction de préférence, il est nécessaire d'avoir des bons exemples d'apprentissage. Ceci implique que les exemples d'apprentissage doivent satisfaire les conditions suivantes:

1. Bien couvrir tout l'espace d'entrée représentant des actions potentielles.
2. Avoir des valeurs de préférence à peu près cohérentes, qui correspondent aux préférences du décideur.

La première condition exprime le fait qu'il est difficile de donner un avis sur quelque chose que l'on ignore. Il est, par exemple, difficile de comparer une Rolls Royce et une Mercedes si l'on a connu uniquement des Citroën 2CV et des Renault 4. Par contre, il est possible de donner un avis sur des voitures qui ressemblent celles que l'on connaît.

La deuxième condition est issue du fait que l'information donnée par le décideur doit effectivement correspondre à ses préférences. Il est important d'obtenir un minimum de cohérence et de rationalité de la part du décideur pour éviter qu'il ne représente plusieurs avis contradictoires. Deux méthodes sont ici proposées pour satisfaire à ces deux conditions.

6.1 Sélection des exemples d'apprentissage par carte auto-organisatrice de Kohonen

La *carte auto-organisatrice* de Kohonen [Kohonen, 1984] est un réseau neuronal qui utilise l'apprentissage non supervisé. Comme pratiquement tous les réseaux de neurones de ce type, l'objectif est d'effectuer une *analyse de données*, permettant d'extraire automatiquement de l'information à partir des données brutes. L'analyse de données effectuée est une identification automatique de classes d'objets dans les données. L'analyse de données générale [Blayo et al., 1993] et la reconnaissance de la parole [Leinonen et al., 1991] sont des cas d'application de ce type de réseau de neurones.

La notion de *conservation de la topologie* des cartes auto-organisatrices est essentielle pour choisir de bons exemples d'apprentissage pour apprendre la fonction de préférence. Il faut que les exemples de préférence soient aussi représentatifs que possible de l'ensemble d'actions potentielles. L'ensemble d'entraînement de la carte auto-organisatrice est, pour cette raison, l'ensemble d'actions potentielles. La carte auto-organisatrice essaye toujours de couvrir tous les exemples de l'ensemble d'entraînement aussi bien que possible, et de respecter la notion de conservation de la topologie.

6.1.1 Modèle de connectivité

La carte auto-organisatrice est un réseau neuronal en deux couches, comme illustré dans la Figure 6-1. La première couche est une couche d'entrée, qui sert uniquement à distribuer les valeurs d'entrée aux neurones de la deuxième couche. La deuxième couche est directement la couche de sortie, qui fournit des sorties binaires. Il s'agit d'un réseau de neurones complètement connecté, c'est-à-dire que chaque neurone de la première couche est connecté à chaque neurone de la deuxième couche.

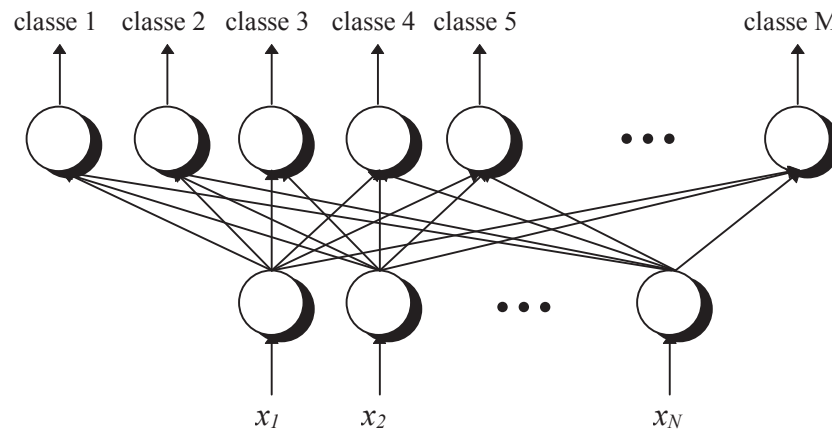


Figure 6-1. Modèle de connectivité de la carte auto-organisatrice.

Quand la carte effectue la classification d'un vecteur de valeurs d'entrée, la sortie est zéro pour tous les neurones de la deuxième couche, sauf pour le neurone correspondant à la classe identifiée qui a une valeur de un. Le neurone ayant la valeur de un est le neurone « gagnant » de la compétition effectuée pour déterminer le neurone i de la deuxième couche ayant le vecteur poids \bar{w}_i le plus « proche » du vecteur d'entrée \bar{x} . Une distance Euclidienne est souvent utilisée comme mesure de distance, mais ce n'est pas toujours le cas. L'effet obtenu par ce type de compétition est une tessellation de Voronoï de l'espace d'entrée, où chaque neurone est le « propriétaire » d'une sous-espace définie par des séparations linéaires.

6.1.2 Apprentissage par compétition

Dans le cas de l'apprentissage non supervisé, les exemples d'apprentissage ont uniquement des valeurs pour les entrées qui sont, dans notre cas, des critères de sélection. Il n'y a aucune information sur la bonne réponse correspondante, c'est-à-dire la valeur de préférence recherchée. Souvent la bonne réponse n'est même pas connue avant l'apprentissage, puisque l'objectif de cette analyse de données est d'aider le décideur à mieux comprendre les données afin qu'il puisse fournir cette réponse plus facilement. On dispose alors, dans ce cas, d'un ensemble d'entraînement, mais pas d'un ensemble de validation.

L'apprentissage du réseau se fait par le principe d'*apprentissage par compétition*, qui signifie que seul(s) le ou les neurone(s) sélectionnés ont le droit de modifier leur poids. Le principe s'exprime alors par la règle d'apprentissage suivante:

$$\Delta \bar{w}_i = \alpha (\bar{x} - \bar{w}_i) \times z_i \quad (9)$$

où:

$\Delta\vec{w}_i$: Vecteur de modification des poids du neurone i .

α : Coefficient d'apprentissage déterminant la vitesse de l'apprentissage.

\vec{x} : Vecteur d'entrée.

\vec{w}_i : Vecteur poids du neurone i .

z_i : Sortie du neurone i , qui est 1 si le neurone est sélectionné, 0 sinon.

L'apprentissage par compétition fonctionne de la même manière que les nuées dynamiques [Bertier & Bourouche, 1975]. Il permet ainsi aux neurones de la deuxième couche d'identifier des classes d'objets. Dans le cas idéal, le vecteur poids de chaque neurone correspond au barycentre d'une concentration de vecteurs (en anglais: *cluster*), qui appartiennent tous à la même classe. Il existe beaucoup de données de ce type, comme les sons vectorisés des lettres dans le domaine de la reconnaissance de la parole [Leinonen et al., 1991], où les vecteurs correspondant à la même lettre se trouvent concentrés à un certain endroit de l'espace d'entrée.

6.1.3 Conservation de la topologie

L'avantage des cartes auto-organisatrices par rapport aux autres méthodes de classification, comme les nuées dynamiques, la classification hiérarchique, le « K plus proches voisins », l'analyse discriminante ou l'analyse factorielle réside dans la notion de *préservation de la topologie* (pour toutes les méthodes d'analyse de données, voir par exemple [Bertier & Bourouche, 1975], [Lebart et al., 1979], [Celeux et al., 1989] ou [Devillers & Karcher, 1991]). Cette préservation de la topologie se fait grâce à la *structure de voisinage* de la deuxième couche d'une carte auto-organisatrice.

Les neurones de la deuxième couche sont numérotés de 1 à M , où M est le nombre de neurones de la deuxième couche. Cette numérotation sert à identifier les classes et permet de définir la notion de *voisinage* entre les neurones. Le voisinage définit la structure logique du réseau. Si la structure logique est linéaire, chaque neurone a deux voisins sauf pour les deux neurones en extrémité. Si la structure logique est un réseau maillé en deux dimensions, chaque neurone a plutôt huit voisins sauf pour les neurones sur les bords et dans les coins, comme indiqué dans la Figure 6-2. Une représentation plus simple par carreau est souvent

utilisée, comme dans la Figure 7-1. L'objectif de cette notion de voisinage est de faire une projection des données brutes, qui sont des vecteurs numériques de \mathfrak{R}^N selon les relations de voisinage pour permettre une visualisation plus compréhensible des données.

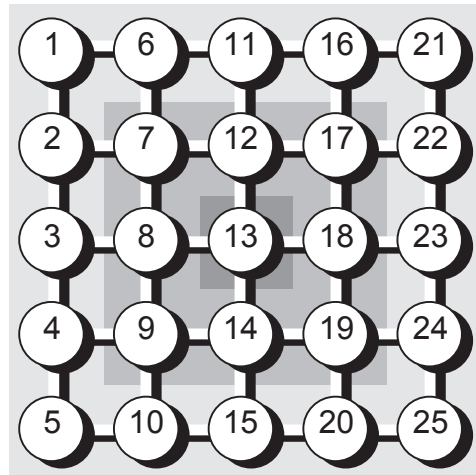


Figure 6-2. Voisinage pour une carte auto-organisatrice de Kohonen avec une structure logique en deux dimensions. Les neurones sont numérotés et les liens de voisinage sont indiqués par les lignes blanches. Les carrés ombragés illustrent la diminution de la zone d'influence du neurone n° 13 pendant l'apprentissage.

Une *zone d'influence* est définie pour chaque neurone qui, dans la plupart de cas, couvre tous les neurones au début de l'apprentissage, ce qui signifie que les poids de tous les neurones sont modifiés pour chaque exemple d'apprentissage. L'effet obtenu est que les poids de chaque neurone se trouvent regroupés au barycentre des exemples d'apprentissage. La taille de la zone d'influence est normalement la même pour tous les neurones. Ensuite, les zones d'influence diminuent en taille pour atteindre zéro après un certain temps, qui est normalement inférieur à la moitié du temps d'entraînement total [Demuth & Beale, 1993]. La diminution successive de la taille des zones d'influence garantit plus ou moins que les neurones voisins seront également proches ou même voisins dans l'espace d'entrée. C'est cet effet que l'on appelle la conservation de topologie et qui se produit quelle que soit la dimension de l'espace d'entrée.

La conservation de la topologie est illustrée par les deux cartes auto-organisatrices présentées dans la Figure 6-3. Dans cette figure, la dimension d'entrée est la même que celle de la structure logique de la carte auto-organisatrice, ce qui permet d'indiquer les coordonnées des exemples d'apprentissage et des poids des neurones dans la même figure. Deux choses sont à retenir de ces deux illustrations:

- La tendance de la carte à couvrir tout le sous-espace d'entrée où il y a des exemples d'apprentissage, mais rien de plus.
- Même si la structure logique de la carte est un réseau maillé, les relations de voisinage sont conservées.

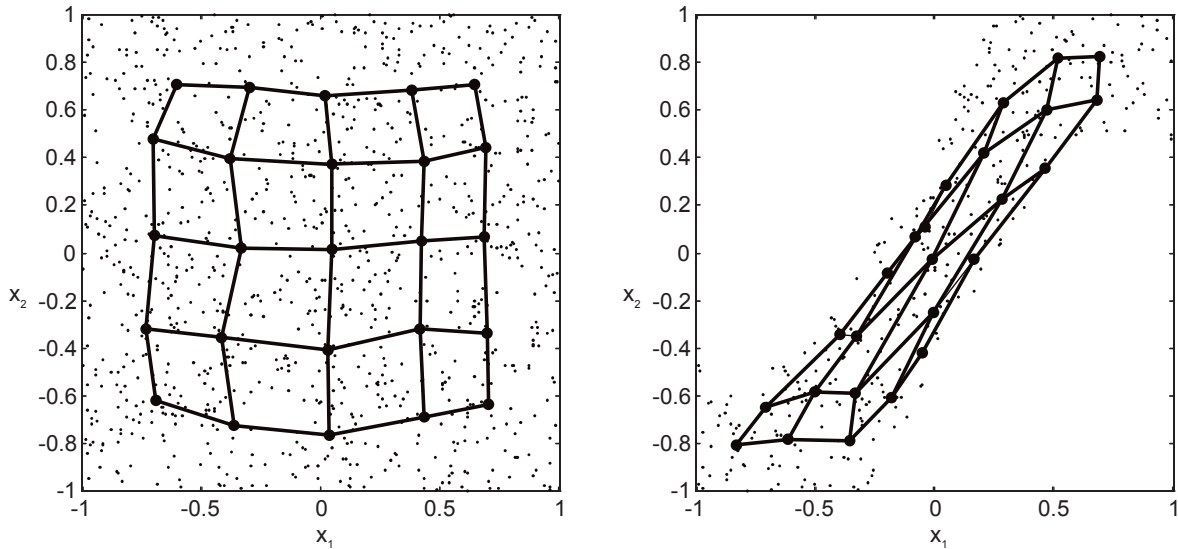


Figure 6-3. Illustrations de l'apprentissage par carte auto-organisatrice dans le cas d'un espace d'entrée en deux dimensions et une structure logique en deux dimensions, ce qui permet d'afficher les exemples d'apprentissage comme des points et les poids des neurones avec leurs connexions de voisinage dans une même figure. L'apprentissage a été arrêté après 2000 cycles d'apprentissage.

Ces deux propriétés sont toutes les deux très importantes pour pouvoir sélectionner les exemples d'apprentissage pour l'identification de la fonction de préférence. Même pour des espaces d'entrée avec des dimensions beaucoup plus grandes, cette démarche garantit de couvrir les combinaisons de valeurs des critères de sélection susceptibles d'exister parmi les actions potentielles. Parmi les méthodes étudiées au cours de ce travail, la carte auto-organisatrice offre les meilleures possibilités.

Pour choisir les exemples d'apprentissage, le décideur doit d'abord retenir des exemples qu'il connaît bien. Ensuite, à l'aide de la carte auto-organisatrice, il vérifie si l'ensemble d'actions potentielles est bien couvert par ses exemples. Si ce n'est pas le cas, le décideur choisit des classes aussi éloignées que possible des classes pour lesquelles il existe déjà un exemple. Pour chaque classe choisie, il retient l'action qu'il connaît le mieux comme exemple. Le choix de

classes peut également se faire en fonction des actions pour obtenir des exemples aussi bons que possible.

6.2 Saisie de préférences

Dans notre cas, les valeurs de préférence sont des notes cardinales. Pour le décideur, il est néanmoins plus facile de donner des notes ordinales, ce qui a déjà été constaté dans [Saaty, 1981] pour les poids des critères de sélection en particulier. Il est même préférable d'utiliser des notes ordinales avec plus de précision, de type « A est largement meilleur que B » ou « A est de peu meilleur que B ». De telles notes sont dans ce document appelées des *notes ordinales quantifiées*. Une méthode pour effectuer le passage d'un type de notes à l'autre est celle proposée dans [Saaty, 1981, 1984], la *méthode de hiérarchisation multicritère* ou MHM, qui est, à l'origine, utilisée pour transformer des poids de critères de sélection. Les notes de voitures présentées dans le Tableau 7-1 sont obtenues par cette méthode à partir de comparaisons binaires avec neuf niveaux de précision, ce qui est le nombre de niveaux le plus couramment utilisé.

MHM est une méthode d'analyse multicritère qui est fondée sur l'utilisation d'arborescences hiérarchiques de concepts pondérés (méronomies, en anglais: *part-whole hierarchies*). Les poids des concepts ayant le même père dans la hiérarchie sont définis les uns par rapport aux autres. Les poids comparatifs obtenus sont ensuite transformés en poids absolus, de façon à garder la somme de ces poids égal à un. La normalisation de la somme des poids permet d'évaluer les actions par la méthode de la somme pondérée pour n'importe quel nombre de niveaux dans la hiérarchie.

Seule la partie acquisition et transformation des poids de la MHM sera utilisée ici. Dans la MHM les critères sont pondérés deux à deux comme dans la Figure 6-4, qui s'inspire de l'interface graphique de l'outil Descriptor [Grand & Juliot, 1990]. Le décideur peut, par ce moyen, saisir de façon interactive des poids ordinaux quantifiés.

Cette méthode d'acquisition de poids est directement applicable à l'acquisition de valeurs de préférence, ce qui est également le cas pour le niveau le plus bas de la MHM, correspondant à l'évaluation de chaque action potentielle pour chaque critère. La saisie de valeurs de préférence ordinales quantifiées globales des actions est possible de la même façon, permettant ainsi d'obtenir des notes pour les exemples d'apprentissage.

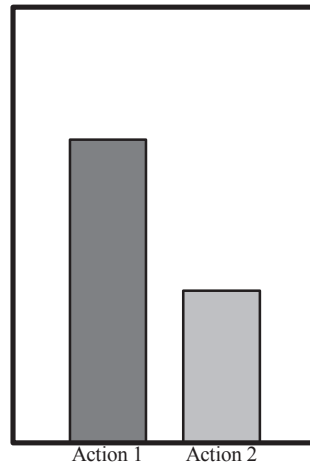


Figure 6-4. Saisie graphique des notes ordinales quantifiées par le principe de MHM. La comparaison signifie que « Action 1 » est deux fois meilleure que « Action 2 ».

Malheureusement, cette méthode devient très lente s'il y a un grand nombre d'actions à comparer. Elle ne donne pas nécessairement des valeurs complètement cohérentes non plus. Il est même possible d'obtenir des situations où la condition de transitivité ($a < b$ & $b < c \Rightarrow a < c$) ne s'applique plus. Si, par exemple, la Citroën XM est meilleure que la Renault 5, qui est meilleure que la Citroën 2CV, il reste possible pour le décideur de dire que la Citroën 2CV soit meilleure que la Citroën XM.

Heureusement la MHM permet de faire la transformation de valeurs ordinales quantifiées en valeurs cardinales même si les valeurs ordinales sont incohérentes, ce qui ne peut jamais être le cas pour les valeurs cardinales. MHM permet également de calculer un indice de cohérence qui exprime la cohérence des valeurs ordinales.

Les incohérences peuvent parfois venir des difficultés du décideur de fournir des valeurs de préférence globales, comme évoqué par exemple dans [Pomerol & Barba-Romero, 1993]. Il est bien connu qu'il y a une limite cognitive chez l'être humain, qui rend difficile la prise en compte de plus de cinq à neuf aspects d'un problème à la fois. La limite même peut varier en fonction de l'individu. Il peut donc être difficile pour le décideur de fournir des valeurs de préférence aux actions caractérisées par un nombre de critères de sélection plus grand que leur limite cognitive.

Dans un grand nombre de situations la limite cognitive n'a pas, ou peu, d'importance. Dans le cas d'un décideur abstrait, les valeurs de préférence sont des données techniques ou statistiques, pour lesquelles il n'y a pas de limite cognitive. Dans le cas du choix de voiture, le

décideur a souvent des expériences pratiques des voitures utilisées comme exemples de préférence. Il peut également avoir une opinion établie depuis longtemps grâce aux autres sources d'information, comme des journaux ou des amis.

Il est également motivé de se demander s'il est vraiment plus difficile pour un décideur de donner des valeurs de préférence aux actions que de donner des poids aux critères de sélection. Une valeur de préférence d'une action est une information « locale » qui concerne uniquement l'action en question. Par contre, le poids d'un critère concerne toutes les actions potentielles, sans exception. Nous retrouvons ainsi le problème de fonctions de préférence contextuelle. Si la fonction de préférence n'est pas contextuelle du tout et si cela est connu par avance, une approche par poids et fonctions d'utilité peut être la meilleure. Dans le cas contraire, il vaut mieux choisir plutôt une approche par exemples de préférence et apprentissage automatique.

Une fois que le nombre d'exemples de préférence est suffisamment grand, il est même possible d'utiliser directement la carte auto-organisatrice comme un SAD très simple. Il suffit d'ajouter une couche supplémentaire avec un seul neurone, qui devient la nouvelle couche de sortie, supprimer les neurones de la deuxième couche pour lesquels il n'y a pas d'exemple de préférence et connecter les neurones restants au neurone de sortie. Après, il suffit d'utiliser les valeurs de préférence des exemples pour initialiser les poids du neurone de sortie pour obtenir un simple réseau de « *counter-propagation* » [Hecht-Nielsen, 1987, 1990]. Ce réseau identifie la classe la plus proche de l'action évaluée et donne comme sortie la valeur de préférence de l'exemple de préférence correspondant. Néanmoins, il faut avoir un grand nombre d'exemples de préférence pour que cette approche donne des résultats comparables avec ceux obtenus par l'utilisation du réseau neuronal INKA.

6.2.1 Analyse de sensibilité et de robustesse

Puisqu'il reste impossible d'être sûr de la rationalité et de la cohérence des valeurs de préférence fournies par le décideur, il est toujours recommandé de faire une analyse de sensibilité et de robustesse pour tout problème de décision réel. Ces concepts sont définis comme suit dans [Maystre et al., 1994]:

Définition 1: L'analyse de sensibilité consiste à répéter l'analyse multicritère originale en faisant varier, isolément puis collectivement, les valeurs attribuées à l'origine aux évaluations des actions, aux seuils et aux poids.

Définition 2: L'analyse de robustesse est une analyse cherchant à déterminer le domaine de variation de certains paramètres dans lequel la sélection d'une action reste la même.

La définition de l'analyse de sensibilité n'est pas directement applicable aux méthodes développées dans ce document puisqu'elles n'utilisent ni des seuils ni des poids (au sens multicritère de ces méthodes). Les paramètres à faire varier sont plutôt les exemples de l'ensemble d'entraînement et les valeurs de préférence de ces exemples. La définition suivante, plus générale, serait plus appropriée:

Définition: L'analyse de sensibilité consiste à répéter l'analyse multicritère originale en faisant varier, isolément puis collectivement, les valeurs attribuées à l'origine à tous les paramètres de l'analyse multicritère ou aux facteurs ayant une influence sur les valeurs de ces paramètres.

L'analyse de sensibilité par rapport aux valeurs de préférence demande simplement de faire varier ces valeurs dans des limites raisonnables, refaire l'apprentissage de INKA, recalculer le classement et étudier l'effet sur le classement. Pour analyser la sensibilité par rapport à l'ensemble d'entraînement, il suffit de supprimer un ou plusieurs exemples d'apprentissage et de refaire l'apprentissage et le classement, comme pour les valeurs de préférence.

L'analyse de robustesse consiste à faire varier les mêmes paramètres que dans l'analyse de sensibilité, mais en étudiant les changements plutôt du côté des actions potentielles pour voir si elles restent constamment à peu près à la même position dans le classement. Si c'est le cas, il est possible de constater que ces actions sont robustes aux changements des paramètres du modèle. S'il y a des actions qui restent constamment en tête du classement, le décideur peut être assez sûr de ne pas se tromper de beaucoup en choisissant une de ces actions.

Les autres paramètres du SAD proposé sont les paramètres d'apprentissage de INKA. Il ne sera pratiquement jamais réaliste de demander à tout décideur ou analyste de comprendre le fonctionnement interne d'un réseau neuronal. Il est donc nécessaire de « cacher » ces paramètres le plus possible. Le nombre de paramètres assez réduit de INKA et la faible sensibilité des résultats aux valeurs de ces paramètres simplifie heureusement le travail. Pour

les applications étudiées, les valeurs par défaut des paramètres ont toujours donné des résultats pratiquement aussi bons qu'avec des valeurs optimales. Grâce à la simplicité et à la robustesse de INKA, il est donc pratiquement toujours possible d'utiliser les valeurs par défaut, ou au moins de les déterminer de façon pratiquement automatique.

L'analyse de sensibilité et de robustesse sera traitée encore plus en détail dans la Partie II, qui développe le SAD présenté dans cette partie vers un SIAD. Ces deux analyses sont beaucoup utilisées, entre autres, pour déterminer la condition d'arrêt du processus interactif. C'est souvent le moment quand la fonction de préférence apprise par INKA offre un niveau de non-sensibilité et de robustesse assez élevé. Avant de traiter l'aspect SIAD, nous étudierons encore le fonctionnement du SAD proposé sur un problème de choix classique, le choix de voiture.

6.2.2 Réduction du nombre de critères de sélection

Acquérir des exemples représentant les préférences du décideur coûte cher, soit au niveau du temps, soit au niveau de prix de consultation ou d'études permettant d'obtenir suffisamment de données. C'est pour cette raison qu'une grande partie de ce document est consacrée à diminuer le nombre d'exemples aussi bien qu'à faciliter leur saisie. Un facteur qui augmente très vite le nombre d'exemples d'entraînement nécessaire est le nombre de critères de sélection utilisés. En général, plus il y a des variables dans la fonction à identifier, plus il faut d'exemples, car plus il y a de variables, plus il y a de paramètres à fixer dans le système d'identification. Ceci est vrai pour les réseaux de neurones aussi bien que pour tout autre système d'identification.

Il est pour cette raison très important d'éliminer des variables (ou, dans notre cas, des critères de sélection) qui n'ont pas d'importance ou qui pourraient être combinés avec d'autres variables sans que l'on ne perde trop d'information. Il existe un certain nombre de méthodes d'analyse de données qui permettent de le faire.

Une méthode qui est très souvent utilisée avant de traiter les données par réseaux de neurones est celle d'*analyse en composantes principales*. Cette méthode permet de substituer les variables d'origine avec d'autres sans perdre trop d'information. Les nouvelles variables peuvent ainsi être moins nombreuses, permettant un apprentissage plus facile.

Le principe de l'analyse en composantes principales est d'identifier des variables dont les valeurs sont très corrélées, ce qui veut dire qu'elles peuvent être remplacées par une seule qui les combine. Dans le domaine de l'aide à la décision il faut néanmoins faire très attention en faisant cette opération. En effet, même si les valeurs de deux critères de sélection sont très corrélées entre elles et avec la valeur de préférence, il est quand même possible que certaines valeurs de ces critères changent radicalement la valeur de préférence. C'est par exemple le cas pour les règles de choix d'une voiture présentées dans le Chapitre 1.3, où le critère « puissance maximale » est très corrélé avec le critère « prix » (et d'autres critères aussi) pour la plupart des valeurs, sauf pour des prix supérieurs à 200.000,- francs français ou des puissances maximales supérieures à 200 chevaux.

Aucun effort n'a été fait pour réduire le nombre de critères de sélection dans les applications présentées dans ce document. Soit il a été décidé de conserver un grand nombre de critères pour mieux montrer les performances des méthodes face à la complexité à traiter, soit la nature des critères n'a pas permis la réduction de leur nombre. Il reste néanmoins très important de ne pas négliger l'analyse préalable des critères de sélection pour établir leur pertinence, quelle que soit la méthode d'aide à la décision utilisée.

7. Exemple d'application: Choix de voiture

Les méthodes présentées seront appliquées à un problème plus complexe et, avant tout, plus réaliste qu'un simple exemple en trois dimensions. Ce problème est le problème classique de choix de voiture, qui permet à un grand nombre de lecteurs de comprendre la problématique et vérifier les résultats. L'application est présentée plus en détail dans l'Annexe A.

7.1 Critères de sélection

Afin de rendre cet exemple aussi réaliste que possible, les critères de sélection ont été choisis sur la base de ceux utilisés dans deux revues d'automobile françaises (« Action Auto Moto » et « l'Auto-journal »). Il est alors facile d'obtenir des données réelles et objectives pour tous les critères, sauf pour les critères « image de marque » et « esthétique », qui nécessitent une saisie de valeurs subjectives. Il est évident que les critères choisis ne sont pas les meilleurs

pour un problème réel, car la plupart des critères sont fortement corrélés. En plus, les deux critères « type de transmission » et « roues motrices » sont des critères pratiquement sans importance. Ce choix de critères a été fait pour rendre le problème plus réaliste au niveau des données utilisées par les vrais décideurs. Les critères de sélection, leurs unités de mesure et les poids utilisés pour la somme pondérée sont:

1. Prix, exprimé en francs français. Poids 12.
2. Puissance maximale, exprimée en chevaux « réels ». Poids 11.
3. Type de transmission, automatique ou manuelle. Poids 6.
4. Roues motrices, les possibilités sont traction, propulsion arrière ou à quatre roues. Poids 5.
5. Dimensions extérieures, en m^3 calculés par la multiplication longueur*largeur*hauteur. Poids 6.
6. Capacité du coffre en dm^3 (litres). Poids 6.
7. Poids en kg. Poids 4.
8. Vitesse maximale en km/h. Poids 10.
9. Parcours 1000 mètres avec départ à l'arrêt, exprimé en secondes. Poids 10.
10. Consommation moyenne en litres par 100 km. Poids 10.
11. Esthétique, information subjective du décideur (l'auteur du document, dans ce cas). Exprimé par une note sur une échelle de 1 à 5. Poids 8.
12. Image de marque, information subjective du décideur (l'auteur du document, dans ce cas). Exprimé par une note sur une échelle de 1 à 4. Poids 5.
13. Équipement en série, note de 0 à 83, calculée à partir d'un sous-ensemble de critères. Poids 10.

Parmi les critères de sélection, il y a des critères quantitatifs aussi bien que des critères qualitatifs. Puisque INKA (et presque tout autre réseau neuronal) peut avoir uniquement des variables d'entrée quantitatives, il est nécessaire de faire certaines transformations des données. Les valeurs des critères quantitatives sont utilisées telle quelle. Par contre, les valeurs des critères qualitatifs sont codées par des chiffres entiers de la même façon que dans StocDech. D'autres types de codage pourraient dans certains cas être plus efficaces [Lorquet, 1992], mais l'approche utilisée est assez simple et universelle, tout en donnant de bons résultats.

7.2 Modèle de préférence

Le modèle du décideur utilisé n'est pas celui d'un vrai décideur car il serait impossible d'obtenir des mesures de performances du réseau neuronal. La fonction de préférence de départ est alors une somme pondérée calculée en utilisant les poids indiqués dans la liste des critères et en utilisant des fonctions d'utilité linéaires. La valeur de préférence obtenue par la somme pondérée est encore modifiée par les trois règles suivantes, qui ajoutent des non-linéarités remarquables à la fonction de préférence:

« SI le prix de la voiture > 200.000,- francs français
ALORS l'importance des autres critères diminue de façon
quadratique pour atteindre zéro à 300.000,- francs
français »

« SI la consommation moyenne > 11L/100km ALORS l'importance
des autres critères diminue de façon quadratique pour
atteindre zéro à 14L/100km »

« SI la puissance maximale dépasse 200 ch alors qu'il n'y a
pas de traction 4x4 ALORS l'utilité de la puissance
maximale est réduite de façon quadratique
proportionnellement à la puissance pour atteindre 30% de
l'utilité normale à 240 ch »

Ce sont les mêmes règles que celles présentées au Chapitre 1.3 avec des précisions au niveau numérique. La fonction obtenue est une fonction non-linéaire en quatorze dimensions (une pour la valeur de préférence calculée et treize pour les critères de sélection). Les valeurs de préférence calculées par cette fonction de préférence « modèle » sont ensuite utilisées pour produire l'ensemble d'entraînement nécessaire pour l'apprentissage supervisé aussi bien que l'ensemble de validation utilisée pour la validation numérique des résultats.

7.3 Sélection et saisie des exemples d'apprentissage

Douze voitures parmi 113 étaient sélectionnées comme ensemble d'entraînement initial à l'aide d'une carte auto-organisatrice. La distribution des voitures parmi les classes identifiées par la carte est indiquée dans la Figure 7-1. Chaque maille correspond à un neurone, qui

correspond à une classe de voitures. Les classes ayant des voitures membre sont indiquées avec un « x ». Les classes qui sont numérotées sont les classes d'où vient un exemple de préférence. Les chiffres indiquent l'ordre de sélection d'exemples.

2	x			x	9	x	x		5
x	x			x	x		x		x
x		6			x	11			
						x			x
		x				x		7	
					4	x	x		12
x		x	x	x			x		x
3			8	x	x				x
		x			x	x	x		
x					x	10	x	x	1

Figure 7-1. Représentation symbolique de la carte de Kohonen utilisée.

Parmi les 100 classes possibles, 53 sont vides. Ceci peut sembler étonnant, car les cartes auto-organisatrices ont plutôt tendance à utiliser toutes les classes disponibles. Le grand nombre de neurones « vides » vient probablement du grand nombre de critères de sélection, qui introduit des vecteurs d'entrée d'une grande dimension. Il en résulte un grand nombre de neurones qui se situent toujours entre les neurones stabilisés en premier (c'est-à-dire ayant identifié des classes), sans jamais arriver à trouver la bonne classe à représenter. Une prolongation de l'apprentissage n'avait pas d'effet sur ce phénomène.

Les voitures choisies et les deux exemples représentant la voiture idéale et anti-idéale constituaient l'ensemble d'entraînement de l'apprentissage supervisé de INKA. Pour ces quatorze voitures, l'auteur appliquait également la méthode d'acquisition de poids de la MHM pour effectuer une comparaison des voitures deux à deux. Même si les comparaisons étaient effectuées tout en gardant en tête la fonction de préférence d'origine, les valeurs de préférence obtenues et indiquées dans le Tableau 7-1 sont assez différentes de celles obtenues avec la somme pondérée modifiée. L'essentiel est néanmoins de garder le même ordre entre les actions de l'ensemble d'entraînement, ce qui est le cas malgré l'incohérence due à la non-rationalité du décideur et le bruit introduit par l'expression des préférences par la MHM.

Les valeurs des critères de sélection de la voiture idéale étaient les valeurs maximales de toutes les voitures pour les critères à maximiser et minimales pour les critères à minimiser.

Les valeurs des critères de la voiture anti-idéale étaient les valeurs minimales de toutes les voitures pour les critères à maximiser et maximales pour les critères à minimiser. Leurs notes sont calculées de la même façon que pour les autres voitures de l'ensemble d'entraînement.

<i>Voiture</i>	<i>Note calculée</i>	<i>Note MHM</i>
Ford Mondeo 2000 Ghia	63.35	34.96
Mazda Xedos 9	64.65	38.88
Rover 220 GSi	60.30	26.79
Peugeot 405 2.0	62.67	32.62
Volvo 850 T5	76.40	73.64
WW Passat VR6	64.64	37.43
Opel Omega CD V6	69.15	56.48
Citroen ZX Aura 1.8 i	48.80	15.82
Citroen XM V6-24	4.93	10.01
SAAB 900 S 2.0-16	62.52	30.48
Audi A6 2.8	58.92	23.67
BMW 520i	68.88	48.20
Voiture idéale	103.00	100.00
Anti-idéal	2.23	10.01

Tableau 7-1. Valeurs de préférence calculées par la somme pondérée modifiée par règles et celles obtenues par la MHM.

7.4 Évaluation des résultats obtenus par INKA

Après l'apprentissage, le réseau neuronal résultant était utilisé pour estimer les valeurs de préférence pour les 113 voitures (l'ensemble d'actions potentielles). Les dix premières voitures des classements obtenus par la fonction de préférence modèle et par les deux réseaux de neurones sont indiqués dans le Tableau 7-2.

Les valeurs des paramètres d'apprentissage de INKA étaient fixées aux valeurs suivantes pour les deux apprentissages:

- Limite pour l'erreur moyenne = 1.18.
- $d^2 = 1$ (constant).
- $c = 0$.

Ce ne sont pas les valeurs optimales, mais les performances obtenues avec des valeurs optimales ne sont pas tellement supérieures. L'expérience a aussi montré que ces valeurs de paramètres donnent des bons résultats pour un grand nombre d'applications. Il est alors inutile de modifier ces valeurs sauf dans des cas exceptionnels, ce qui facilite largement le travail du décideur. Ceci est un avantage de INKA par rapport à la plupart des réseaux de neurones existants.

(a) Classement « correct »		(b) Classement INKA 1		(c) Classement « MHM »	
Opel Vectra V6	77.70	SAAB 900 SE 2.0 Turbo	76.91	Volvo 850 T5	73.64
SAAB 900 SE 2.0 Turbo	76.69	Volvo 850 T5	76.41	BMW 325i	63.98
Volvo 850 T5	76.40	BMW 325i	74.73	SAAB 900 SE 2.0 Turbo	62.19
BMW 325i	74.26	BMW 525i Worldline	71.09	BMW 525i Worldline	58.59
Lancia Thema Turbo 16v LS	73.81	Opel Omega CD 2.0	70.64	Lancia Dedra HF Integrale	57.91
SAAB 900 SE 2.5 V6	72.21	Volvo 850 GLT 2.5	70.63	SAAB 9000 CSE 2.3 Turbo	57.87
Opel Omega MV6	70.52	SAAB 900 SE 2.5 V6	70.51	Peugeot 405 T16	56.50
Volvo 850 GLT 2.5	70.45	BMW 320i Pack	70.34	Opel Omega CD V6	56.48
BMW 525i Worldline	70.32	Opel Vectra V6	70.16	Opel Omega CD 2.0	55.25
Honda Civic 1.6 VTi	69.82	Mercedes C 280	69.83	Opel Omega MV6	54.55

Tableau 7-2. Comparaison des classements entre le classement « correct », le classement obtenu après apprentissage avec l'ensemble d'entraînement dont les valeurs de préférence viennent de la fonction « correcte » et le classement obtenu avec l'ensemble d'entraînement dont les valeurs de préférence sont saisies à l'aide de MHM. Uniquement les dix premières voitures des classements sont incluses. Les voitures en gras étaient présentes dans l'ensemble d'entraînement.

L'apprentissage a été très rapide, moins d'une seconde. Seulement 11 neurones cachés étaient générés pour le premier ensemble d'entraînement, ce qui implique la présence d'exemples redondants. Pour le deuxième ensemble d'entraînement, 14 neurones cachés étaient générés, ce qui impliquerait que les exemples d'apprentissage sont moins redondants et qu'ils contiennent plus d'information utile (ou plus de bruit).

Une comparaison entre le classement (a) et le classement (b) permet de constater que parmi les 10 premières voitures du classement (a), 7 se retrouvent parmi les 10 premières du classement (b). Les permutations entre les voitures ne posent pas vraiment de problèmes puisque le décideur devra de toute façon prendre en compte un certain nombre de voitures en

tête du classement pour prendre sa décision finale (car il s'agit, comme toujours, de l'aide à la décision uniquement).

Des dix voitures en tête du classement (a), c'est l'Opel Omega MV6 pour laquelle il y a l'erreur la plus grande, 104^{ème} au lieu de 7^{ème}. Cette erreur est assez grande et vient du fait qu'il manque un exemple assez proche de cette voiture pour obtenir une bonne valeur de préférence. L'Opel Omega MV6 est, en fait, plutôt associé à la Citroën XM V6-24, qui a une très mauvaise note due à son prix très proche de 300.000,- francs français. La Lancia Thema Turbo 16v LS se trouve 55^{ème} au lieu de 5^{ème} et la Honda Civic 1.6 VTi 30^{ème} au lieu de 10^{ème}. La raison de ces différences reste essentiellement la même - manque d'exemple assez voisin.

Des dix voitures en tête du classement (b), il y a l'Opel Omega CD 2.0 qui se trouve 5^{ème} au lieu de 20^{ème}, la BMW 320i Pack qui se trouve 8^{ème} au lieu de 16^{ème} et la Mercedes C 280 qui se trouve 10^{ème} au lieu de 19^{ème}. Normalement les différences dans ce sens-là ont moins d'importance, car le décideur verra par lui-même que ce ne sont pas les voitures qui lui conviennent.

Pour l'apprentissage effectué avec l'ensemble d'entraînement dont les notes sont obtenues par la méthode MHM, il est intéressant d'étudier les raisons de la création de 14 neurones cachés au lieu de 11 avec le premier ensemble d'entraînement. Ceci vient le plus probablement du fait que MHM amplifie les différences des notes trop proches, ce qui rend plus facile l'extraction de l'information sur l'importance des critères par apprentissage automatique. Ce phénomène devrait, dans la plupart des cas, être favorable pour l'identification de la fonction de préférence. Pour cette application, l'utilisation de MHM ne simplifie pas uniquement la saisie des exemples, mais elle semble en plus faciliter l'apprentissage automatique de la fonction de préférence.

Une comparaison entre le classement (a) et le classement (c)⁹ permet de constater que des 10 premières voitures du classement (a), 5 se retrouvent parmi les 10 premières du classement (c). Ceci est un peu moins bon que pour le classement (b), mais la différence n'est pas très significative. Une grande différence du classement (c) par rapport au classement (b) vient de

⁹ La comparaison est effectuée avec le classement (a) car nous ne connaissons pas les valeurs de préférence « correctes » pour toutes les 113 voitures car leur saisie serait trop difficile.

l'Opel Omega MV6 qui se trouve parmi les 10 premières au lieu d'être complètement en bas du classement. Le classement des deux autres Opel parmi les 10 premiers est certainement lié à ce phénomène, mise à part le fait qu'elles sont très bien placées dans le classement (a) déjà.

Des dix voitures en tête du classement (a), c'est la Honda Civic 1.6 VTi pour laquelle il y a l'erreur la plus grande, 49^{ème} au lieu de 10^{ème}. Cette erreur est plus petite que l'erreur la plus grande du classement (b). La Volvo 850 GLT 2.5 se trouve 42^{ème} au lieu de 8^{ème}, la Lancia Thema Turbo 16v LS 30^{ème} au lieu de 5^{ème}, la SAAB 900 SE 2.5 V6 19^{ème} au lieu de 6^{ème} et la Opel Vectra V6 14^{ème} au lieu de 1^{er}.

Dans l'autre sens, il y a la SAAB 9000 CSE 2.3 Turbo qui se trouve 6^{ème} au lieu de 109^{ème}, la Peugeot 405 T16 qui se trouve 7^{ème} au lieu de 34^{ème}, la Lancia Dedra HF Integrale qui se trouve 5^{ème} au lieu de 21^{ème}, la Opel Omega CD 2.0 qui se trouve 9^{ème} au lieu de 20^{ème} et la Opel Omega CD V6 qui se trouve 8^{ème} au lieu de 14^{ème}. Il faut constater que les erreurs sont plus grandes dans ce sens, mais comme nous l'avons déjà dit, les erreurs dans ce sens sont moins graves.

Ces résultats montrent que les principes de MHM sont applicables pour donner des notes aux actions de l'ensemble d'entraînement. Même s'il y a moins de voitures communes parmi les dix premières, le résultat est au moins aussi bon que celui du classement (b). Il faut également se rappeler que le classement (a) n'est pas le classement correct pour la fonction de préférence représenté par cet ensemble d'entraînement. Les transformations et le bruit introduits pendant la saisie des comparaisons entre les actions de l'ensemble d'entraînement font que la fonction de préférence modèle ne correspond pas vraiment à la « vraie » fonction de préférence du classement (c).

Les classements obtenus montrent qu'il est possible d'identifier des fonctions de préférence de grande dimension par des réseaux de neurones et notamment par INKA. La quantité d'information à fournir par le décideur est largement moindre que celle nécessaire à fournir pour une expression explicite de la fonction de préférence, tout en obtenant pratiquement le même résultat à la fin. Nous pouvons constater que le développement des SAD par les principes présentés offre une nouvelle approche à l'analyse multicritère et des possibilités intéressantes pour le décideur.

8. Conclusion

Cette partie était consacrée au développement des SAD utilisant l'apprentissage automatique, notamment par réseaux de neurones. Les résultats obtenus montrent les possibilités de cette technique. Le réseau de neurones INKA est l'élément principal qui permet cet apprentissage automatique grâce à son potentiel d'apprentissage et de généralisation même avec un petit nombre d'exemples. Même si INKA est développé pour correspondre aux besoins de l'analyse multicritère, il constitue un réseau de neurones d'intérêt général. Il est utilisable dans d'autres domaines d'application présentant des caractéristiques similaires à celles de l'analyse multicritère.

Cependant cette nouvelle approche demande d'offrir des outils qui guident le décideur dans la sélection des exemples à fournir et qui lui permettent d'exprimer ses préférences d'une façon aussi simple que possible. La nature des paramètres à définir par le décideur nécessite également une adaptation des procédures d'analyse de sensibilité et de robustesse. Les méthodes et les concepts présentés ici s'intègrent ainsi dans une méthodologie générale pour la construction des SAD utilisant l'apprentissage automatique par réseaux de neurones.

Des nouvelles méthodes ou techniques développées dans le cadre de ce travail et présentées dans cette partie sont:

- La modélisation de fonctions de préférence contextuelle, monotones ou non.
- Un nouveau réseau de neurones « INKA » qui permet un apprentissage avec peu d'exemples d'apprentissage, tout en gardant de bonnes caractéristiques de généralisation.
- Une méthode permettant de guider le décideur dans le choix d'exemples d'apprentissage qui permet de couvrir au mieux l'ensemble des actions potentielles.
- Analyse de sensibilité et analyse de robustesse applicables aux SAD neuronaux et facilité par le petit nombre de paramètres dans INKA.

L'aspect interaction avec le décideur demeure cependant à approfondir, ce qui permettrait à celui-ci d'analyser et de modifier ses préférences au cours du travail. La partie suivante de ce document présente une méthodologie de SIAD fondée sur les idées développées dans cette partie, mais permettant de modifier la fonction de préférence de façon interactive.

Partie II

Méthodologie interactive d'aide à la décision utilisant des réseaux de neurones

Les décideurs ont tendances à modifier leurs préférences au cours du processus de décision [Pomerol & Barba-Romero, 1993]. Les Systèmes Interactifs d'Aide à la Décision (SIAD) sont des outils qui ont été développés pour prendre en compte cette modification successive de préférences. Ils offrent souvent des interfaces utilisateur très évoluées qui permettent de visualiser les actions potentielles de plusieurs façons et de facilement modifier les modèles de préférence.

L'objectif de cette partie est de montrer que la méthodologie de SIAD neuronale développée dans le cadre de ce travail permet de traiter des problèmes de décision multicritère à l'échelle réelle. L'objectif est également d'élargir le domaine d'application des SIAD aux problèmes pour lesquels le décideur est représenté par toute une population, l'environnement ou un système social. La particularité de ce type de décideurs abstraits est qu'il n'est pas possible d'interagir directement avec eux pour établir un modèle de préférence. Le seul moyen de le faire est par des sondages, des enquêtes ou d'autres moyens indirects, d'où il faut extraire un modèle de préférence.

L'apprentissage automatique par réseaux de neurones est un des moyens pour le faire. L'introduction du concept de *pseudo-exemples* permet de corriger la fonction de préférence apprise de façon interactive même dans le cas de décideurs abstraits. Dans ce cas, l'interaction s'effectue par l'intermédiaire d'un *homme d'études* ayant des connaissances suffisantes pour effectuer des corrections simples et plus ou moins évidentes à la fonction de préférence apprise.

Le Chapitre 1 donne une présentation de l'état de l'art des SIAD pour situer la méthodologie neuronale dans un contexte général et pour expliquer le choix de cette technique. Le Chapitre 1 présente les composants et les idées générales de la méthodologie neuronale développée. Les Chapitres 2, 4 et 13 présentent les composants de la méthodologie en détail. Dans le Chapitre 14 sont décrites les méthodes d'analyse de sensibilité et de robustesse. La description complète de la méthodologie est ensuite, dans le Chapitre 15, suivie par l'application au problème de choix de sites de stockage de déchets ultimes dans la région Rhône-Alpes. Cette Partie II se termine par la conclusion au Chapitre 16.

1. Introduction aux systèmes interactifs d'aide à la décision

Vincke [Vincke, 1989], définit une méthode interactive d'aide à la décision multicritère comme suit:

Définition: Une méthode interactive consiste en une alternance d'étapes de calculs et d'étapes de dialogue avec le décideur. La première étape de calculs fournit une première solution. Celle-ci est présentée au décideur qui réagit en apportant des informations supplémentaires sur ses préférences (étape de dialogue). Cette information est injectée dans le modèle utilisé et permet de construire une nouvelle solution.

Le dialogue signifie dans ce cas une contribution active du décideur à la construction de la solution au lieu de définir a priori les paramètres du calcul utilisé, comme c'est le cas dans les SAD. Le processus utilisé est un processus interactif où le SIAD présente à chaque étape ses recommandations au décideur, qui estime si ce résultat lui est satisfaisant ou non à la base des informations fournies par le SIAD. Dans le cas où il n'est pas satisfait, il peut continuer à

introduire de l'information sur ses préférences pour trouver des solutions plus satisfaisantes. Cette procédure peut être illustrée par le schéma emprunté à [Dong, 1992] de la Figure 1-1.

L'objectif d'une telle méthodologie interactive n'est pas d'identifier un modèle de préférence pré-existant dans la tête du décideur. L'interactivité permet également au décideur de mieux comprendre ses préférences concernant le problème traité et de les modifier en fonction d'une compréhension progressivement améliorée. Cette approche facilite en même temps l'acceptation des recommandations du SIAD, puisque le décideur peut continuer le dialogue jusqu'à ce qu'il soit convaincu que le modèle de préférence est bien le sien.

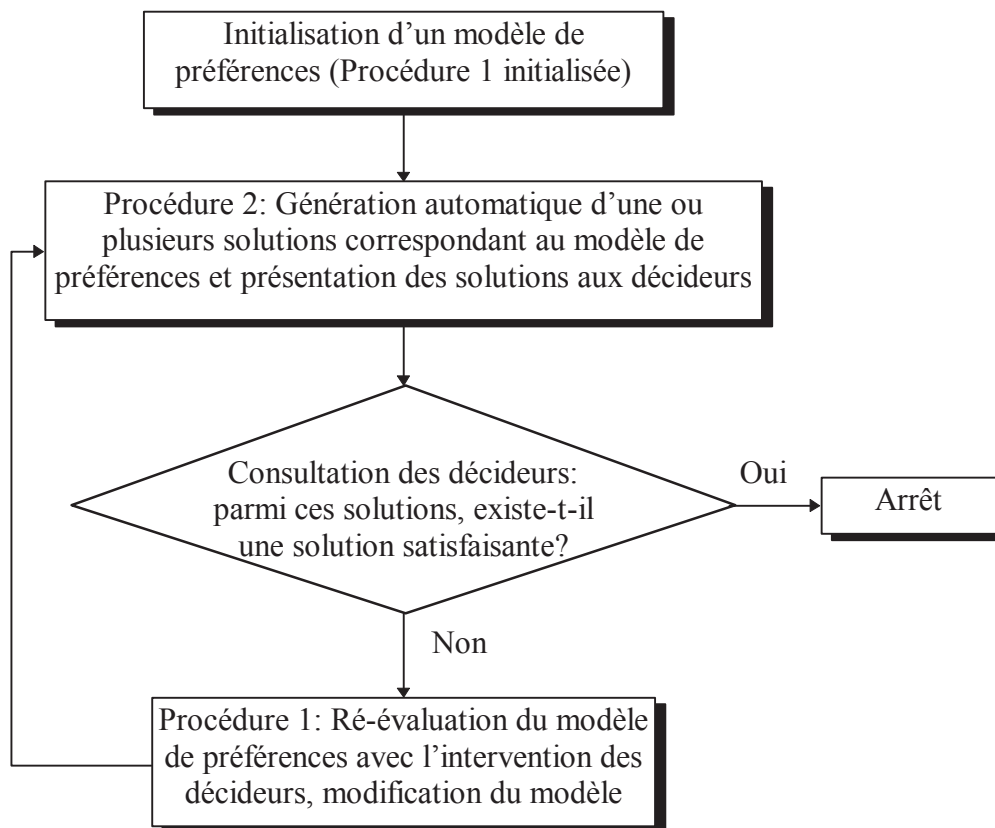


Figure 1-1. Algorithme général d'un SIAD [Dong, 1992].

Les modifications dans les préférences du décideur font partie de la non-rationalité générale des décideurs, qui est un aspect bien évoqué dans [Simon, 1955] et davantage approfondi dans [Simon, 1983]. Pour cette raison il est impossible ou non souhaitable d'essayer de déterminer automatiquement si une recommandation est vraiment satisfaisante pour le décideur ou non. Simon constate que seul le décideur lui-même peut déterminer son niveau de satisfaction concernant une recommandation. C'est pour cette raison qu'une approche d'optimisation classique n'est pas appropriée pour résoudre les problèmes de décision multicritère. Les SAD

présentés en Partie I souffrent des mêmes problèmes, ce qui nécessite une approche interactive.

1.1 Méthodes interactives de recherche de la solution

L'algorithme interactif général présenté en Figure 1-1 a été appliqué de plusieurs façons. Il est inutile de traiter toutes les méthodes correspondantes ici car elles sont très bien décrites dans les ouvrages [Vincke, 1989], [Roy & Bouyssou, 1993] et [Pomerol & Barba-Romero, 1993].

Les méthodes les plus anciennes utilisent des *actions de compromis* qui sont présentées au décideur: STEM [Benayoun et al., 1971], la méthode de Geoffrion, Dyer et Feinberg [Geoffrion et al., 1972], la méthode du point de mire [Roy, 1976] et la méthode de Vincke [Vincke, 1976]. Le choix de la première action de compromis dépend de la méthode. Le décideur indique ensuite dans quel sens il voudrait améliorer cette action de compromis. Il peut le faire en modifiant les paramètres du modèle, comme ses limites maximales de concession sur les critères ou les taux de substitutions entre critères. Les nouvelles actions de compromis sont ensuite déterminées par des méthodes de programmation linéaire, où les contraintes implicitement introduites par le décideur permettent de définir un problème d'optimisation. Un inconvénient majeur avec ce type de méthodes réside dans le fait que les informations nécessaires sont souvent difficiles à obtenir du décideur.

La méthode de Zionts et Wallenius [Zionts & Wallenius, 1976], [Zionts, 1981], appartient elle aussi à cette catégorie. Elle utilise une fonction de préférence additive et linéaire pour choisir une première action de compromis. Dans cette méthode, ce sont les poids des critères qui sont modifiés. Un certain nombre de variations des poids est proposé au décideur, qui les accepte ou les rejette. Les poids modifiés servent ensuite à calculer une nouvelle action de compromis jusqu'au moment où le décideur accepte l'action de compromis comme solution. Malgré le grand nombre de problèmes lié à la modification directe des poids, cette méthode a intéressé un grand nombre de chercheurs qui ont essayé de l'améliorer.

La méthode du point de référence [Wierzbicki, 1980, 1982] utilise une fonction scalarisante contrôlée par des « *niveaux d'aspiration* » sur les critères (définis par les points de référence). La fonction scalarisante est définie par les niveaux d'aspiration et un vecteur de pondération fourni par le décideur. Les niveaux d'aspiration de chaque critère sont modifiés par le décideur à chaque itération. Plus le niveau d'aspiration d'un critère est éloigné de la valeur

actuelle, plus la valeur actuelle devrait changer. Les niveaux d'aspiration sont souvent plus compréhensibles et plus faciles à fixer pour le décideur que les poids et les fonctions d'utilité, ce qui constitue l'avantage principal de cette méthode.

Une autre méthode qui utilise des niveaux d'aspiration un peu de la même façon est celle de Korhonen et Laakso [Korhonen & Laakso, 1986], mais avec des méthodes d'interaction avec le décideur qui sont plus développées. La méthode de Steuer et Choo [Steuer & Choo, 1983] utilise, elle aussi, une notion de distance, mais les changements des poids sont déterminés à partir d'une nouvelle action de référence choisie à chaque étape par le décideur à partir d'un ensemble d'actions proposées par le système. La méthode de Vanderpooten [Vanderpooten, 1989] et la méthode LBS [Jaszkiewicz & Slowinski, 1994] peuvent être considérées être du même type.

1.2 Méthodes pour l'identification interactive d'un modèle de préférence

Les méthodes qui utilisent des actions de compromis pour avancer dans la recherche de solutions utilisent un modèle de préférence local autour de l'action de référence. Ce modèle de préférence peut être contextuel, puisqu'il dépend de l'action étudiée, mais il n'est connu que pour ce contexte. Quand un SIAD essaie d'identifier un modèle (ou fonction) de préférence global (valable pour toutes les actions potentielles), il s'agit plutôt de l'apprentissage automatique au sens donné par le domaine de l'intelligence artificielle. Néanmoins, toutes les méthodes développées autour de ce principe ne sont pas des méthodes interactives. Certaines d'entre elles utilisent des règles comme dans les systèmes experts et d'autres des représentations mathématiques.

La méthode de Jacquet-Lagrèze, Meziani et Slowinski [Jacquet-Lagrèze et al., 1987] est une extension de la méthode PREFCALC [Jacquet-Lagrèze, 1984] qui permet d'identifier une fonction de préférence additive à partir d'exemples de décision. Le décideur donne d'abord son ordre de préférence pour un sous-ensemble d'actions non-potentielles générées par la méthode. L'ordre est utilisé pour estimer des fonctions d'utilité marginales linéaires par morceaux par un programme linéaire. L'interactivité de la méthode vient de la possibilité de redéfinir les contraintes et les paramètres du programme linéaire (poids, nombre de morceaux à utiliser, ...) à chaque itération pour améliorer la fonction de préférence.

L'utilisation de fonctions linéaires par morceaux permet d'introduire des non-linéarités dans les fonctions d'utilité de chaque critère. Malgré l'aspect non-linéaire, il ne s'agit pas d'une fonction de préférence contextuelle dans le sens indiqué dans le Chapitre 1.3 car les importances et les fonctions d'utilité ne changent pas en fonction de l'action traitée. Il ne s'agit pas vraiment de l'apprentissage automatique, puisque le décideur fournit la plupart des paramètres du modèle. Un autre inconvénient de cette méthode est que la construction interactive de la fonction de préférence se fait seulement pour l'ensemble d'entraînement. Le décideur n'a donc pas de possibilité de réagir face à la solution de compromis finale.

Une autre approche de SIAD est celle des systèmes experts tels qu'ils sont définis dans le domaine de l'intelligence artificielle. La méthode PRIAM, *Programme utilisant l'Intelligence Artificielle en Multicritère* [Lévine & Pomerol, 1986], utilise une base de règles de production comme modèle de préférence. Ils constatent les mêmes problèmes que ceux évoqués dans le Chapitre 2.1, qui sont la non-continuité du modèle de préférence et le grand nombre de règles nécessaires [Lévine & Pomerol, 1989].

Il y a également une tendance de plus en plus forte à proposer des interfaces et des représentations graphiques des préférences. Tel est le cas de la méthode GAIA, *Geometrical analysis of Interactive Assistance* [Mareschal & Brans, 1988], qui permet de visualiser les poids et les actions sur un plan donné par la méthode d'analyse en composantes principales. Les représentations graphiques peuvent aussi permettre de mieux visualiser les actions potentielles. C'est le cas des maisons plus ou moins déformées [Korhonen, 1991] et des toiles d'araignée [Kasanen et al., 1991]. Cette tendance ne peut pas être négligée dans les SIAD modernes, qui peuvent profiter du développement rapide des interfaces utilisateur en informatique.

Le choix des réseaux de neurones comme technique vient de leur capacité d'apprentissage automatique de fonctions mathématiques continues, comme les fonctions de préférence contextuelle. Le chapitre suivant illustrera comment ils s'intègrent dans un SIAD, où l'aspect mathématique présente des avantages pour les représentations graphiques et pour l'interaction avec le décideur.

2. SIAD utilisant l'apprentissage automatique par réseaux de neurones

La méthodologie développée ici permet de construire un SIAD utilisant les réseaux de neurones, et particulièrement le réseau neuronal INKA. La Figure 2-1 illustre cette méthodologie, qui est une modification de l'algorithme général de la Figure 1-1. Les étapes un à quatre ont déjà été traitées dans la Partie I. Il nous reste à étudier les étapes cinq à huit.

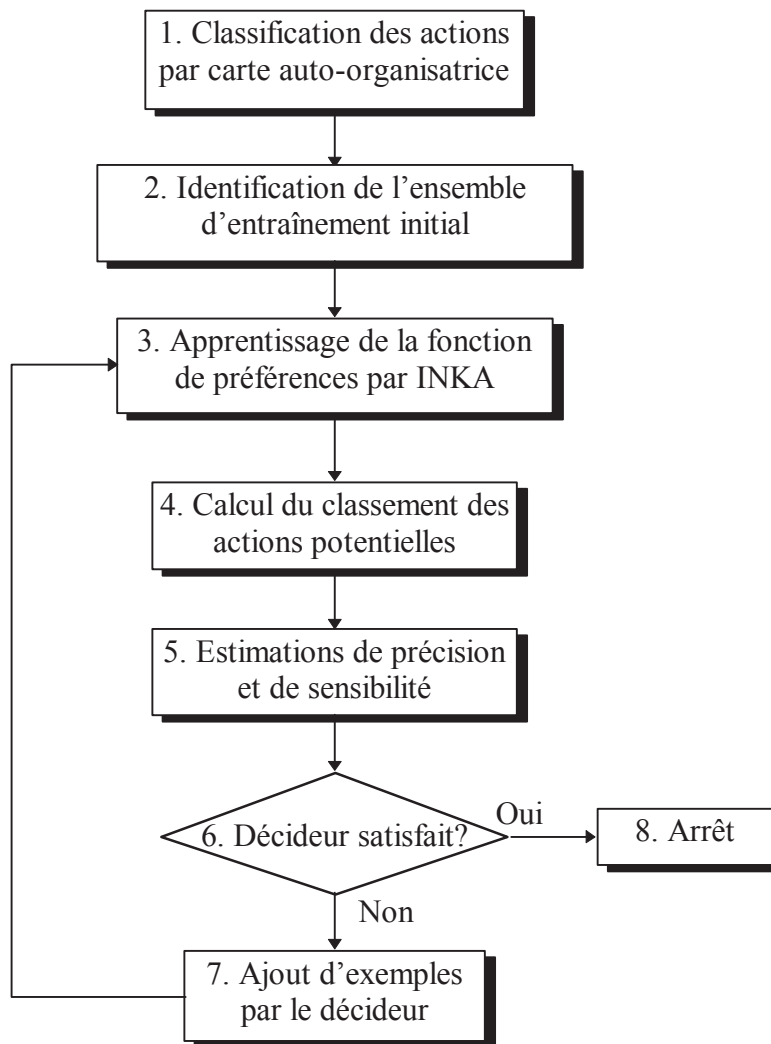


Figure 2-1. Méthodologie interactive d'aide à la décision intégrant l'approche neuronale.

L'objectif des étapes un à quatre est de permettre l'apprentissage d'une fonction de préférence initiale. Les résultats obtenus pour le choix de voiture dans la Partie I montrent qu'il est possible de le faire avec une assez grande précision à partir d'un nombre d'exemples

relativement limité, du même ordre que le nombre de critères de sélection. Pour des fonctions de préférence plus difficiles, ce nombre devient insuffisant. Dans le cas d'un réseau neuronal, les exemples constituent notre seul moyen d'interaction avec le décideur et l'amélioration des résultats dépendra du nombre d'exemples traités. Or, comme nous l'avons déjà constaté, la saisie d'exemples devient vite trop difficile quand le nombre d'exemples dépasse la dizaine.

Le contenu de l'étape n° 7 constitue une nouvelle occasion de permettre au décideur d'introduire ses préférences. Ceci s'effectue à l'aide de « *pseudo-exemples* », qui ne sont pas des actions potentielles ou même réalistes, mais qui servent à mieux couvrir l'espace d'entrée défini par les critères de sélection. Les pseudo-exemples sont introduits par le décideur d'une façon graphique qui lui permet de modifier la fonction de préférence de façon plus directe que pour les exemples « réels ».

La méthodologie permet au décideur de poursuivre l'introduction d'information sur ses préférences jusqu'au moment où il juge le résultat satisfaisant. Pour faciliter au décideur l'estimation de sa satisfaction à l'étape n° 6, il est nécessaire au SIAD d'estimer la précision et la sensibilité de la fonction de préférence apprise. Ces estimations font l'objet de l'étape n° 5.

Les éléments de la méthodologie ont été développés avec l'outil de calcul matriciel « Matlab ». Tous les résultats (classements, valeurs de préférences, indicateurs, graphes) ont été produits par ce programme. Le programme consiste d'à peu près 250 fichiers macro de Matlab, qui peuvent servir pour un futur logiciel d'aide à la décision complet.

2.1 Détermination du nombre d'exemples optimal à l'aide de modèles coût / bénéfice

L'obtention d'exemples d'apprentissage est une procédure très lourde car elle demande beaucoup de réflexion et de travail de la part du décideur. Dans certains cas, l'obtention d'exemples représente un coût. Ceci est particulièrement vrai dans les cas où les données sont obtenues par sondages ou par d'autres études techniques. Par contre, il est beaucoup plus difficile d'estimer la valeur ajoutée liée à l'introduction d'un exemple supplémentaire. Ce problème est étudié depuis longtemps dans le domaine de la gestion d'entreprise, où des méthodes d'aide à la décision de type coût / bénéfice sont fréquemment utilisées [Schlaifer, 1959, 1969].

S'il est effectivement possible d'estimer la valeur ajoutée d'une meilleure précision du modèle de préférence, il devient possible de déterminer un nombre d'exemples optimal comme indiqué dans la Figure 2-2. Le coût d'obtention d'exemples reste normalement constant, ce qui implique une augmentation linéaire du coût en fonction du nombre d'exemples. Par contre, lorsque le nombre d'exemples augmente trop, l'information (et le bénéfice) apportée par chaque exemple diminue.

La détermination du nombre optimal se traduit par la définition d'une *règle d'arrêt* (en anglais: *stopping rule*), qui est très facile à déterminer dans le cas de la Figure 2-2 car il suffit de maximiser le bénéfice, qui est égal à la valeur ajoutée moins le coût. En pratique, la définition de la règle d'arrêt est assez difficile. Ce problème est discuté par certains auteurs spécialisés en statistique et en théorie de la décision, comme [Govindarajulu, 1987].

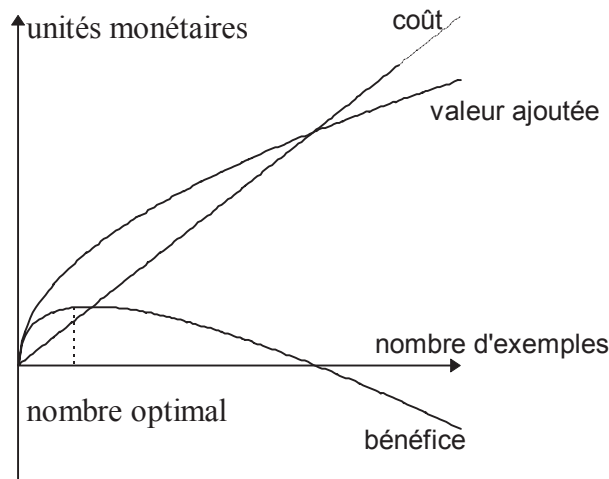


Figure 2-2. Evolution de la valeur ajoutée par rapport au coût d'obtention des données [Schlaifer, 1959]. Le bénéfice est calculé comme la valeur ajoutée moins le coût.

2.2 Détermination du nombre optimal d'exemples de préférence pour l'apprentissage neuronal

Dans le cas d'apprentissage de fonctions de préférence par réseaux de neurones, il est difficile pour le décideur d'estimer si le résultat est suffisamment bon, stable et complet. Il est alors nécessaire d'introduire des indicateurs facilement compréhensibles pour permettre au décideur d'estimer si la valeur ajoutée obtenue par l'introduction de nouveaux exemples est compatible avec leur coût d'acquisition ou s'il doit se contenter du résultat actuel. Ces indicateurs sont utiles au décideur pour traduire les risques de se tromper ou le coût d'avoir

choisi une action qui n'est pas optimale pour résoudre le problème. Afin de faciliter cette tâche, les indicateurs de performance proposés sont les suivants:

Classement de l'ensemble d'actions potentielles donné par INKA (Chapitre 7).

Représentation graphique de la fonction de préférence apprise (Chapitre 2).

Évolution des indicateurs de précision sur la fonction de préférence apprise et les capacités de généralisation obtenues en fonction du nombre d'exemples (Chapitre 13).

Évolution de la sensibilité de la fonction de préférence apprise et de la robustesse du classement (Chapitre 14).

Ratio (nombre de neurones cachés créés pendant l'apprentissage) / (nombre d'exemples dans l'ensemble d'entraînement), indiquant le degré de redondance des exemples d'apprentissage.

Ces indicateurs doivent être utilisés de manière complémentaire. En ce qui concerne le dernier indicateur, la question suivante pourrait être posée: pourquoi ne pas utiliser un neurone par exemple pour obtenir la meilleure précision possible? Ceci est vrai pour des exemples bien distincts et obtenus pour une fonction de préférence parfaite. En pratique les exemples représentent presque toujours une fonction plus ou moins bruitée, à laquelle il est nécessaire d'appliquer les principes de régression. Un apprentissage qui modélise le bruit perd vite ses capacités de généralisation, à cause du phénomène de *surapprentissage*, problème bien connu de l'apprentissage automatique par réseaux de neurones.

Pour augmenter le bénéfice maximal, il est également possible d'essayer de diminuer le coût et d'augmenter la valeur ajoutée par exemple. La diminution du coût correspond dans notre cas à une saisie des exemples rendue plus facile par l'introduction des pseudo-exemples. L'augmentation de la valeur ajoutée a déjà été partiellement abordée dans le cadre de la sélection d'exemples par carte auto-organisatrice, qui permet au décideur d'utiliser des exemples qui soient aussi significatifs que possible. L'objectif est le même pour les pseudo-exemples, même si les moyens sont différents. L'analyse graphique de la fonction de préférence apprise est un de ces moyens.

3. Analyse de la fonction de préférence apprise

Le choix d'utiliser une fonction mathématique comme modèle de préférence permet la visualisation graphique de la fonction de préférence. Il devient néanmoins vite difficile de visualiser des fonctions en plus de deux dimensions, ce qui limite le traitement à un critère de sélection à la fois, le premier axe étant la valeur du critère et le deuxième la valeur de préférence. Nous étudions donc la variation de la valeur de préférence en fonction de la valeur d'un critère de sélection, les valeurs des autres critères de sélection restant constantes. Il serait également possible de visualiser la fonction en trois dimensions, mais cela devient vite peu compréhensible pour le décideur.

3.1 Analyse graphique de la fonction de préférence

La Figure 3-1 illustre le principe de visualisation d'une fonction de préférence non-linéaire comme celle du Chapitre 5. Cette fonction de préférence n'a que deux critères de sélection, ce qui permet une visualisation facile de la fonction en trois dimensions. Pour une étude en deux dimensions, où l'effet d'un seul critère de sélection est étudié, il est nécessaire de fixer les valeurs de tous les autres critères de sélection et de faire varier uniquement la valeur du critère étudié. Dans la Figure 3-1, la fonction de préférence est étudiée en fonction du critère x_1 pour l'action 10, ce qui donne une valeur de 0.31 au critère x_2 .

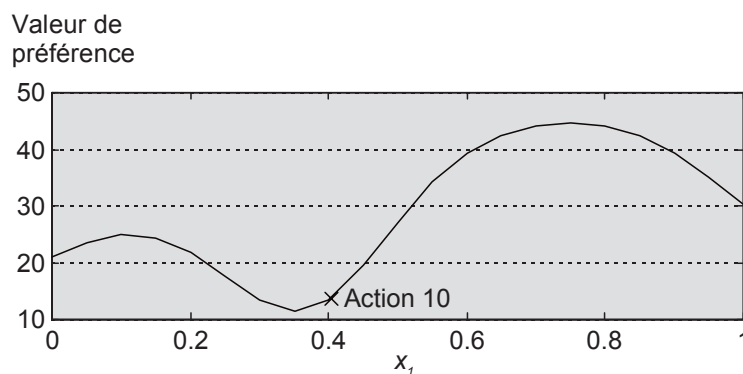


Figure 3-1. Évolution de la valeur de préférence pour l'action 10 en fonction du critère x_1 dans le cas de la fonction de préférence de la Figure 2-2. L'action 10 se trouve presque au milieu du « trou », indiqué par la croix représentant la valeur actuelle pour le critère x_1 .

Pour que la visualisation ait un sens, il est nécessaire de bien choisir l'action autour de laquelle la fonction sera étudiée, puisque l'action détermine les valeurs des critères différents de celui qui est visualisé. Le choix naturel conduit à étudier d'abord la fonction pour les exemples de l'ensemble d'entraînement, auxquels le décideur est déjà familier. Puisque le décideur est censé bien connaître ces exemples, il devrait facilement pouvoir vérifier si la fonction lui semble correcte ou non.

Il est important de préciser que la visualisation ne s'effectue que pour chaque critère de sélection, sans prendre en compte les corrélations éventuelles avec les autres. Il est certain que ceci peut nous conduire à des actions complètement irréalisables, conduisant à choisir par exemple une Rolls Royce qui coûte zéro franc.

Il est vrai que dans le cas du choix de voitures, par exemple, la vitesse maximale augmente pratiquement toujours d'une façon très corrélée avec le nombre de chevaux. Si ces deux critères sont vraiment corrélés, la visualisation peut se faire en faisant varier les valeurs des deux critères en même temps, mais ceci demande une connaissance a priori de la corrélation.

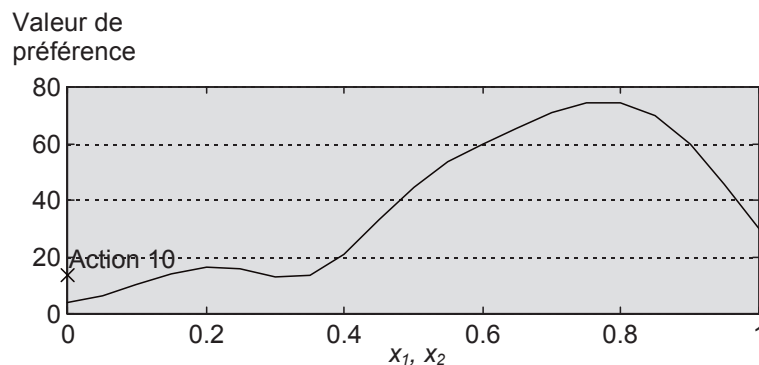


Figure 3-2. Évolution de la valeur de préférence en fonction des critères x_1 et x_2 (valeurs égales pour les deux) pour la fonction de préférence de la Figure 2-2.

Dans une visualisation simultanée des deux critères de la fonction de préférence en trois dimensions du Chapitre 5, la fonction serait visualisée le long de la diagonale passant du point $[0,0]$ au point $[1,1]$, comme illustré par la courbe de la Figure 3-2. Dans cette figure la croix indique uniquement la valeur de préférence de l'action, car il est difficile d'indiquer les valeurs de deux critères de sélection différents de façon claire en même temps.

Une autre fonctionnalité intéressante est la possibilité d'étudier la préférence d'une action par rapport à une autre en fonction de la valeur d'un ou plusieurs critères, comme dans la Figure

Chapitre 3. Analyse de la fonction de préférence apprise

3-3. Une telle comparaison permet au décideur d'analyser, entre autres, la cohérence de la fonction de préférence. Il y a un problème s'il voit que la fonction apprise est très différente pour des actions où elle devrait être pratiquement équivalente. De la même façon, il peut voir si la fonction apprise est similaire pour des actions où elle devrait être très différente.

Cette fonctionnalité permet d'élargir le domaine d'utilisation des SIAD à l'optimisation des actions les unes par rapport aux autres. Dans l'application du choix de voiture personnelle, ceci signifie que l'utilisateur de la méthodologie ne serait pas forcément la personne qui veut choisir sa voiture, mais aussi le vendeur de voitures qui voudrait savoir comment augmenter ses ventes auprès d'une certaine clientèle.

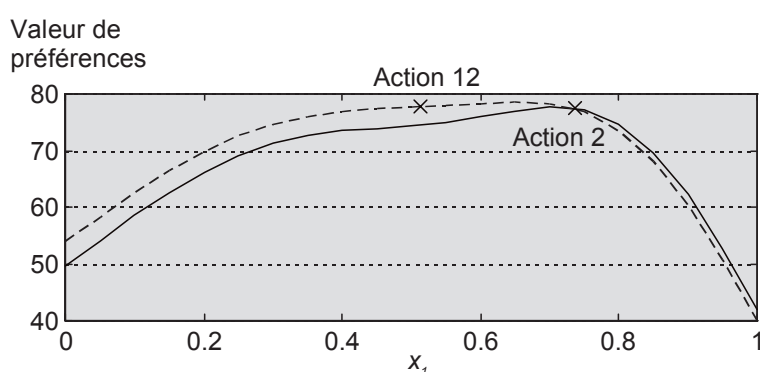


Figure 3-3. Comparaison de la fonction de préférence pour deux actions (deux valeurs différentes pour x_2 , permettant d'analyser comment l'une pourrait dépasser l'autre en modifiant la valeur du critère x_1 . Les croix indiquent les valeurs actuelles des actions pour ce critère.

L'apprentissage automatique permet d'identifier les préférences des consommateurs, comme le permettent d'ailleurs les méthodes d'analyse de données dans [Bertier & Bourouche, 1975] et les méthodes UTA [Jacquet-Lagrèze & Siskos, 1982] et PREFCALC [Jacquet-Lagrèze, 1984]. Ce sont néanmoins des méthodes linéaires qui ne permettent pas l'expression de fonctions de préférence contextuelle. Par contre, les méthodes fondées sur des règles de décision permettent le traitement de modèles de préférence contextuels [Slowinski, 1994]. Dans ce cas il s'agit néanmoins d'un modèle de préférence essentiellement symbolique, qui se prête difficilement à une analyse graphique.

L'objectif de l'analyse graphique est donc de permettre au décideur de voir si la fonction de préférence apprise ne correspond pas à la sienne à cause d'un nombre d'exemples insuffisant ou à cause d'une erreur. Le cas où le nombre d'exemples n'est pas suffisant n'est pas toujours

facile à voir à partir de l'analyse de la fonction apprise. Les coefficients de certitude, qui peuvent également faire l'objet d'une analyse graphique, peuvent constituer un indicateur intéressant.

3.2 Traitement des coefficients de certitude

Il est possible de calculer des coefficients de certitude pour les valeurs de sortie de INKA, les valeurs de sortie étant dans notre cas des valeurs de préférence. Il est donc possible de savoir si l'ensemble d'actions potentielles est bien couvert par INKA, ou s'il y en a pour lesquelles INKA ne saura pas répondre avec un bon coefficient de certitude.

3.2.1 Coefficients de certitude dans INKA

Le coefficient de certitude d'une action potentielle dépend de sa similarité par rapport à un exemple d'apprentissage. Si l'action est exactement similaire à l'exemple d'apprentissage, il y a une certitude de 100% que la réponse donnée par INKA soit bonne (sauf en cas de bruit). Plus les caractéristiques de l'action sont différentes de celles de l'exemple d'apprentissage utilisé, plus la certitude diminue.

Le calcul choisi pour obtenir le coefficient de certitude pour une action donnée consiste à prendre la valeur de sortie des neurones cachés la plus élevée comme coefficient de certitude:

$$cc = \max_{\forall i} (h_i) \quad (10)$$

où cc est le coefficient de certitude et h_i la valeur de sortie du neurone caché i . Le choix de prendre la valeur maximale correspond à des calculs de coefficient de certitude classiques. La variation du coefficient de certitude en fonction de x_1 et x_2 pour la fonction de préférences en trois dimensions du Chapitre 5 est illustrée dans la Figure 3-4. Le coefficient de certitude est bien sûr de 100% pour les valeurs correspondant aux deux exemples d'apprentissage et diminue en s'éloignant.

La valeur du paramètre d_i^2 utilisée pendant l'apprentissage de INKA joue un rôle important sur le coefficient de certitude obtenu, puisqu'elle détermine la largeur de la fonction de sortie des neurones cachés. Dans les applications traitées, d_i^2 a une valeur constante de un, parce que

Chapitre 3. Analyse de la fonction de préférence apprise

cette valeur a permis d'obtenir de bons résultats. En règle générale on peut constater que plus la fonction à apprendre est linéaire, plus la valeur optimale de d_i^2 augmente, ce qui élargit la fonction de sortie des neurones cachés de INKA et donne des coefficients de certitude plus grands. Par contre, si la fonction à apprendre est non-linéaire, la valeur optimale de d_i^2 sera plus petite, ce qui donne des coefficients de certitude plus petits.

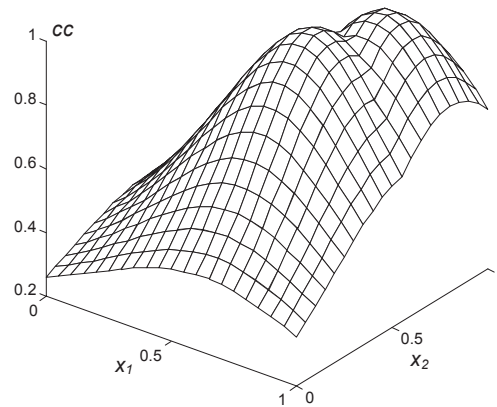


Figure 3-4. Visualisation des coefficients de certitude de INKA en fonction des valeurs de x_1 et de x_2 après l'introduction des deux premiers exemples de la fonction de préférence non-linéaire de la Figure 2-2.

Ce fonctionnement est tout à fait logique, car une fonction linéaire permet de faire des grandes généralisations loin des centroïdes des neurones cachés, tandis que pour une fonction non-linéaire très irrégulière il est difficile de généraliser. Pour prendre en compte cette possibilité, il peut être utile d'utiliser la version de l'algorithme d'apprentissage de INKA utilisant un d_i^2 variable, comme expliqué dans l'Annexe B.

Il y a un lien très fort entre les coefficients de certitude de la logique floue et les coefficients de certitude de INKA. Les principes des réseaux de neurones de type RBF sont, en effet, très proches des principes de la logique floue. Chaque neurone caché traite un sous-espace radial de l'espace qui correspond à des règles comme:

« SI x_1 est petit ET x_2 est grand ALORS y est moyen »

où x_1 et x_2 sont des critères de sélection, y est la valeur de préférence et « petit », « moyen » et « grand » sont des ensembles flous ayant des formes Gaussiennes. La valeur de sortie du neurone RBF est pratiquement équivalent au coefficient de certitude sur la conclusion de la règle, habituellement calculé par l'opérateur MIN associé à l'opération logique ET [Zadeh, 1965].

Dans la logique floue, il y a normalement un grand nombre de règles permettant de déduire que «y est moyen». Le coefficient de certitude final est habituellement calculé par l'opérateur MAX associé à l'opération logique OU, qui prend comme argument les coefficients de certitude de chaque règle ayant déduit que «y est moyen». Un résultat quantitatif peut ensuite être obtenu par une procédure dite de «defuzzification», qui permet de transformer les résultats qualitatifs en résultat quantitatif à l'aide des coefficients de certitude associés aux résultats qualitatifs.

Dans un réseau RBF il n'y a pas de valeurs qualitatives intermédiaires car le résultat est directement un résultat quantitatif. La similarité avec les opérations logiques OU de la logique floue est malgré tout assez claire, ce qui justifie l'utilisation de l'opérateur MAX pour les coefficients de certitude d'INKA. L'opérateur MAX n'est pas le seul possible. Il existe un grand nombre de calculs de coefficients de certitude associés à l'opérateur OU, MAX est celui qui est utilisé le plus fréquemment [Micro Data Base Systems, 1990].

Le coefficient de certitude permet d'estimer si l'ensemble d'actions potentielles est bien couvert par INKA. Par contre, il ne permet pas de déterminer où les problèmes de certitude sont les plus graves. Une analyse graphique est encore une fois utile pour le faire.

3.2.2 Analyse graphique des coefficients de certitude de INKA

Le coefficient de certitude peut être visualisé en fonction des valeurs des critères de sélection de la même façon que la valeur de préférence, comme le montre la Figure 3-5 pour la fonction de préférence en trois dimensions.

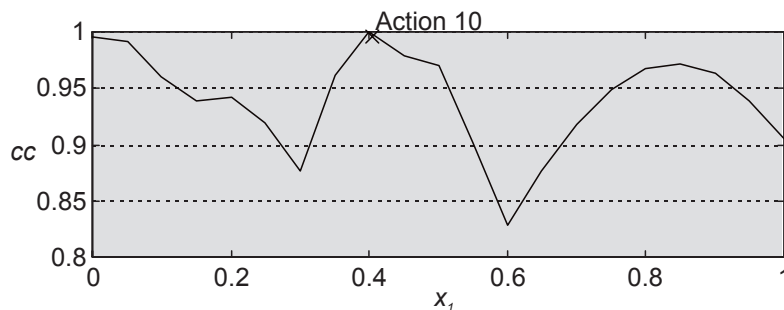


Figure 3-5. Évolution du coefficient de certitude en fonction de la valeur du critère x_1 pour l'action 10.

Chapitre 3. Analyse de la fonction de préférence apprise

L'analyse graphique des coefficients de certitude peut notamment permettre d'identifier les exemples d'apprentissage manquant dans l'espace d'entrée. Cette information peut, entre autres, servir à choisir les pseudo-exemples décrits au Chapitre 4.

3.2.3 Coefficients de certitude d'un ensemble d'actions

Il est intéressant pour le décideur de connaître un coefficient de certitude non seulement pour une action, mais également pour l'ensemble d'actions potentielles. Le coefficient de certitude moyen sur l'ensemble d'actions potentielles sert d'indicateur de certitude sur leur classement. L'évolution de cet indicateur peut être étudiée en fonction du nombre d'exemples d'apprentissage utilisés, comme dans la Figure 3-6. Cet indicateur peut, et doit, être utilisé en combinaison avec d'autres indicateurs de précision pour estimer la valeur ajoutée de chaque nouvel exemple. Dans le domaine de la statistique d'autres techniques sont utilisées pour estimer des intervalles de confiance en fonction du nombre d'exemples [Wright, 1991].

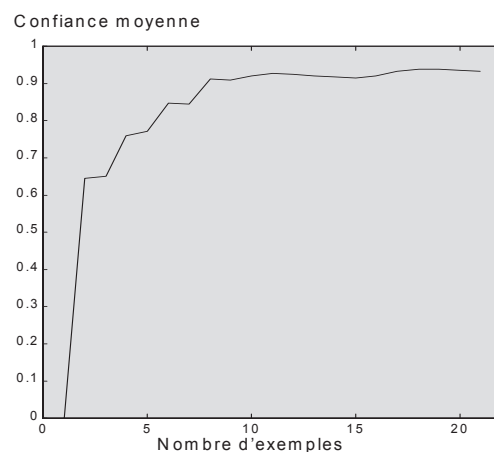


Figure 3-6. Évolution de la moyenne des coefficients de certitude sur toute l'espace d'entrée en fonction du nombre d'exemples pour la fonction en trois dimensions de la Figure 2-2 (a).

Il est également possible d'obtenir un coefficient de certitude sur toute la fonction de préférence apprise par une simulation de Monte-Carlo, par exemple. Ceci correspond au coefficient de certitude moyen de INKA sur tout l'espace d'entrée. La valeur moyenne des coefficients de certitude de la Figure 3-4 est de 0.65.

4. Correction de la fonction de préférence par pseudo-exemples

L'analyse graphique de la fonction de préférence apprise par INKA peut permettre de détecter des situations où les préférences apprises par INKA ne correspondent pas à celles du décideur. Deux types d'erreurs dans la fonction de préférence sont à envisager. Le premier type est évident et correspond au cas où la fonction de préférence apprise n'est pas monotone pour un critère de sélection pour lequel elle devrait l'être. C'est le cas par exemple pour le critère de choix de voiture « volume du coffre », pour lequel la fonction de préférence apprise au niveau de la « VW Passat VR6 » est illustrée dans la Figure 4-1. Dans ce cas, plus le volume est grand, meilleure devrait être la valeur de préférence.

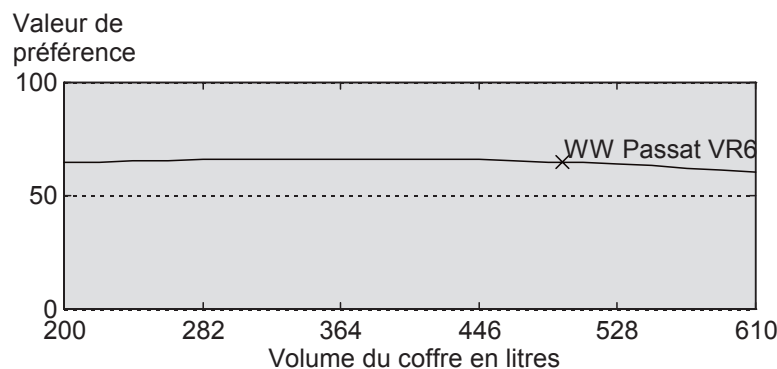


Figure 4-1. Fonction de préférence pour le choix de voiture, critère « volume du coffre », qui devrait être monotone et croissante mais qui ne l'est pas.

L'autre type d'erreurs est dû au manque d'exemples. Ces erreurs sont détectables uniquement par le décideur lui-même et nécessitent une interaction directe. Ce type d'erreur est plus difficile à détecter, spécialement dans le cas de décideurs abstraits (la population locale, l'environnement, ...). Un exemple de ce type d'erreur pour le choix d'une voiture serait de ne pas avoir des exemples représentant le fait que le décideur veut éviter de dépenser plus de 200.000 francs français, comme exprimé par les règles introduites au Chapitre 7.

Il est néanmoins très difficile ou même impossible de modifier la fonction de préférence apprise directement, car il faudrait modifier les poids et la structure du réseau neuronal. La correction de la fonction de préférence peut alors se faire uniquement par des exemples

d'apprentissage supplémentaires. La notion de *pseudo-exemples*, présentée au chapitre suivant, permet une correction de la fonction simple et rapide.

4.1 Pseudo-exemples

Pour corriger et augmenter la précision de la fonction de préférence apprise, il est possible d'augmenter l'ensemble d'entraînement par l'introduction de *pseudo-exemples*, qui ne sont pas des actions réalisables, mais pour lesquels le décideur peut facilement exprimer sa préférence. De tels exemples sont les actions idéale et anti-idéale, qui sont souvent utilisées comme des points de référence dans les SIAD. Une voiture idéale pourrait par exemple être une Rolls-Royce pour le prix d'une Citroën 2CV, mais une telle voiture existe très rarement en réalité. L'utilisation de tels exemples dans l'ensemble d'entraînement de INKA est néanmoins recommandée. Tout d'abord, il est facile pour le décideur de donner des valeurs de préférence à de telles actions. Ensuite, de telles actions apportent beaucoup d'information sur les préférences du décideur.

D'autres pseudo-exemples peuvent être introduits de façon interactive par le décideur dans des cas où il trouve que la fonction de préférence apprise ne correspond pas à la sienne. La Figure 4-1 montre que la fonction est mal apprise au moins pour la valeur maximale du critère « volume du coffre ». Dans ce cas, il lui suffit d'indiquer une valeur de préférence plus correcte pour la valeur maximale de ce critère, qui constitue ainsi un nouvel exemple de ses préférences.

Les pseudo-exemples peuvent également être construits à partir de comparaisons entre deux actions, comme dans la Figure 3-3. Il est souvent facile d'estimer la valeur d'un critère de sélection qui permettrait à l'action a de passer devant l'action b . Dans de nombreuses situations il n'y a qu'une seule caractéristique qui empêche une action d'être la meilleure. Il est dans ce cas particulièrement facile pour le décideur de déterminer quelle serait la valeur à attribuer au critère en question pour permettre à a de devenir meilleur que b .

La valeur du coefficient de certitude de INKA (Figure 3-5) permet au décideur de voir immédiatement quelles sont les valeurs pour lesquelles la fonction de préférence est le plus susceptible d'être incorrecte. Si la fonction de préférence est aussi incorrecte pour deux valeurs du même critère, il vaut normalement mieux introduire le pseudo-exemple pour la valeur où la certitude est plus petite.

Les pseudo-exemples sont par défaut introduits là où la fonction de préférence est la plus incorrecte et où l'information obtenue à l'aide de chaque pseudo-exemple supplémentaire reste relativement grande. Ce principe rejoint bien celui de l'algorithme d'apprentissage de INKA, où l'on cherche à exprimer le plus d'information possible avec chaque neurone caché. Les pseudo-exemples sont donc pris en compte de façon assez directe. L'utilisation de pseudo-exemples est un moyen d'augmenter la taille de l'ensemble d'entraînement de façon rapide et efficace.

4.1.1 Saisie de pseudo-exemples

La façon la plus simple pour saisir les pseudo-exemples est de les introduire dans le graphe illustrant la fonction de préférence. Il suffit au décideur d'indiquer le point sur la fonction qu'il désire corriger et le nouveau point par lequel la fonction devrait passer. Cela peut se faire à l'aide d'une souris ou d'autres moyens d'interaction.

Comme indiqué dans la Figure 2-1, chaque introduction d'un nouvel exemple est suivie par un nouvel apprentissage par INKA. Après l'apprentissage le décideur doit ré-afficher la fonction de préférence apprise pour vérifier que la nouvelle fonction correspond bien à ce qu'il voulait. L'introduction de pseudo-exemples s'effectue critère par critère pour ne pas dépasser les capacités cognitives du décideur, même s'il est en principe possible de le faire pour plusieurs critères. Si la fonction de préférence semble correcte pour un critère, le décideur peut continuer à étudier le critère suivant jusqu'au moment où la fonction de préférence apprise lui semble suffisamment correcte.

La saisie de pseudo-exemples ne peut se faire que par rapport aux exemples d'apprentissage, sinon il est pratiquement sûr qu'ils seront incohérents avec les exemples déjà introduits. Dans ce cas il y a de grands risques pour que la fonction de préférence apprise ne converge jamais.

Cette procédure d'introduction de pseudo-exemples est une procédure qui peut être assez amusante et intéressante pour le décideur, qui découvre ses propres préférences. Au fur et à mesure qu'il découvre ses préférences, il peut également avoir une tendance à les modifier. C'est pour cette raison qu'il faut laisser la possibilité au décideur de modifier facilement les exemples déjà saisis à l'aide de l'interface graphique.

4.1.2 Transformations des valeurs de préférence

Les valeurs de préférence des pseudo-exemples sont des valeurs cardinales, contrairement aux valeurs ordinales quantifiées données aux exemples d'apprentissage initiaux. Si le décideur désire introduire encore un nouvel exemple « réel », ceci peut poser un problème. Il lui reste quelques possibilités pour le faire:

Comparer une fois de plus les exemples initiaux plus le nouvel exemple deux à deux pour obtenir des nouvelles valeurs de préférence ordinales quantifiées.

Introduire le nouvel exemple directement avec une valeur de préférence cardinale ou le situer uniquement par rapport à ses deux exemples réels voisins.

Dans le premier cas, il devient nécessaire de recalculer les valeurs de préférence des pseudo-exemples précédemment introduits pour correspondre aux nouvelles valeurs de préférence des exemples « réels ». Il est possible de le faire, puisque chaque pseudo-exemple a été saisi relativement à un exemple réel ou à un autre pseudo-exemple. Il est également possible de refaire le chemin parcouru avec les pseudo-exemples en les représentant un par un pour permettre au décideur de les résituer.

Ce calcul peut se faire de plusieurs façons, en augmentant ou en diminuant les valeurs de préférence des pseudo-exemples par la même valeur que la valeur de préférence de l'exemple réel. Le calcul peut également se faire par règle de trois relativement aux valeurs de préférence idéale et anti-idéale ou relativement aux valeurs de préférence des exemples « voisins » dans l'ordre de préférence. La meilleure méthode reste encore à déterminer.

Dans le cas d'une introduction de valeur de préférence cardinale, il y a un risque de diminution de cohérence au niveau de la quantification des préférences ordinales, même si l'ordre entre les exemples reste correct. Si l'erreur introduite n'est pas trop grande, cette approche est plus facile car les valeurs de préférence déjà introduites restent constantes.

Il reste au décideur à évaluer l'approche à utiliser, en se souvenant que l'ordre des exemples est le plus souvent plus important que la précision numérique. Pour faciliter son travail et pour obtenir des résultats aussi bons que possible, le décideur devrait fournir un ensemble d'entraînement initial aussi bon que possible.

4.2 Application au problème du choix de voiture

Dans le cas du SAD pour le choix d'une voiture personnelle, l'ensemble d'entraînement était uniquement constitué de douze exemples « réels » et de deux pseudo-exemples correspondant aux voitures idéale et anti-idéale. Le classement obtenu était déjà assez proche du classement « correct », mais une analyse graphique de la fonction de préférence montre vite qu'il y a encore des choses à améliorer.

Le nombre total de pseudo-exemples introduits est de 48, tout en corrigeant uniquement les erreurs évidentes de monotonie de la fonction de préférence. Un autre type d'erreur dans cette application vient des grandes non-linéarités pour le prix de la voiture, la consommation d'essence et la puissance maximale.

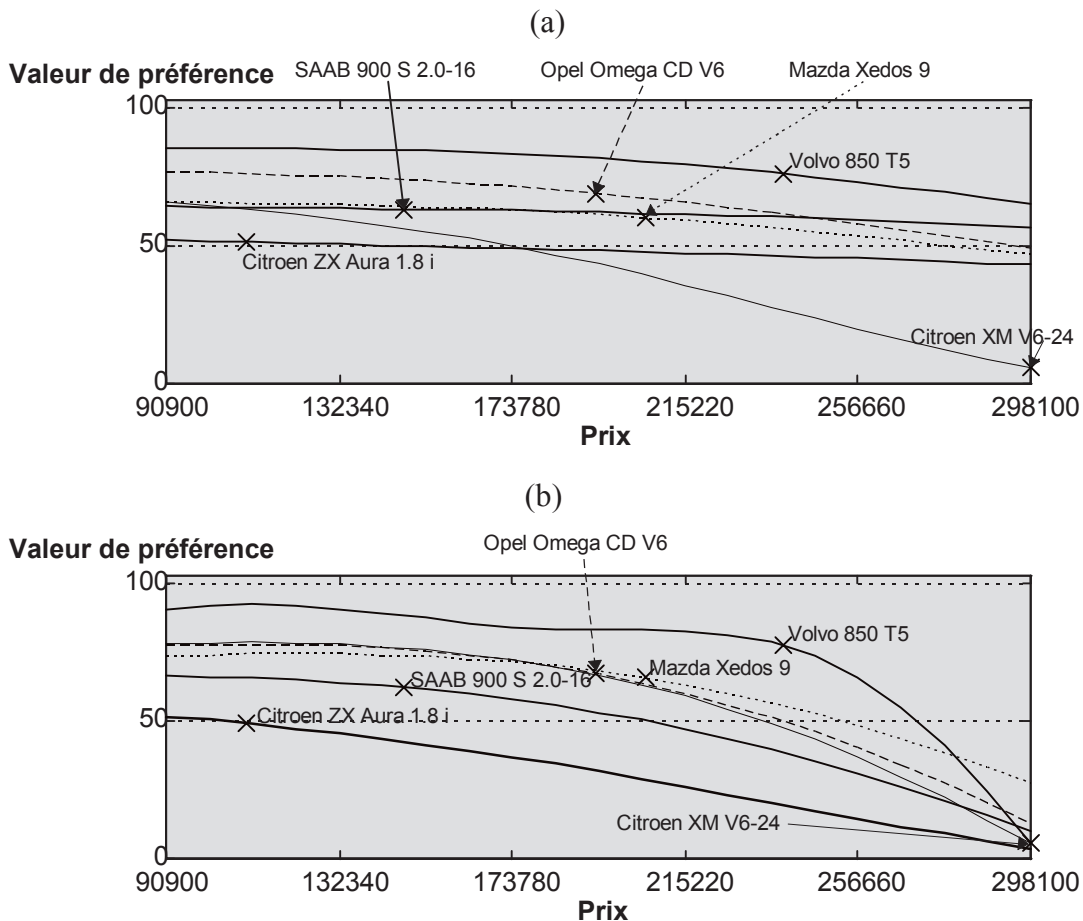


Figure 4-2. Analyse graphique de la fonction de préférence pour le critère prix (a) avant l'introduction de pseudo-exemples, (b) après l'introduction de 9 pseudo-exemples pour corriger la fonction pour le critère prix.

Chapitre 4. Correction de la fonction de préférence par pseudo-exemples

Le travail est dans ce cas simplifié par le nombre limité d'exemples réels, car il est possible d'afficher la fonction de préférence apprise pour tous les exemples d'apprentissage dans la même figure. Il y en a que 6 inclus dans la Figure 4-2 pour des raisons de clarté (des couleurs permettent de les mieux illustrer sur l'écran d'un ordinateur). Cette illustration permet de s'assurer de la cohérence de la fonction apprise pour plusieurs voitures à la fois, car l'allure de la fonction est dans la plupart de cas la même pour toutes les voitures, mais la fonction est uniquement décalée en hauteur.

Dans la Figure 4-2(a) il est évident que la fonction n'est pas correcte ni pour la Citroën ZX Aura 1.8i, ni pour la SAAB 900 S 2.0-16. La fonction n'est pas entièrement correcte pour la Volvo 850 T5 non plus, car elle devrait atteindre presque zéro pour le prix maximal (298100, le prix de la Citroën XM V6-24).

Il n'est pas nécessaire d'introduire des pseudo-exemples pour toutes les erreurs grâce à la fonctionnalité de généralisation des réseaux neuronaux. La généralisation signifie ici que la correction de la fonction pour une voiture corrige plus ou moins la fonction pour d'autres voitures. Plus une voiture présente des similitudes avec la voiture pour laquelle la fonction est corrigée, plus sa valeur de préférence sera corrigée aussi.

Après l'introduction de 9 pseudo-exemples, la fonction de préférence est celle de la Figure 4-2(b). Les erreurs les plus graves ont été corrigées et celles qui restent ne sont pas jugées trop importantes. Certaines corrections sont plus difficiles que d'autres. Dans la plupart des cas, il suffit d'introduire un seul pseudo-exemple. Par contre, pour corriger la fonction de préférence pour la Volvo 850 T5, il était nécessaire d'introduire trois pseudo-exemples. La grande non-linéarité nécessite dans ce cas de corriger la fonction non seulement pour les valeurs extrêmes, mais également pour des valeurs intermédiaires.

La saisie des 48 pseudo-exemples prend moins d'une heure. Deux critères principaux sont utilisés pour choisir le pseudo-exemple à introduire à chaque étape: l'erreur dans la fonction elle-même et le coefficient de certitude associé. Le pseudo-exemple doit être introduit pour un exemple et une valeur d'un critère pour lesquels la fonction de préférence présente une erreur évidente et pour lesquels le coefficient de certitude est faible.

Au fur et à mesure que des pseudo-exemples sont ajoutés, il devient de moins en moins facile de choisir le pseudo-exemple à introduire. Le passage au critère de sélection suivant se fait lorsque le décideur ne trouve plus de pseudo-exemple évident à introduire. Les améliorations

obtenues par l'introduction de nouveaux pseudo-exemples deviennent souvent négligeables dans ce moment-là.

La comparaison entre les deux classements du Tableau 7-2 (a) et (c) montre qu'il y a 6 voitures communes parmi les 10 premières, au lieu de 7 pour le classement (b). La différence est devenue plus grande qu'avant l'introduction de pseudo-exemples! Une étude plus approfondie montre néanmoins que ce classement est en réalité meilleur.

Des voitures du classement (a) qui ne figurent pas parmi celles du classement (c), c'est encore la Opel Omega MV6 pour laquelle il y a la différence la plus grande (93^{ème} au lieu de 7^{ème}). Les autres sont la BMW 525i Worldline qui se trouve 47^{ème} au lieu de 9^{ème}, la BMW 325i qui se trouve 13^{ème} au lieu de 4^{ème} et la Honda Civic 1.6 VTi qui se trouve 18^{ème} au lieu de 10^{ème}. La Opel Omega MV6 et la BMW 525i Worldline souffrent du manque d'exemple des voitures avec des prix autour de 250.000,- francs français. Ce manque pourrait être corrigé soit par l'introduction d'un exemple supplémentaire ou par des pseudo-exemples bien choisis.

(a) Classement « correct »		(b) Classement INKA 1		(c) Classement « pseudo »	
Opel Vectra V6	77.70	SAAB 900 SE 2.0 Turbo	76.91	SAAB 900 SE 2.0 Turbo	87.47
SAAB 900 SE 2.0 Turbo	76.69	Volvo 850 T5	76.41	Opel Vectra V6	87.24
Volvo 850 T5	76.40	BMW 325i	74.73	SAAB 900 SE 2.5 V6	82.63
BMW 325i	74.26	BMW 525i Worldline	71.09	Volvo 850 GLT 2.5	79.37
Lancia Thema Turbo 16v LS	73.81	Opel Omega CD 2.0	70.64	Opel Vectra CDX 2.0	78.36
SAAB 900 SE 2.5 V6	72.21	Volvo 850 GLT 2.5	70.63	SAAB 900 S 2.3-16	77.30
Opel Omega MV6	70.52	SAAB 900 SE 2.5 V6	70.51	Lancia Thema Turbo 16v LS	77.30
Volvo 850 GLT 2.5	70.45	BMW 320i Pack	70.34	Volvo 850 T5	76.69
BMW 525i Worldline	70.32	Opel Vectra V6	70.16	Mazda 626 Gti 2500	74.66
Honda Civic 1.6 VTi	69.82	Mercedes C 280	69.83	Toyota Carina E 2.0 GLi	73.69

Tableau 4-1. Comparaison du (a) classement « correct », (b) classement par la fonction de préférence apprise avec 14 exemples, (c) classement par la fonction apprise avec 14 exemples plus 48 pseudo-exemples. Uniquement les dix premières voitures des classements sont incluses. Les voitures en gras étaient présentes dans l'ensemble d'entraînement.

Des voitures du classement (c) qui ne figurent pas parmi celles du classement (a), il y a la Toyota Carina E 2.0 GLi qui se trouve 10^{ème} au lieu de 32^{ème}, la Mazda 626 Gti 2500 qui se trouve 9^{ème} au lieu de 28^{ème}, la SAAB 900 S 2.3-16 qui se trouve 6^{ème} au lieu de 22^{ème} et l'Opel Vectra CDX 2.0 qui se trouve 5^{ème} au lieu de 11^{ème}. Les différences maximales ont un peu diminué ainsi que la différence moyenne par rapport au classement (b), mais l'amélioration n'est pas très grande. L'avantage principal obtenu par l'introduction de pseudo-exemples est l'augmentation de la stabilité du classement d'actions potentielles. Il devient également plus facile d'étudier et de comprendre la fonction de préférence.

Plus il y a de pseudo-exemples par rapport aux vrais exemples, moins efficace devient l'introduction de nouveaux pseudo-exemples, puisqu'ils sont définis par rapport aux exemples réels. Pour cette raison, le sous-espace où la fonction de préférence est définie peut rester limité. Il devient donc nécessaire d'augmenter le nombre de vrais exemples pour couvrir tout

l'espace d'entrée. Dans ce cas, le décideur devra choisir s'il veut le faire en utilisant des valeurs de préférence ordinales quantifiées ou des valeurs cardinales, comme indiqué dans le Chapitre 4.1.2.

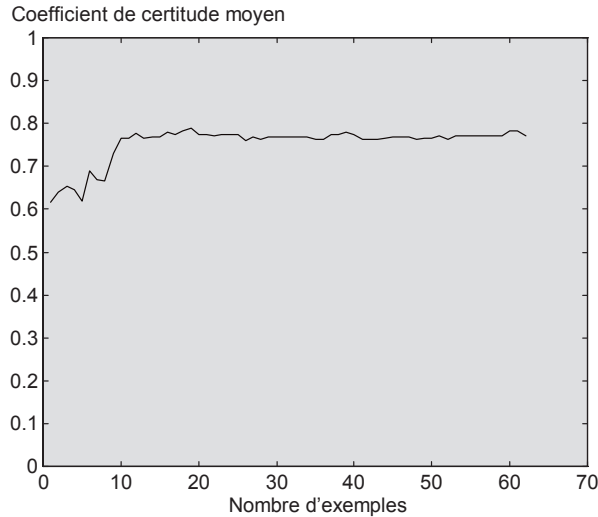


Figure 4-3. Évolution du coefficient de certitude moyen de INKA sur l'ensemble de voitures potentielles en fonction du nombre d'exemples.

Le degré de couverture de l'ensemble des actions potentielles peut être estimé par la moyenne des coefficients de certitude de INKA sur chaque action potentielle. L'évolution de cet indicateur par rapport au nombre d'exemples est indiquée dans la Figure 4-3. Le coefficient de certitude moyen augmente en fait très peu avec l'introduction de pseudo-exemples, même si la fonction de préférence apprise devient plus exacte. Ceci s'explique par l'éloignement des pseudo-exemples des voitures potentielles.

Par contre, si le coefficient de certitude moyen est calculé sur tout l'espace d'entrée, il augmente assez bien, comme c'était le cas dans la Figure 3-6. Pour augmenter le coefficient de certitude moyen sur l'ensemble d'actions potentielles, il faudrait augmenter le nombre d'exemples réels. L'effet des pseudo-exemples sur les autres indicateurs de précision et de sensibilité est illustré dans les chapitres suivants.

5. Indicateurs de précision

Estimer la précision de la fonction de préférence apprise, signifie ici d'estimer les capacités de généralisation du réseau neuronal. Des bonnes capacités de généralisation sont nécessaires pour obtenir des valeurs de préférence et des ordres corrects d'actions potentielles. Plusieurs indicateurs sont utilisés pour montrer l'évolution de la précision en fonction du nombre d'exemples. Ces indicateurs devraient aider le décideur à déterminer son niveau de satisfaction du résultat et ainsi lui permettre de choisir le moment d'arrêt de la procédure. Un premier indicateur, celui du coefficient de certitude moyen pour un ensemble d'actions a déjà été présenté dans le Chapitre 12.

La méthode la plus souvent utilisée pour estimer les capacités de généralisation consiste à diviser l'ensemble d'apprentissage en deux sous-ensembles indépendants - *l'ensemble d'entraînement* et *l'ensemble de validation* (de test). Il y a souvent 2/3 de tous les exemples dans l'ensemble d'entraînement et 1/3 dans l'ensemble de validation, mais cette partition varie fortement en fonction de l'application. L'ensemble d'entraînement sert à construire le réseau neuronal par l'apprentissage et l'ensemble de validation à estimer ses capacités de généralisation.

Pour obtenir de bons résultats il est néanmoins recommandé d'avoir au moins 1000 exemples dans l'ensemble de validation [Weiss & Kulikowski, 1991]. Il faut donc un total de 3000 exemples ou plus (ceci bien sûr en fonction du nombre de variables d'entrée et de la complexité de la fonction à apprendre). Il est évident qu'il est très difficile d'obtenir de tels nombres d'exemples sur les fonctions de préférence des décideurs. Ce faible nombre d'exemples pose des problèmes non seulement pour apprendre la fonction de préférence, mais aussi pour estimer la précision et les capacités de généralisation obtenues.

D'autres techniques existent qui permettent d'estimer la vraie précision obtenue, même avec peu d'exemples. Ce sont des techniques de « *resampling* », qui consistent à réutiliser tous les exemples en les utilisant parfois dans l'ensemble d'entraînement et parfois dans l'ensemble de validation. La technique retenue en particulier dans le cadre de ce travail est appelée « *k-fold* ».

cross validation », qui donne de bonnes estimations et qui est simple [Weiss & Kulikowski, 1991].

5.1 Validation des résultats d'apprentissage pour un faible nombre d'exemples

Il existe des méthodes bien connues dans le domaine de la statistique qui permettent d'estimer les capacités de généralisation des systèmes construits à partir d'exemples. Ce sont néanmoins des méthodes assez mal connues dans les domaines de l'apprentissage automatique et des réseaux de neurones [Baxt & White, 1995], [Orr, 1995]. Il s'agit de méthodes comme le « *bootstrapping* », « *k-fold cross validation* », « *leaving-one-out* », etc., décrites par exemple dans [Weiss & Kulikowski, 1991] et [Hall, 1992].

Le principe de ces méthodes est de partager l'ensemble d'apprentissage en un ensemble d'entraînement et un ensemble de validation et de refaire l'apprentissage et la validation plusieurs fois. L'indicateur des capacités de généralisation utilisé est normalement la moyenne de chaque essai. L'algorithme de « *k-fold cross validation* » est présenté plus en détail dans la Figure 5-1.

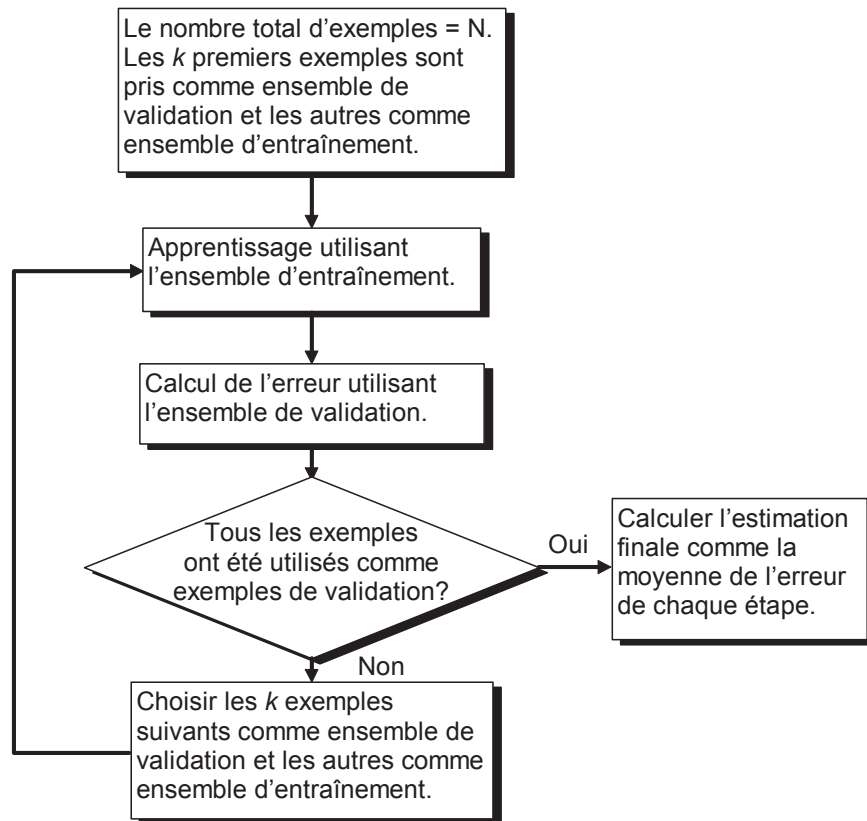


Figure 5-1. Algorithme de « k-fold cross validation ».

« Leaving-one-out » est une version de « k-fold cross-validation », où k est égal à un. Les avantages de « leaving-one-out » sont d'être simple, non-biaisée et relativement fiable même pour des petits nombres d'exemples. L'inconvénient de cette méthode est qu'elle demande souvent des temps de calcul très longs. Quand le nombre total d'exemples augmente, il est souvent utile d'augmenter la valeur de k . Les limites et les conditions pour l'emploi de telle ou telle méthode sont bien présentées dans [Weiss & Kulikowski, 1991].

Ce sont des méthodes qui sont utilisées avant tout pour des problèmes de classification, ce qui n'est pas le cas ici. Dans des problèmes de classification, l'erreur est exprimée en fonction du nombre d'exemples mal classifiés. Dans les problèmes d'identification de fonctions, l'erreur est plutôt estimée par la racine de l'erreur quadratique moyenne (en anglais: *root mean square error*, RMSE), mais cette différence ne change pas l'applicabilité des méthodes.

La comparaison du RMSE estimé par « leaving-one-out » et le vrai RMSE de la Figure 5-2 pour le problème de choix de voiture montre que l'estimation d'erreur obtenue par « leaving-one-out » est un peu pessimiste. Après le traitement de 20 exemples la différence devient néanmoins tolérable. Il est difficile d'obtenir de meilleurs résultats pour moins d'exemples, même si d'autres méthodes que « leaving-one-out » pourraient parfois faire mieux.

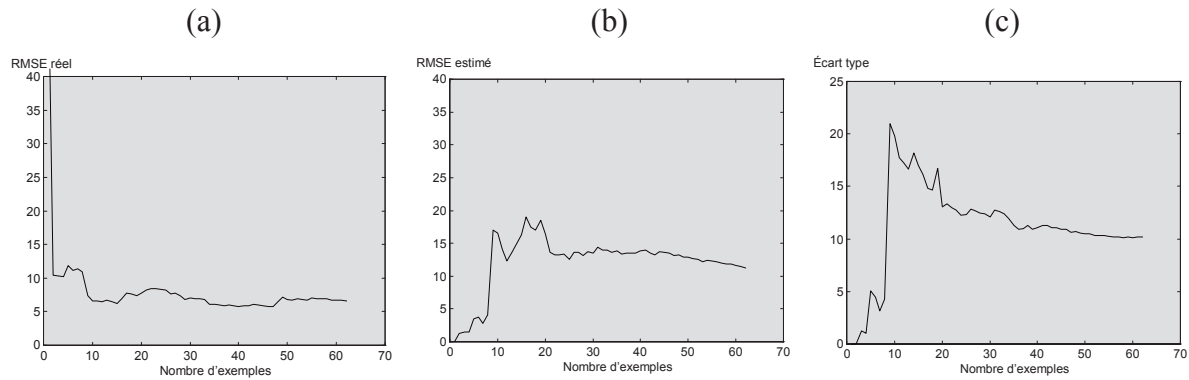


Figure 5-2. (a) RMSE calculé pour l'ensemble d'actions potentielles, (b) valeur moyenne et (c) écart type de RMSE estimés par « leaving-one-out » en fonction du nombre d'exemples pour le choix de voiture.

En retournant la courbe d'évolution du RMSE estimée par « leaving-one-out » en fonction du nombre d'exemples, elle correspond à la courbe « valeur ajoutée » de la Figure 10-2. Cette comparaison permet de montrer les problèmes pratiques de l'estimation de la valeur ajoutée par chaque exemple, car la précision n'augmente pas toujours de façon régulière. Ceci est particulièrement vrai pour des petits nombres d'exemples.

La Figure 5-2(c) montre comment évolue l'écart type de RMSE calculé pour chaque itération de « leaving-one-out ». Plus la variance est grande, plus le résultat de l'apprentissage est sensible à l'ajout ou à la suppression de nouveaux exemples. Une très grande variance implique que la fonction de préférence n'est pas stable par rapport à l'ensemble d'entraînement. Il s'agit donc d'une première version d'analyse de sensibilité.

L'intérêt de la présentation de tels graphes au décideur est de lui permettre de voir quand l'ajout d'exemples n'augmente plus de façon significative ni la précision, ni la stabilité de la fonction de préférence apprise. Une traduction des courbes (b) et (c) serait que la fonction de préférence apprise avec 25 exemples est déjà assez bonne, mais qu'il en faut plus de 50 pour obtenir une bonne stabilité pour les valeurs de préférences.

5.2 Calcul de la différence entre deux classements

L'indicateur d'évolution de l'erreur numérique par RMSE est un moyen général pour estimer les performances d'approximation et de généralisation d'un réseau neuronal. Dans le domaine des SIAD, cet indicateur n'est pas bien adapté, car le décideur n'a guère envie d'essayer de

comprendre la signification de cette erreur numérique sur sa prise de décision. Il faut, pour cette raison, introduire d'autres indicateurs qui correspondent plus à la réalité du décideur. L'indicateur d'erreur utilisé à la place du RMSE est celui de la différence entre deux classements d'un ensemble d'actions.

Prenons deux classements de l'ensemble d'actions potentielles: $O_1(A)$ et $O_2(A)$. Pour chaque action a_i de l'ensemble A , nous calculons sa place dans les deux classements, $O_1(a_i)$ et $O_2(a_i)$. La différence de place de l'action a_i dans les deux classements est:

$$d(O_1(a_i), O_2(a_i)) = |O_1(a_i) - O_2(a_i)| \quad (11)$$

Une fois que cette différence est connue pour chaque action a_i , il est possible de calculer des indicateurs de différence plus globaux, comme la différence moyenne et maximale. Puisque le classement correct n'est pas connu dans des problèmes de décision (sinon il n'y a pas de problème de décision), il est nécessaire d'estimer ces indicateurs autrement. Une version modifiée de « k -fold cross validation » permet de proposer une solution à ce problème.

5.3 Estimation de l'erreur moyenne et maximale par k -fold cross validation

L'utilisation du RMSE comme indicateur d'erreur ne pose pas de problèmes car les valeurs de préférence correctes sont des valeurs cardinales. Par contre, l'utilisation de la différence entre deux classements comme indicateur présente certains problèmes. Le classement d'une action revient à lui donner une valeur de préférence ordinale. Or, une valeur de préférence ordinale est toujours définie par rapport à un ensemble d'actions et un classement spécifiques. Si nous utilisons le « leaving-one-out », il n'y a qu'une seule action dans l'ensemble de validation et il n'existe pas de classement. Même l'utilisation de « k -fold cross validation » pose des problèmes car le nombre k reste toujours petit. Plusieurs solutions sont envisageables pour résoudre ce problème.

La solution qui semble être théoriquement la plus correcte serait de partir du seul classement connu et sûr, le classement fourni par le décideur pour les exemples d'apprentissage « réels ». Il serait trop dangereux d'inclure les pseudo-exemples, puisque leurs valeurs de préférence sont des valeurs cardinales au départ, dont la cohérence n'est pas certaine. L'ensemble de validation à utiliser à chaque étape du « k -fold cross validation » serait ainsi k exemples

Chapitre 5. Indicateurs de précision

d'apprentissage réels, pour lesquels le classement correct entre eux est connu. Ceci donnerait des indicateurs de différences moyenne et maximale comme un chiffre de x sur k , qui peut facilement être transformé en un chiffre y sur N , où N est le nombre d'actions potentielles.

Pour illustrer cette méthode, supposons que nous avons 12 exemples d'apprentissage et 113 actions potentielles. Par la méthode de « k -fold cross validation » nous obtenons alors une différence moyenne de, par exemple, 0.7 pour un k égal à 3. La différence normalisée est de 0.7 sur 3, ou 26.4 sur 113. Nous pouvons ainsi estimer que la différence moyenne entre le classement correct et le classement calculé soit de 26.4 pour l'ensemble d'actions potentielles.

Il y a malheureusement plusieurs problèmes avec une telle approche. Tout d'abord, le nombre k restera toujours petit, ce qui implique une très mauvaise précision pour les indicateurs obtenus. Dans le cas de « leaving-one-out » l'analyse devient même impossible. Ensuite, pour que l'estimation obtenue soit correcte il faudrait que la distribution des valeurs de préférence soit la même pour l'ensemble d'apprentissage et l'ensemble d'actions potentielles. Or, les exemples d'apprentissage ont plutôt été choisis pour bien couvrir l'espace d'entrée et il n'y a aucune garantie au niveau de la distribution des valeurs de préférence. Il est, par exemple, possible que 50% des actions potentielles aient des valeurs de préférence entre 50 et 60, mais que seulement 20% des exemples d'apprentissage aient des valeurs dans le même intervalle. Les valeurs des indicateurs obtenues sont dans ce cas erronées.

Le même problème se produit pour toute autre approche essayant d'utiliser le classement correct des exemples d'apprentissage « réels » pour obtenir des indicateurs et les appliquer à l'ensemble d'actions potentielles. L'approche proposée pour éviter ces problèmes est une autre adaptation de « k -fold cross validation », où la différence moyenne et la différence maximale sont estimées directement pour l'ensemble d'actions potentielles. Mais malheureusement, nous ne connaissons pas le classement de l'ensemble d'actions potentielles. Il faut ainsi d'abord définir ce qu'est ce classement « correct » pour l'ensemble d'actions potentielles.

Le classement « correct » choisi est celui calculé par INKA après un apprentissage avec tous les exemples d'apprentissage, qui devient donc le classement de référence. Ensuite, les indicateurs de différence moyenne et maximale sont calculés par l'algorithme de « k -fold cross validation » en utilisant la différence entre le classement de référence et le classement calculé comme indicateur d'erreur à chaque étape.

Cette approche a été utilisée par la suite, malgré le fait que le classement « correct » ne l'est pas vraiment. La validité des indicateurs ainsi obtenus a été testée sur plusieurs applications et elle a été relativement bien vérifiée dans tous les cas, comme cela est illustré au chapitre suivant pour l'application au choix de voiture.

5.3.1 Évolution des indicateurs de précision pour le choix de voiture

Les différences moyenne et maximale pour l'ensemble d'actions potentielles étaient calculées en fonction du nombre d'exemples dans l'ensemble d'apprentissage pour l'application de choix de voiture. Les indicateurs étaient d'abord calculés, en comparant le bon classement obtenu par la fonction de préférence « correcte » et le classement calculé par la fonction de préférence apprise par INKA. Les indicateurs ont été ensuite estimés par la méthode de « leaving-one-out ».

Pour la différence moyenne, l'estimation est plutôt optimiste, comme l'indique la Figure 5-3. Une fois que ce phénomène est connu il devient possible de se faire une idée de la différence moyenne réelle. L'essentiel est d'être capable d'étudier le progrès obtenu par l'ajout d'exemples. Le progrès donné par la valeur estimée correspond assez bien à la valeur réelle de l'indicateur. Il semble que la valeur ajoutée au niveau de cet indicateur n'augmente pratiquement plus à partir de 50 exemples.

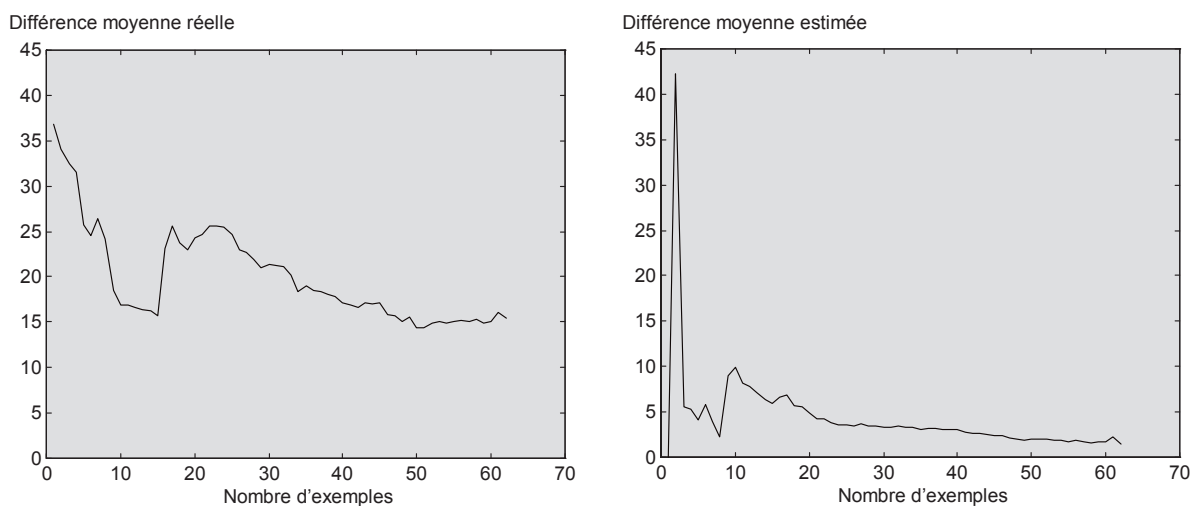


Figure 5-3. Comparaison entre la différence moyenne réelle et estimée par « leaving-one-out » en fonction du nombre d'exemples pour le choix d'une voiture.

Pour la différence maximale, l'estimation est assez bonne à partir d'une dizaine d'exemples. La différence maximale estimée augmente très fort après l'introduction de la Citroën XM V6-24 dans l'ensemble d'entraînement. Cet effet vient du fait que INKA n'avait pas « vu » d'exemple de voiture très chère auparavant, ce qui veut dire que la fonction de préférence apprise ne prenait pas en compte ce type de cas.

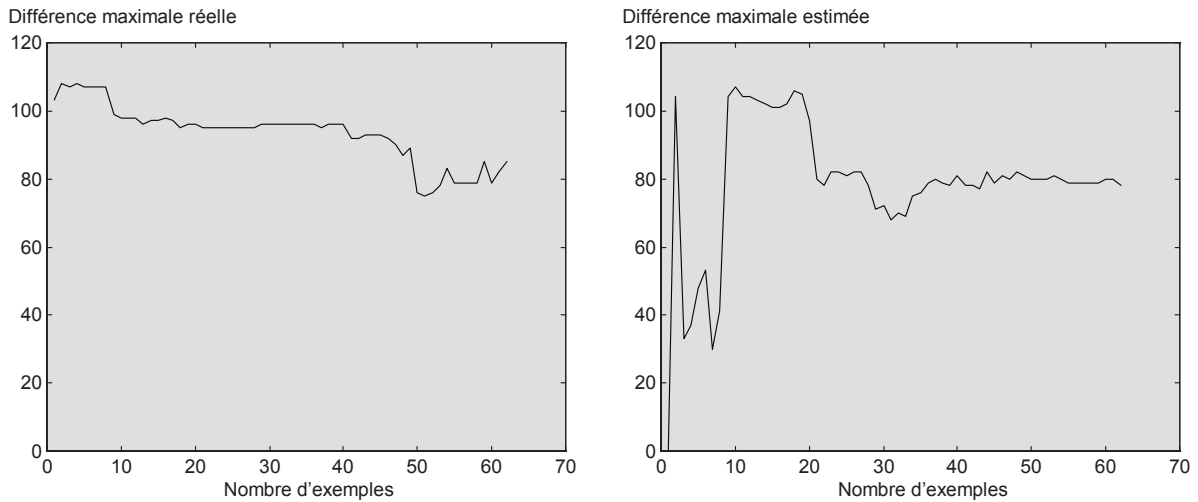


Figure 5-4. Comparaison entre la différence maximale réelle et estimée par « leaving-one-out » en fonction du nombre d'exemples pour le choix d'une voiture.

La différence maximale indique la dernière position à laquelle il est encore possible de trouver la meilleure action. Si c'est la différence maximale que le décideur veut absolument réduire, les graphes indiquent qu'il devrait continuer à introduire des exemples. Le test de robustesse conduit au Chapitre 6.2 est un moyen de l'éviter.

6. Analyse de sensibilité et test de robustesse

Une analyse de sensibilité consiste à étudier les conséquences de changements dans les paramètres d'un modèle sur les résultats obtenus [Maystre et al., 1994]. Les paramètres à étudier sont dans notre cas:

Les paramètres du réseau neuronal INKA.

L'ensemble d'apprentissage utilisé.

Les paramètres de INKA ne concernent pas le décideur. Les valeurs de ces paramètres peuvent être déterminées de façon heuristique, comme cela est indiqué dans l'Annexe B. Dans tous les calculs indiqués, les paramètres sont constants mais ne sont pas optimisés. Les performances de INKA ne dépendent pas beaucoup des valeurs de ces paramètres dans le cadre du problème d'aide à la décision traité ici. L'analyse de sensibilité est donc étudiée uniquement pour l'ensemble d'apprentissage fourni par le décideur.

Le test de robustesse consiste à faire varier les valeurs de ces paramètres comme pour l'analyse de sensibilité, mais en étudiant plutôt la stabilité du classement obtenu pour les actions qui sont en tête du classement et qui sont donc susceptibles d'être choisies. Une action qui est en tête du classement mais qui a une très mauvaise stabilité peut souvent être un choix plus risqué qu'une action un petit peu moins bien classée, mais plus stable. Une action robuste est donc une action dont la place dans le classement reste stable dans le test de robustesse.

6.1 Analyse de sensibilité

L'analyse de sensibilité se fait uniquement pour l'ensemble d'apprentissage constitué d'exemples fournis directement par le décideur. La méthode retenue pour cette analyse est encore une version de « k -fold cross validation », sujet déjà abordé pour l'écart type de RMSE au Chapitre 5.1. Ici, au lieu d'utiliser l'écart type de RMSE comme indicateur de sensibilité, nous utiliserons directement la sensibilité du classement obtenu. La sensibilité est calculée comme pour RMSE, mais à partir de l'écart type des différences moyennes calculées à chaque étape du « k -fold cross validation ». L'évolution de cet indicateur de sensibilité en fonction du nombre d'exemples pour le choix de voiture est indiquée dans la Figure 6-1.

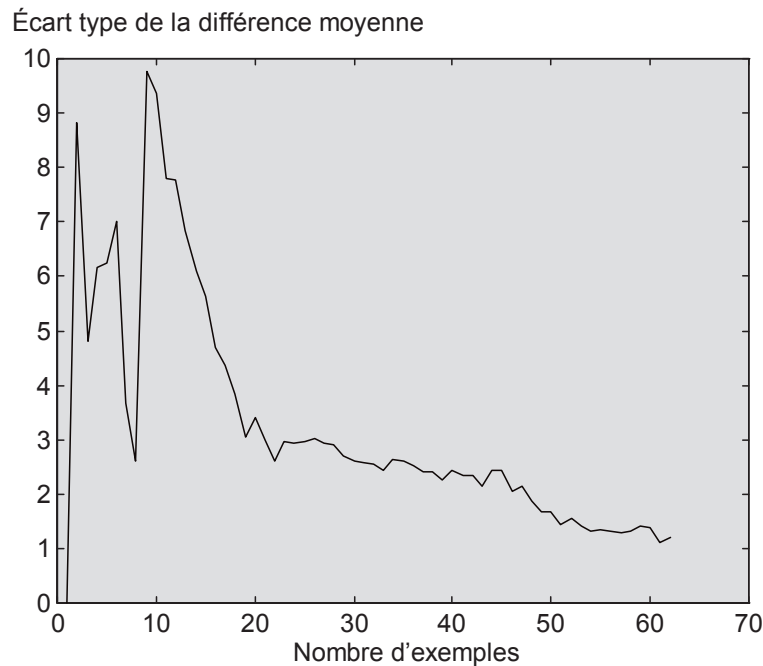


Figure 6-1. Evolution de l'écart type de la différence moyenne estimée par « leaving-one-out » pour l'application choix de voiture.

La sensibilité donnée par cet indicateur diminue si le nombre d'exemples augmente. Cette évolution est cependant assez irrégulière. Par exemple, l'introduction de l'exemple Citroën XM V6-24 augmente la sensibilité. Il est nécessaire d'introduire encore une dizaine d'exemples pour de nouveau diminuer la sensibilité. Un tel comportement est tout à fait normal et attendu et montre la validité de l'indicateur.

6.1.1 Pourcentage de stabilité

Il est possible de calculer un autre indicateur de la sensibilité, appelée ici *pourcentage de stabilité* d'un classement. Pour un nombre d'actions donné, on calcule la différence moyenne maximale possible entre deux classements. Cette différence moyenne est maximale pour deux classements inverses l'un de l'autre. Dans le cas de 113 voitures potentielles la différence moyenne maximale est de 56.5. Le pourcentage de stabilité est calculé par rapport à cette différence moyenne maximale:

$$ps = 1 - \frac{\bar{d}(O_1, O_2)}{\bar{d}_{\max}(O_1, O_2)} \quad (12)$$

où ps est le pourcentage de stabilité, $\bar{d}(O_1, O_2)$ est la différence moyenne et $\bar{d}_{\max}(O_1, O_2)$ est la différence moyenne maximale entre les deux classements O_1 et O_2 .

Un pourcentage de stabilité de zéro implique une sensibilité maximale et un pourcentage de 100% implique une sensibilité minimale, ce qui revient à dire que le classement ne change pas si l'on supprime n'importe quel exemple. L'évolution du pourcentage de stabilité pour le choix de voiture en fonction du nombre d'exemples est illustrée par la Figure 6-2.

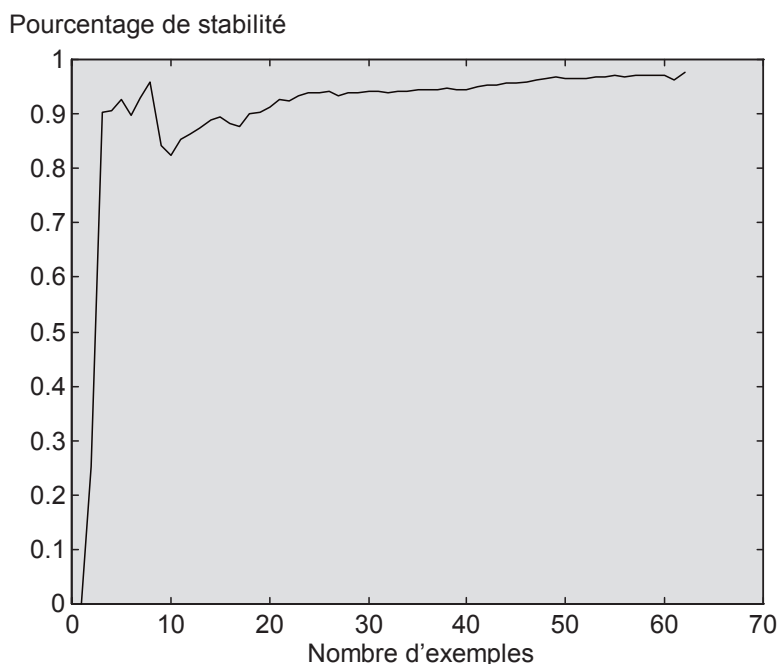


Figure 6-2. Pourcentage de stabilité en fonction du nombre d'exemples pour l'application de choix de voiture.

Ce graphe est en réalité le graphe inverse de la différence moyenne estimée par « leaving-one-out » dans la Figure 5-3. Une différence moyenne de zéro signifie un pourcentage de stabilité de 100% et une différence moyenne égale à $\bar{d}_{\max}(O_1, O_2)$ signifie un pourcentage de stabilité de zéro.

Le pourcentage de stabilité pourrait, tout comme la différence moyenne, être utilisé comme un indicateur de précision. C'est uniquement la normalisation par rapport à un indicateur de non-stabilité maximale qui le définit plutôt comme un indicateur de sensibilité. La distinction entre la précision estimée par le « k -fold cross validation », et la sensibilité, est souvent une question d'interprétation.

6.1.2 Sensibilité aux valeurs de préférence des exemples d'apprentissage

Une analyse de sensibilité complète devrait permettre d'estimer la sensibilité du modèle aux variations des valeurs de préférence des exemples d'apprentissage saisies par le décideur. Il faudrait faire varier ces valeurs de préférence tout en respectant le classement initial des « exemples réels » fournis par le décideur. Ce principe ressemble à celui utilisé dans la méthode QualiFlex [Pomerol & Barba-Romero, 1993], où les poids cardinaux des critères sont changés tout en respectant l'ordre d'importance donné par le décideur. Les changements que cela produit dans le classement des actions potentielles sont ensuite visualisés pour permettre au décideur de repérer les actions qui semblent rester le plus souvent en tête du classement.

Cette méthode serait tout à fait adaptable à l'approche neuronale présentée dans le cadre de ce travail. Pour un nombre d'actions potentielles important il devient cependant impossible de les visualiser toutes et il faudrait se limiter aux actions placées en tête du classement. Dans ce cas, il s'agit plutôt d'un test de robustesse qui nécessiterait un mode de traitement des valeurs de préférence cardinales des pseudo-exemples permettant de rester cohérent avec les modifications des valeurs de préférence des exemples réels.

6.2 Test de robustesse

Le test de robustesse est fait ici à partir des exemples de l'ensemble d'apprentissage en utilisant les mêmes principes de « k -fold cross validation ». Dans le cas d'un test de robustesse, il est important de connaître la stabilité du classement calculé, notamment pour les actions placées en tête du classement. Le décideur peut définir un nombre M d'actions en tête pour lesquelles il veut connaître la robustesse, afin de lui permettre d'identifier un « noyau » d'actions potentielles bonnes et stables à la fois.

Le choix de la valeur de M est fait par le décideur en fonction du nombre d'actions stables souhaitées. S'il suffit de choisir une seule action, il suffit de repérer l'action qui reste constamment parmi les M premières, pour laquelle il y a une forte probabilité pour qu'elle soit une des meilleures. Dans ce cas, il peut être avantageux de retenir cette action plutôt que continuer d'entrer d'autres exemples juste pour déterminer s'il existe une action qui serait encore un petit peu meilleure.

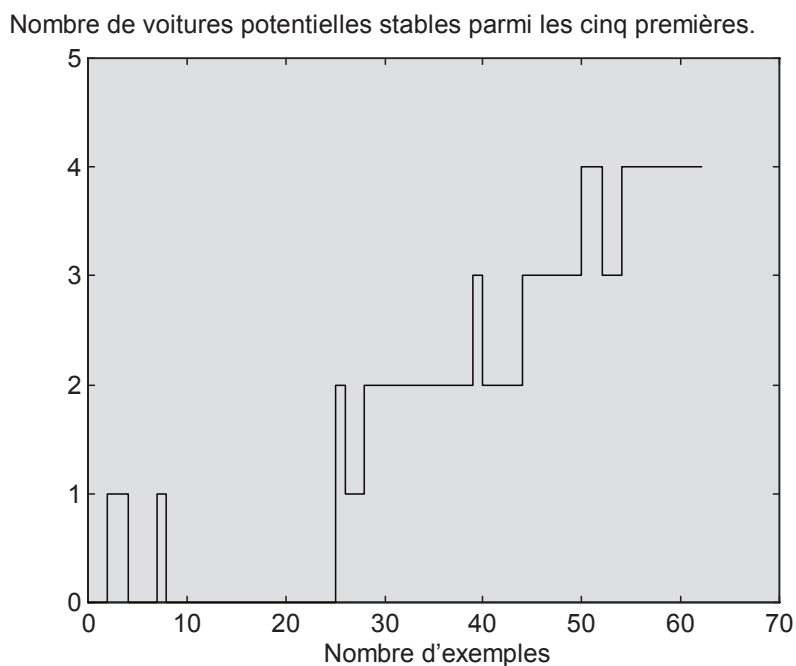


Figure 6-3. Nombre de voitures constamment parmi les cinq meilleures en fonction du nombre d'exemples utilisé pour le problème de choix de voiture.

L'évolution du nombre de voitures stables parmi les cinq premières en fonction du nombre d'exemples est illustrée par la Figure 6-3. Les quatre voitures « stables » sont finalement la SAAB 900 SE 2.0 Turbo, la Opel Vectra V6, la SAAB 900 SE 2.5 V6 et la Volvo 850 GLT 2.5. La première voiture « stable » qui apparaît ne doit pas être retenue car le nombre d'exemples traités est trop faible. Il est nécessaire de vérifier que l'action qui consiste à choisir cette voiture reste robuste après avoir introduit quelques exemples supplémentaires avant de prendre la décision. Il vaut également mieux attendre d'avoir de bonnes valeurs de précision et de sensibilité pour éviter de faire un choix trop rapide.

Un test de robustesse du même type pourrait bien sûr également se faire pour les valeurs de préférence des exemples d'apprentissage, au lieu de le faire uniquement pour l'ensemble des exemples. Il serait, par exemple, possible de générer plusieurs ensembles de valeurs de préférence respectant l'ordre donné par le décideur. Une telle analyse souffre malheureusement des mêmes problèmes qu'une analyse de sensibilité s'il s'agit de traiter les valeurs de préférence cardinales des pseudo-exemples.

7. Choix d'un site de stockage de déchets industriels ultimes

Le choix d'un site de stockage de déchets, quelle que soit la nature des déchets, est toujours un problème difficile à résoudre à cause du nombre important d'acteurs intervenant dans la résolution de ce problème complexe [Graillet et al., 1993]. Si la population locale est complètement contre le projet de stockage de déchets industriels, l'échec est immédiat. Si les facteurs économiques n'ont pas été pris suffisamment bien en compte, l'échec peut se produire dans les années qui suivent la décision. L'échec peut également être dû à la non prise en compte du système écologique, qui ne supporte pas les effets de l'implantation du site [Brodhag, 1992].

Il existe un grand nombre de travaux qui traitent des problèmes de ce type [Keeney, 1979], [Simos, 1990], [Frantzis, 1991], [OCDE, 1991], [Mierzwa et al., 1992], [Collette & Installé, 1993]. Ce type de problème est encore loin de trouver une solution et il y a un grand besoin de nouvelles méthodes d'aide à la décision pour les résoudre.

7.1 Données du problème

Le problème de choix de site traité dans ce chapitre est le choix d'un site de stockage de déchets industriels ultimes dans la région Rhône-Alpes en France [Meurs & Payen, 1992]. Cette étude faisait l'objet d'un projet de l'École des Mines de Saint-Étienne effectué pour l'ADEME (Agence de l'Environnement et de la Maîtrise de l'Énergie) et la SEMEDDIRA (Société d'Économie Mixte pour l'Étude de Décharges pour Déchets Industriels en Rhône-Alpes) en collaboration avec la Compagnie de Prospection Géologique Française CPGF Horizon [SEMEDDIRA, 1993]. Le SAD StocDech présenté dans l'Annexe C était développé et utilisé dans le projet. Les critères, les pondérations et les données utilisées viennent directement de ce projet réel, qui est décrit plus en détail dans l'Annexe A.

7.1.1 Critères de sélection

Quinze critères de sélection ont été utilisés pour choisir le site. Tous les critères de sélection sont quantitatifs. Les critères de sélection sont de nature très variée et leurs valeurs sont très peu corrélées entre elles, comme l'a montré une analyse en composantes principales. Il est pratiquement impossible de comprendre les valeurs des critères sans leur définition exacte, car elles sont calculées par des formules assez complexes, compréhensibles uniquement par l'expert initié au problème. Les critères de sélection et les poids utilisés pour la somme pondérée sont les suivants:

Chapitre 7. Choix d'un site de stockage de déchets industriels ultimes

Protection des captages d'alimentation en eau potable. Poids 19.

Caractéristiques du sous-sol. Poids 19.

Protection des aquifères. Poids 19.

Protection des zones agricoles labellisées ou d'Appellation d'Origine Contrôlée. Poids 5.8.

Proximité des zones de production de déchets. Poids 5.8.

Densité de la population. Poids 5.8.

Protection du patrimoine bâti. Poids 5.8.

Protection des zones écologiques. Poids 5.8.

Protection des zones touristiques. Poids 2.

Protection des zones agricoles aménagées. Poids 2.

Aménagement du territoire. Poids 2.

Proximité des réseaux de transport. Poids 2.

Nuisances liées au trafic. Poids 2.

Protection du paysage. Poids 2.

Protection des ressources forestières. Poids 2.

7.1.2 Modèle de décideur

Le nombre d'acteurs est important et il s'agit de décideurs « abstraits », ce qui rend difficile la modélisation de leurs préférences. L'apprentissage de ces préférences par des méthodes d'analyse de données ou d'identification est pratiquement la seule façon de prendre en compte l'avis de ces acteurs dans un SAD ou SIAD. Les données nécessaires pour effectuer cet apprentissage sont obtenues à l'aide de sondages ou d'autres types d'études.

Il était envisagé d'obtenir les préférences à partir de rapports concernant les onze sites de stockage de déchets industriels ultimes existants, mais il s'est avéré difficile de les obtenir. Il était alors nécessaire d'utiliser une fonction de préférence « modèle », qui permettait de

produire les données nécessaires à l'apprentissage. Une fonction par somme pondérée, combinée avec des fonctions d'utilité linéaires a été choisie.

7.1.3 Sélection et saisie des exemples d'apprentissage

Comme pour le choix de voiture, l'ensemble d'exemples d'apprentissage initial a été sélectionné à l'aide d'une carte auto-organisatrice de Kohonen. Douze exemples (Tableau 7-1) ont été choisis avec deux pseudo-exemples de sites idéal et anti-idéal.

Site	612	2773	2981	488	621	595	559	1415	2929	1220	1168	1358
Valeur préf.	69.8	58.3	62.2	62.2	68.0	78.1	54.8	75.5	65.1	48.1	64.0	67.9

Tableau 7-1. Valeurs de préférence calculées par la somme pondérée pour les exemples d'apprentissage du choix d'un site de stockage de déchets industriels ultimes.

Les valeurs de préférence utilisées pour l'apprentissage sont celles calculées par la somme pondérée. Il serait bien sûr possible de saisir des valeurs de préférence selon le principe de la MHM. Dans ce cas, les expériences du Chapitre 7 montrent qu'il n'y a pas trop de différence au niveau du classement des actions potentielles.

7.2 Évaluation des résultats obtenus par INKA

Les valeurs de préférence calculées par la somme pondérée étaient utilisées pour construire un premier SAD. Les dix premiers sites du classement par somme pondérée et celui donné par INKA après apprentissage sont indiqués dans le Tableau 7-2. Les sites indiqués en caractère gras appartenaient à l'ensemble d'entraînement.

(a) Classement somme p.		(b) Classement INKA		(c) Classement « pseudo »	
1416	83.10	595	78.01	1416	81.16
408	79.60	1477	75.71	595	78.91
856	78.60	1415	75.40	1477	78.90
287	78.20	734	75.31	596	78.22
595	78.10	1416	74.91	856	77.23
596	77.60	532	74.48	1415	77.11
1477	77.50	856	73.62	857	76.87
919	76.60	793	73.57	349	76.77
733	76.10	232	73.55	408	76.70
793	76.00	596	73.37	287	76.69

Tableau 7-2. Comparaison entre les classements obtenus par (a) la somme pondérée, (b) la fonction de préférence apprise avec 14 exemples, (c) la fonction apprise avec les 49 pseudo-exemples.

La méthodologie interactive a ici été utilisée pour augmenter la précision de la fonction de préférence apprise, ce qui a permis de corriger les erreurs évidentes en prenant soin qu'elles puissent l'être aussi par un homme d'étude pour le cas d'un décideur abstrait. Le classement obtenu à l'aide de la fonction de préférence apprise avec ce nouvel ensemble d'entraînement (49 pseudo-exemples) est également indiqué dans le Tableau 7-2. L'ensemble d'actions potentielles comporte 250 sites potentiels (en réalité des *secteurs*, le mot site est utilisé pour des raisons de simplicité), dont les noms sont simplement des chiffres codés.

Les résultats sont même meilleurs que pour l'application choix de voiture, grâce à la fonction de préférence linéaire qui est plus facile à apprendre correctement avec peu d'exemples. Par contre, l'apprentissage est rendu plus difficile par le nombre de critères de sélection plus élevé, leur manque total de corrélation et le nombre d'actions potentielles plus grand.

Parmi les 10 premiers sites du classement (b), il y a six sites en commun avec le classement (a). Pour les sites qui ne figurent pas dans les deux listes, les erreurs sont assez minimales, comme indiqué dans le Tableau 7-3. Comme pour l'application « choix de voiture », la proximité des actions potentielles par rapport aux exemples d'apprentissage a une grande influence sur leurs valeurs de préférence calculées. Tel est le cas pour le site 1415, qui est un des exemples d'apprentissage et qui est pour cette raison trop bien classé. Pour les secteurs 408, 287 et 919 il n'y a pas d'exemple d'apprentissage assez similaire. L'introduction de pseudo-exemples doit notamment permettre de corriger ce type de problèmes.

<i>Site</i>	<i>(a) Classement somme p.</i>	<i>(b) Classement INKA 1</i>	<i>(c) Classement « pseudo »</i>
408	2 ^{ème}	20 ^{ème}	9 ^{ème}
287	4 ^{ème}	21 ^{ème}	10 ^{ème}
919	8 ^{ème}	13 ^{ème}	13 ^{ème}
733	9 ^{ème}	17 ^{ème}	20 ^{ème}
793	10 ^{ème}	8 ^{ème}	12 ^{ème}
1415	11 ^{ème}	3 ^{ème}	6 ^{ème}
857	12 ^{ème}	12 ^{ème}	7 ^{ème}
349	13 ^{ème}	22 ^{ème}	8 ^{ème}
532	22 ^{ème}	6 ^{ème}	11 ^{ème}
232	23 ^{ème}	9 ^{ème}	16 ^{ème}
734	37 ^{ème}	4 ^{ème}	27 ^{ème}

Tableau 7-3. Liste des différences entre les classements obtenus par méthodes d'aide à la décision différentes.

Après l'introduction des pseudo-exemples, il y a 7 sites en commun entre le classement par somme pondérée et celui calculé par INKA. Le classement des dix premiers sites est plus proche du classement par somme pondérée que celui obtenu sans pseudo-exemples. On peut

constater la même chose pour les sites qui ne se trouvent pas parmi les dix premiers dans les deux classements, comme indiqué dans le Tableau 7-3. La fonction de préférence peut difficilement être mieux apprise puisque ces différences sont relativement peu importantes.

7.2.1 Indicateurs de précision et de sensibilité

L'utilisation de pseudo-exemples, dont le nombre peut être facilement augmenté, rend possible et intéressant le suivi de l'évolution de la précision et de la sensibilité de la fonction de préférence apprise. Le pourcentage de stabilité est le seul indicateur qui n'ait pas été pris en compte car il apporte peu d'information supplémentaire par rapport à la différence moyenne du classement.

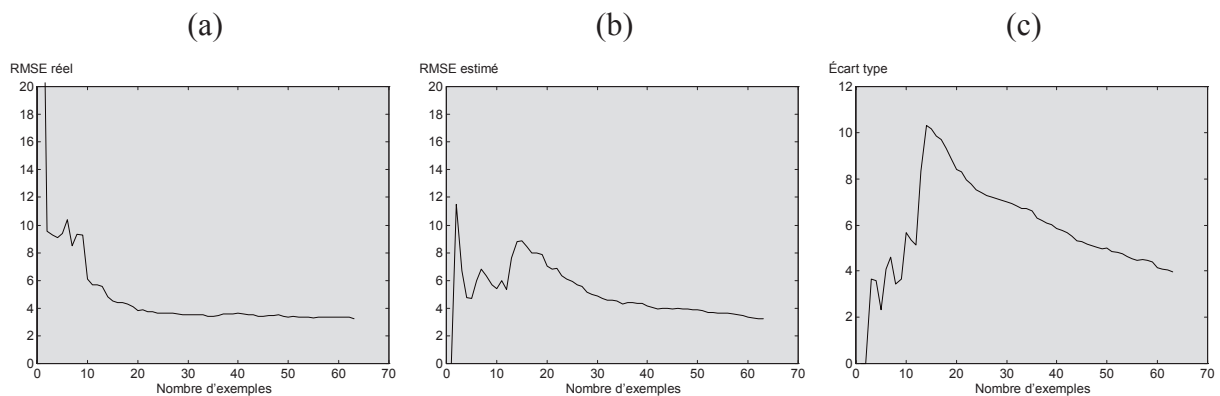


Figure 7-1. (a) RMSE réel entre le classement donné par la somme pondérée et le classement donné par INKA pour l'ensemble d'actions potentielles, (b) valeur moyenne de RMSE et (c) écart type de RMSE estimés par « leaving-one-out » pour le choix de site.

L'évolution de la précision numérique est présentée dans la Figure 7-1. La comparaison du RMSE estimé par « leaving-one-out » et du RMSE réel est plutôt réussie pour cette fonction de préférence. L'amélioration de l'estimation par rapport à celle obtenue pour le choix de voiture vient probablement de la linéarité plus grande de la fonction de préférence pour le choix de site.

Par contre, l'évolution de l'écart type montre qu'il y a encore une certaine sensibilité à la suppression d'exemples dans l'ensemble d'entraînement. La valeur de l'écart type est malgré tout assez basse, ce qui implique qu'il n'est pas forcément très utile de continuer à introduire de nouveaux pseudo-exemples pour la faire diminuer.

La Figure 7-2 indique l'évolution réelle de la différence moyenne entre le classement donné par la somme pondérée et celui calculé par INKA, avec l'estimation par « leaving-one-out ». Comme dans le cas de choix de voiture, la différence estimée est beaucoup trop petite. Cet indicateur est avant tout destiné à permettre au décideur de déterminer le moment d'arrêt de la procédure interactive. La différence moyenne estimée varie à peu près de la même façon que la différence moyenne réelle, même si elle est à un niveau plus bas, ce qui peut suffire pour estimer ce point d'arrêt. L'estimation de la différence maximale par « leaving-one-out » correspond mieux à la différence maximale réelle (Figure 7-3).

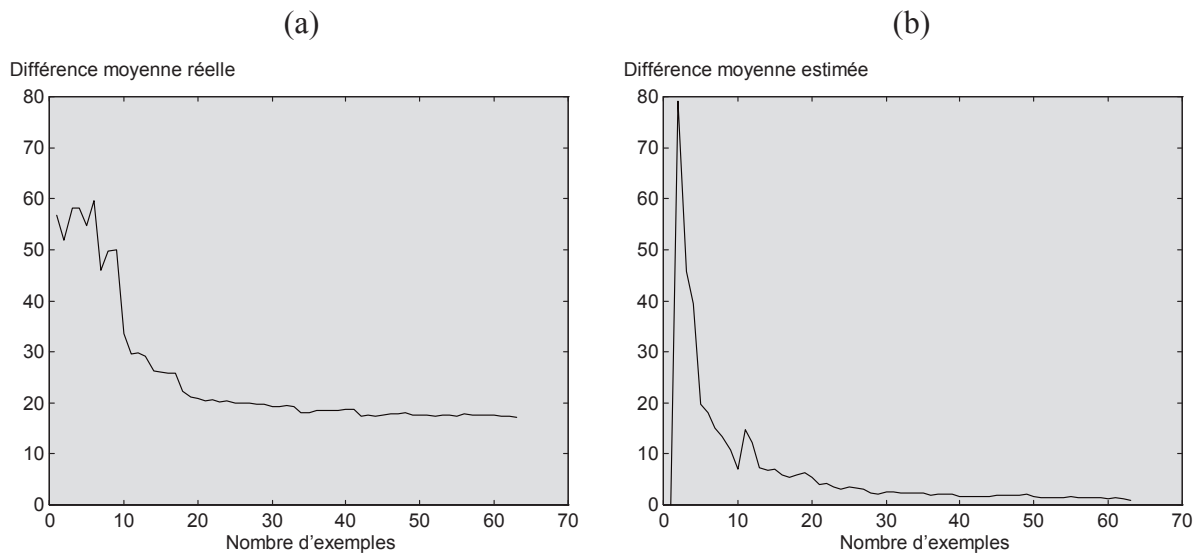


Figure 7-2. Comparaison entre (a) la différence moyenne réelle entre le classement donné par la somme pondérée et le classement donné par INKA et (b) la différence moyenne estimée par « leaving-one-out » pour le choix de site.

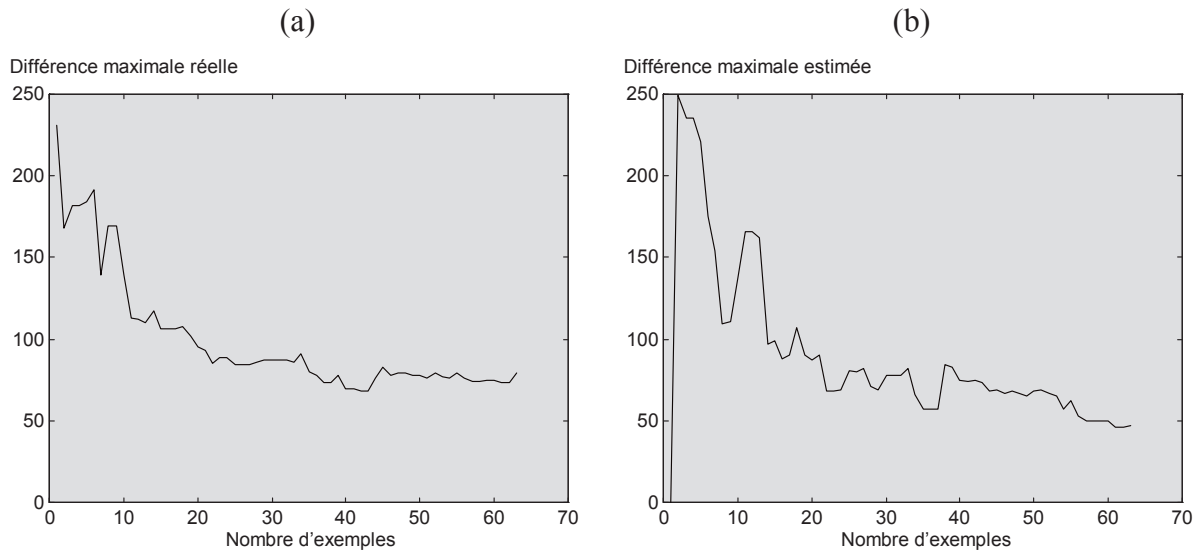


Figure 7-3. Comparaison entre (a) la différence maximale réelle entre le classement calculé par somme pondérée et le classement calculé par INKA et (b) la différence maximale estimée par « leaving-one-out » pour le choix de site.

Un moyen d'améliorer cette estimation pourrait être d'utiliser une valeur autre que une pour k dans le « k -fold cross validation », qui, dans ce cas, pourrait augmenter en fonction du nombre d'exemples. Plus le nombre d'exemples augmente, moins la suppression d'un entre eux a d'influence sur le classement. Une telle méthode serait forcément heuristique et difficile à utiliser dans tous les cas.

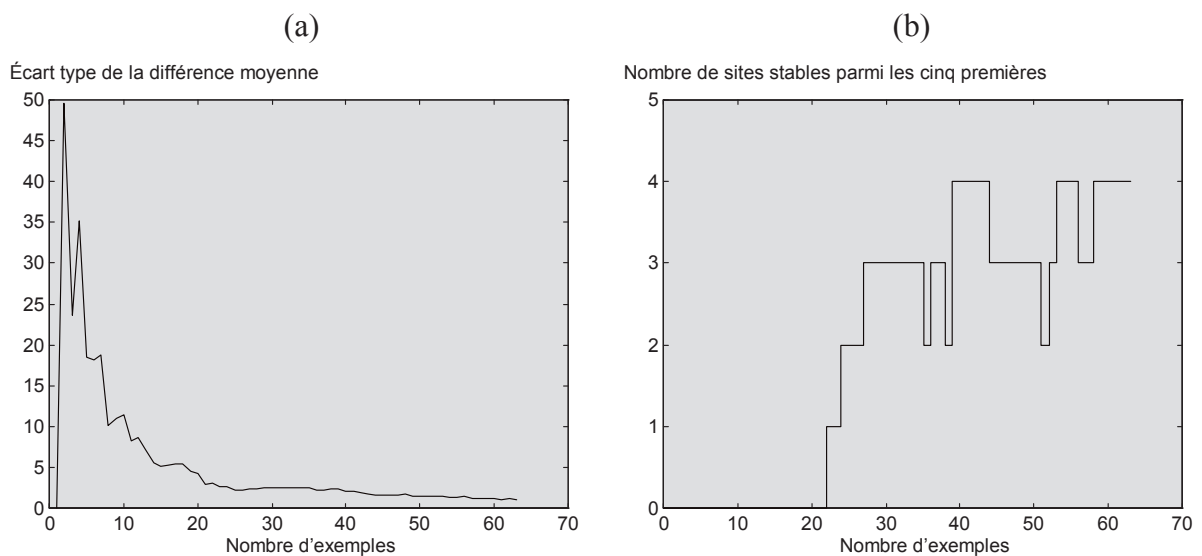


Figure 7-4. (a) Evolution de l'écart type de la différence moyenne donnée par « leaving-one-out » et (b) le nombre de sites constamment parmi les cinq meilleures.

L'analyse de sensibilité par écart type de la différence de place moyenne obtenue par « leaving-one-out » (Figure 7-4) montre que la sensibilité diminue assez peu à partir de vingt exemples. Bien que la différence moyenne soit calculée uniquement pour l'ensemble d'actions potentielles, nous pouvons constater que la sensibilité diminue bien avec l'ajout de pseudo-exemples, même si ces pseudo-exemples sont assez différents des actions potentielles. L'information apportée par les pseudo-exemples complète alors bien celle des exemples réels, permettant d'obtenir pratiquement le même classement même en supprimant des exemples réels.

Le test de robustesse est fait pour un « noyau » comportant 5 éléments. Les premières actions robustes apparaissent après plus de vingt exemples, au moment où la sensibilité de la fonction atteint un bon niveau. Finalement, on obtient 4 sites robustes (le 1416, le 595, le 1477 et le 596), qui sont tous en tête du classement calculé par somme pondérée. Ces quatre sites sont pour cette raison particulièrement intéressants à étudier de plus près. Une taille de noyau supérieure à 5 serait certainement utile pour permettre d'identifier davantage de sites stables (le 408, le 856 et le 287), qui seraient probablement inclus dans un noyau dimensionné à 10.

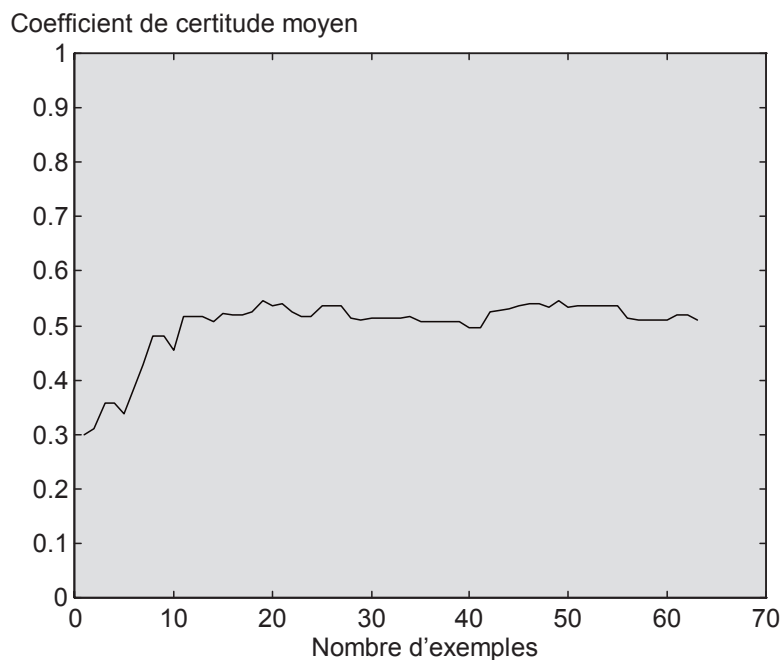


Figure 7-5. Évolution du coefficient de certitude moyen sur l'ensemble d'actions potentielles en fonction du nombre d'exemples pour le choix d'un site de stockage.

Le coefficient de certitude moyen de INKA varie en fonction du nombre d'exemples selon la courbe de la Figure 7-5. L'évolution de cet indicateur peut sembler plutôt décevante,

Chapitre 7. Choix d'un site de stockage de déchets industriels ultimes

puisqu'elle n'augmente pratiquement plus avec l'introduction de nouveaux pseudo-exemples. Ceci est normal puisque les pseudo-exemples sont introduits assez « loin » (au sens de distance Euclidienne) des actions potentielles. Puisque le coefficient de certitude moyen est calculé pour l'ensemble des actions potentielles, il est normal qu'elle n'augmente pas beaucoup. Si le coefficient de certitude moyen était calculé par simulation de Monte-Carlo, ou par une autre méthode qui couvre mieux tout l'espace d'entrée, il serait certainement meilleur.

8. Conclusion

Cette partie propose une méthodologie neuronale pour un système interactif d'aide à la décision multicritère. L'apprentissage automatique par réseau neuronal permet d'utiliser une fonction mathématique continue comme modèle de préférence. Ce type de modèle permet d'utiliser un grand nombre de méthodes d'interaction avec le décideur et des indicateurs mathématiques de précision et de sensibilité. L'interaction par visualisation graphique de la fonction de préférence permet au décideur d'analyser si la fonction apprise correspond effectivement à ses préférences. Elle permet également au décideur d'étudier ses propres préférences, qui peuvent changer au cours du processus d'interaction.

La correction de la fonction de préférence peut se faire directement par pseudo-exemples au fur et à mesure que le décideur découvre des erreurs ou qu'il modifie ses préférences. Les pseudo-exemples permettent de poursuivre le processus interactif aussi longtemps que le décideur le souhaite. Le seul critère d'arrêt est celui de la satisfaction du décideur, ce qui est très important pour l'utilisation pratique de la méthodologie. Dans le cas d'un décideur abstrait, il devient néanmoins difficile de connaître ce niveau de satisfaction. Les indicateurs de précision, de sensibilité et de robustesse permettent d'estimer un niveau de satisfaction, même s'il s'agit d'un décideur abstrait. Dans ce cas, l'interaction se fait avec l'homme d'études, qui est chargé d'estimer le niveau de satisfaction à l'aide de ces indicateurs.

La connaissance des préférences des décideurs abstraits permet d'analyser l'effet de la variation de certaines valeurs des critères de sélection sur le classement des actions potentielles. Un constructeur de voiture sera ainsi capable d'analyser s'il faut plutôt diminuer le prix de sa voiture ou augmenter le niveau d'équipement pour dépasser un concurrent dans les préférences d'une certaine clientèle.

Les résultats expérimentaux montrent que la quantité d'information à obtenir de la part du décideur est plus petite que dans la plupart de SIAD existants. Dans le cas d'une fonction de préférence linéaire, le nombre d'exemples nécessaire est à peu près le même que le nombre de critères. Le travail demandé est du même ordre que celui requis pour l'introduction des poids des critères. Dans le cas de fonctions de préférence non-linéaires, les pseudo-exemples facilitent la saisie des préférences du décideur. Le décideur a donc besoin de fournir un seul

Chapitre 1. Explications dans les systèmes experts

type d'information (les pseudo-exemples), qui sont faciles à introduire grâce à l'interface graphique. La réduction de la quantité d'information à obtenir de la part du décideur permet de traiter des problèmes plus complexes, tout en permettant des fonctions de préférence plus réalistes, comme les fonctions de préférence contextuelle.

Les nouvelles méthodes ou techniques développées dans le cadre de ce travail et présentées dans cette partie sont donc:

- L'analyse graphique de la fonction de préférence apprise, qui permet au décideur de vérifier si cette fonction est correcte ou pas.
- La notion de pseudo-exemples, qui permet au décideur de modifier et de corriger la fonction de préférence pour mieux correspondre à ses propres préférences.
- Une méthodologie interactive utilisant l'apprentissage automatique par réseaux de neurones. Il s'agit d'une des premières méthodologies d'aide à la décision multicritère interactives utilisant l'apprentissage automatique à partir d'exemples de décision.
- Des nouveaux indicateurs de précision, de sensibilité et de robustesse qui aident le décideur à estimer le point d'arrêt optimal de la procédure interactive.
- Des possibilités d'analyser les préférences d'un groupe d'acteurs pour pouvoir leur proposer des actions meilleures.

Le dernier aspect a également été développé dans le cadre d'une proposition de projet à l'Union Européenne pour mieux prendre en compte l'avis de tous les acteurs des problèmes de choix de site (de stockage de déchets ou autres). Il serait, par exemple, utile d'apprendre les préférences de la population locale pour pouvoir identifier les meilleurs compromis possibles avant de prendre une décision finale.

L'utilisation d'un SIAD dans de telles conditions nécessite évidemment des capacités d'explication de la part du SIAD. Malheureusement les SIAD existants négligent pratiquement complètement cet aspect. La troisième partie de ce mémoire sera consacrée à la présentation des méthodes qui permettent aux SIAD de fournir des explications.

Partie III

Explications dans les systèmes d'aide à la décision utilisant des réseaux de neurones

Il est étonnant de constater que l'aspect explication est presque complètement négligé dans les systèmes multicritère d'aide à la décision. Ceci vient peut-être du fait que c'est le décideur qui est censé fournir les explications. Puisque c'est lui qui a défini les poids et les autres paramètres utilisés par le SAD, il est le seul qui puisse justifier le choix des valeurs. Par ailleurs, puisque les SAD sont le plus souvent utilisés par le décideur lui-même, peu d'efforts ont été faits pour expliquer les recommandations d'un SAD.

Dans le domaine des systèmes experts, la situation est complètement différente. Un système expert est censé représenter les connaissances d'un expert sur un domaine limité. Il est destiné à aider des personnes possédant moins d'informations que lui. Afin que ces personnes puissent faire confiance au système expert, il faut que le système expert puisse fournir des explications sur ses recommandations. C'est également pour cette raison que le domaine des explications est actuellement intégré dans le domaine des systèmes experts et de l'intelligence artificielle.

Cette troisième partie est organisée de la façon suivante: après avoir examiné l'état de l'art des explications dans le Chapitre 1, le Chapitre 1 traite des insuffisances des systèmes d'explication actuels. L'application au choix de sites de stockage de déchets industriels ultimes, pour laquelle le SAD StocDech a été développé, montre la nécessité des explications dans les SAD. Le grand nombre de décideurs concernés, le grand nombre de sites potentiels et la complexité des critères de sélection ne permettent pas au décideur d'expliquer facilement ses décisions. Par ailleurs, la nature du problème rend les explications nécessaires pour

Chapitre 1. Explications dans les systèmes experts

justifier auprès de la population locale le choix du site retenu qui recevra les déchets industriels de la région.

Il peut sembler étonnant de proposer un SIAD fondé sur les réseaux de neurones. Les réseaux neuronaux sont souvent critiqués justement pour leur aspect « boîte noire » (en anglais: *black box*), qui signifie qu'ils sont incapables de justifier leurs recommandations. Les concepts d'importance et d'utilité contextuelles introduits dans le Chapitre 3 permettent de combler cette lacune pour améliorer les capacités d'explication des réseaux neuronaux.

Les stratégies permettant de construire des explications à partir de ces concepts sont détaillées dans le Chapitre 4. Ces stratégies sont, au Chapitre 5, appliquées au choix de voiture. Le Chapitre 6 décrit le lien qu'il y a entre ces stratégies et les systèmes explicatifs existants pour pouvoir combiner leurs avantages le mieux possible.

Enfin, le dernier chapitre (Chapitre 7), présente les améliorations envisagées autour du thème de la décision négociée. L'outil ADELAIDE présenté dans ce chapitre est en quelque sorte un cahier de charges pour un logiciel intégrant tous les composants de la méthodologie SIAD, y compris les explications.

1. Explications dans les systèmes experts

L'*explicabilité* est la capacité d'un système à expliquer ses décisions d'une façon compréhensible, exhaustive et convaincante, pour convaincre que la décision proposée est bonne. L'explicabilité des systèmes de décision est un problème qui est identifié depuis la création du système expert de diagnostic médical MYCIN [Buchanan & Shortliffe, 1984].

1.1 Systèmes experts de première génération

Les systèmes experts de première génération utilisent presque exclusivement des règles de production comme méthode de résolution. Les explications données sont fondées directement sur les règles et les inférences effectuées. Les *phrases à trou* sont le moyen le plus utilisé pour fournir des explications en langue naturelle d'un niveau simple. Guru [Micro Data Base Systems, 1990] constitue un bon exemple de systèmes experts de cette génération.

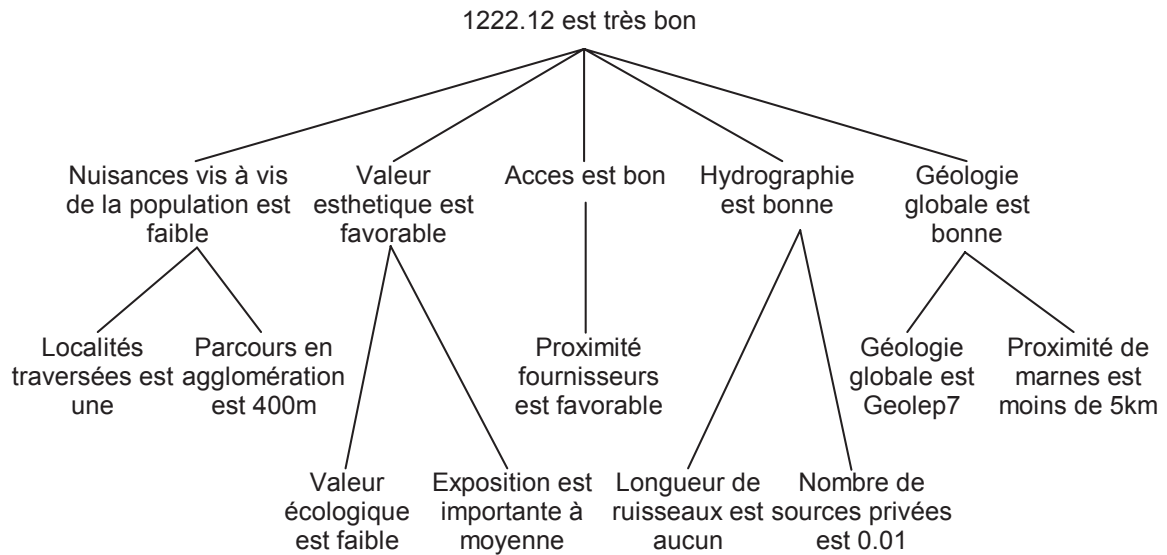


Figure 1-1. Exemple d'un arbre d'inférence pour classer un site de stockage de déchets en Suisse.

C'est la trace des inférences qui est le plus souvent utilisée pour expliquer les conclusions du système de règles. La Figure 1-1 montre un exemple d'un arbre d'inférence pour classer un site de stockage de déchets. L'utilisateur peut poser au système des questions du type *comment* et *pourquoi*.

La question « comment » est posée quand l'utilisateur veut savoir quelles sont les règles qui ont permis de déduire la valeur d'un nœud (un *concept intermédiaire*) de l'arbre d'inférence. La question « pourquoi » est posée par l'utilisateur pour connaître les raisons pour lesquelles une règle a été activée. Le système peut indiquer les hypothèses qu'il a cherchées à vérifier à l'aide de la règle.

L'enchaînement des questions « comment » et « pourquoi » permet d'exploiter tout l'arbre d'inférence niveau par niveau. Ce type de mécanisme n'est pas très efficace et les deux questions servent souvent à programmer des interfaces plus conviviales, comme l'interface de StocDech (Chapitre 1). L'insuffisance des questions « comment » et « pourquoi » est maintenant communément admise [Brezillon, 1992], [Safar, 1992].

Dans les systèmes experts classiques, il y a normalement un seul arbre d'inférence. Si d'autres types de logique sont utilisés, comme la logique floue, plusieurs résultats simultanés peuvent alors posséder des degrés de certitude différents. Il peut également y avoir plusieurs arbres d'inférence correspondant à chaque résultat. Dans ce cas, la valeur explicative d'une récapitulation des règles utilisées est petite.

Chapitre 1. Explications dans les systèmes experts

Puisque les règles sont la seule source d'information utilisable pour la construction des explications, il est nécessaire que le vocabulaire utilisé dans les règles soit compréhensible pour l'utilisateur. Le concepteur de la base de règles est donc obligé de définir une représentation claire de la base de connaissances. Il faut également que l'utilisateur soit familier avec les principes de raisonnement utilisés pour comprendre les explications. Par contre, cette approche ne demande pas beaucoup de travail supplémentaire pour pouvoir produire des explications une fois la base de règles construite.

L'expression de la connaissance sous forme de règles est souvent trop détaillée et trop technique et ne permet pas de comprendre facilement l'ensemble du raisonnement. C'est comme si on essayait d'expliquer ce qu'est une voiture en expliquant chacun de ces éléments (la roue, le piston, ...). Un autre inconvénient porte sur la difficulté à distinguer les règles exprimant la connaissance sur le domaine de la connaissance propre au contrôle de l'inférence. Les premières sont davantage utilisées pour fournir des explications à l'utilisateur. L'apport principal des systèmes experts de deuxième génération permet de résoudre une partie de ces problèmes.

1.2 Systèmes experts de deuxième génération

Les systèmes experts de deuxième génération essaient de prendre en compte a priori les besoins des explications. Ils contiennent pour cette raison beaucoup d'informations supplémentaires par rapport au domaine d'application et au raisonnement utilisé. Il y a également d'autres méthodes d'inférence que les règles, qui permettent souvent de fournir des explications plus facilement. Les réseaux sémantiques en sont un exemple, décrits par exemple dans [Genesereth & Nilsson, 1987].

Dans les systèmes experts de deuxième génération, il est possible et souhaitable de faire une séparation entre le *module de raisonnement* et le *module d'explication*. Le module de raisonnement se charge de déterminer le résultat du système expert et le module d'explication d'expliquer ce résultat. Le meilleur degré d'intégration de ces deux modules est une des questions importantes du domaine. Il existe un grand nombre de systèmes représentant des divers degrés d'intégration. Swartout et Moore [Swartout & Moore, 1993] utilisent quatre systèmes experts comme exemples pour illustrer l'évolution: *NEOMYCIN*, *Generic Tasks*, *EES* et *REX*.

Le système NEOMYCIN [Clancey, 1983] est développé sur les bases de MYCIN. Clancey observait qu'il n'y avait pas de structure dans la base de règles de MYCIN, ce qui rendait difficile l'identification des étapes suivies et les sous-problèmes résolus pendant l'inférence. La solution retenue pour ce problème était de regrouper les règles dans des groupes d'inférence, qui avaient comme objectif de résoudre un certain sous-problème. Certains groupes d'inférence sont également réservés à quelques tâches de contrôle de l'inférence. La communication entre les groupes d'inférence est assurée par des méta-règles, qui contrôlent l'ordre d'activation et la synchronisation des groupes d'inférence. Cette structuration permet d'obtenir des explications plus générales et de plus haut niveau qu'avec MYCIN. Le même type de structuration est utilisé dans l'outil de développement de systèmes experts Nexpert Object [Neuron Data, 1991].

Le système *Generic Tasks* [Chandrasekaran, 1986] utilise de l'information sur des tâches de résolution communes à tout système expert pour définir les stratégies d'explication. Un vocabulaire type et des opérateurs dédiés à l'explication du déroulement de chaque tâche sont définis. La classification heuristique, l'abduction, ..., sont des exemples de tâches qui sont de quelque façon « compris » par le système d'explication et qui peuvent ainsi être facilement expliquées. L'obtention d'explications sur ces tâches demande ainsi peu de travail supplémentaire de la part du cogniticien.

Un grand problème des règles de production est qu'elles présentent uniquement des connaissances factuelles, sans information sur leur raison d'être. Le *Explainable Expert System Framework* (EES) [Neches et al., 1985] permet d'intégrer notamment ce type d'information. EES construit sa base de règles sur de l'information de haut niveau, qui est ensuite détaillée de plus en plus. Cette procédure permet de garder l'information sur le raisonnement utilisé pour aboutir à chaque règle. Il est ainsi possible d'expliquer ce raisonnement aussi bien que le raisonnement de la base de règles.

Dans le système REX [Wick & Thompson, 1992], les modules de raisonnement et d'explication sont encore plus séparés. Le principe utilisé est celui d'*explication reconstruite*, où le module d'explication essaye de construire une histoire « croyable » pour motiver le résultat à partir de l'information transmise du module de raisonnement. Cette approche permet au module de raisonnement d'être complètement optimisé pour la résolution du problème et au module d'explication de se concentrer sur les aspects linguistiques et besoins spécifiques aux explications. Le couplage entre les deux modules est effectué à l'aide d'un filtre

Chapitre 1. Explications dans les systèmes experts

paramétrable, qui permet d'obtenir des explications de plusieurs niveaux de détail qui suivent de plus ou moins près le fonctionnement du module de raisonnement.

Dans le domaine des systèmes experts, la tendance à développer des modules d'explication de plus en plus complexes et indépendants est observée. Pour cette raison, les travaux récents se concentrent sur la construction de tels systèmes, notamment pour les aspects linguistiques et rhétoriques [McKeown, 1985], [Mann & Thompson, 1988]. Les connaissances des utilisateurs sont également de mieux en mieux prises en compte pour leur fournir des explications adaptées. Pour le faire, il est nécessaire d'utiliser des *modèles d'utilisateur* [Sleeman, 1985], [Paris, 1988], [Kass, 1991], [Wu, 1991], qui permettent de connaître le niveau de connaissances de chaque utilisateur et de suivre son évolution.

Les efforts du groupe de travail français AIDE [Kassel, 1995] sont un bon exemple des tendances dans le domaine d'explication par systèmes experts. Le système AIDE permet une structure modulaire avec plusieurs modules de raisonnement, des modules pour l'interface en langage naturel, les modèles d'utilisateur, etc. [Greboval, 1994], [Perréard et al., 1995]. Il existe également des modules dédiés pour les connaissances spécifiques au domaine.

Dans le cas de raisonnement par réseaux neuronaux, comme pour les systèmes experts de deuxième génération, le module d'explication doit être séparé du module de raisonnement. Malgré tout, ceci n'est pas le cas pour la plupart des approches utilisées pour expliquer les résultats des réseaux de neurones. Presque toutes les approches existantes essaient d'extraire des règles à partir des poids, des valeurs d'activation et de sortie des réseaux neuronaux [Diederich, 1992], [Främling, 1992, 1993], [Sestito & Dillon, 1992], [Giles & Omlin, 1993a, b], [Tresp et al., 1993], [Andrews & Geva, 1994], [Fu, 1994].

Comme nous l'avons déjà constaté, les règles ne sont pas vraiment le meilleur moyen pour fournir des explications. Les approches utilisées dans le passé pour les extraire des réseaux de neurones sont peu généralisables au niveau des types de réseaux de neurones et des applications. Une approche d'explication reconstruite semble être la meilleure possibilité pour résoudre le problème. L'approche présentée dans le Chapitre 3 est fondée sur ce principe.

Un inconvénient d'AIDE que l'on trouve dans un grand nombre de systèmes experts de première ou deuxième génération est le manque presque complet de traitement de l'incertitude. Il a néanmoins été constaté par exemple dans [Bouri et al., 1990b], que la prise

en compte de l'incertain augmente la gamme des questions auxquelles un système expert peut répondre:

- Comment, pour l'essentiel, une telle distribution de possibilité a-t-elle été obtenue?
- Quels sont les faits qui jouent un rôle déterminant dans la mauvaise qualité de la réponse?
- Comment varierait la distribution de probabilité de la conclusion en fonction de la distribution de probabilité de tel ou tels faits?

Les fonctionnalités d'explication expliquées dans le Chapitre 3 permettent de traiter l'incertitude et, notamment, de répondre à ces trois questions.

1.3 Critères d'évaluation des explications

Il est communément admis que pour être capable de convaincre un décideur humain par des explications en langage naturel, il faut que ces explications soient construites suivant la même logique que celles utilisées par un humain qui explique son raisonnement. Malgré l'évolution technologique des systèmes experts dans cette direction, il reste encore à définir ce qu'est une explication et comment elle doit être construite. Un certain nombre de critères d'évaluation des explications a pour cette raison été proposé [Moore & Paris, 1991] pour estimer quand une explication est bonne ou mauvaise. Les critères suivants sont retenus de [Swartout & Moore, 1993]:

Fidélité. L'explication doit correspondre au raisonnement utilisé en réalité par le système de raisonnement. Si le module d'explication est trop déconnecté du module de résolution, les explications données risquent d'être fausses. Il peut même être pire de donner une telle explication que de ne rien expliquer.

Compréhensibilité. Il est évident qu'une explication doit être compréhensible par l'utilisateur pour être utile. La compréhensibilité se décompose en:

- *Terminologie utilisée.* L'utilisateur doit être familier avec les termes utilisés dans l'explication.
- *Sensibilité à l'utilisateur.* Les explications doivent prendre en compte les connaissances, objectifs, préférences et intérêts de l'utilisateur.

Chapitre 1. Explications dans les systèmes experts

- *Abstraction.* Le vocabulaire utilisé doit varier en fonction de l'interlocuteur. En diagnostic médical, par exemple, le système doit être capable d'expliquer le problème d'un patient comme infection bactérienne ou, plus scientifiquement, comme une infection *e. coli*.
- *Niveau de détail.* L'explication ne doit pas contenir trop d'information. Pour le choix de voiture, par exemple, un premier niveau d'explication consiste à dire que la voiture est bonne parce qu'elle a des bonnes performances. Le deuxième niveau consiste à préciser que la puissance, la vitesse maximale et l'accélération sont très bonnes. L'accès à ces deux niveaux se fait par le dialogue avec l'utilisateur.
- *Perspectives.* Le système doit pouvoir expliquer des choses de plusieurs points de vue, comme forme versus fonction dans le domaine de la biologie, par exemple.
- *Compétences linguistiques.* Le langage utilisé dans les explications doit être naturel au niveau de la grammaire et de la lisibilité. Ceci implique également de prendre en compte ce qui a déjà été expliqué et ce que l'utilisateur doit déjà savoir. Sinon, il risque d'apparaître dans le dialogue des répétitions, qui sont assez pénibles pour l'utilisateur.
- *Interactivité.* Prise en compte des réactions de l'utilisateur dans les cas où il n'a pas compris une partie de l'explication pour mieux expliquer les parties en question.

Complétude. Le système doit avoir suffisamment de connaissances pour répondre aux questions des utilisateurs, même concernant son propre fonctionnement. Pour être capable de le faire, il est nécessaire de prendre en compte les besoins d'explication des futurs utilisateurs déjà au début de la conception du système expert.

- *Explications sur le comportement du système.* Cet aspect concerne des explications sur les modalités de résolution du problème, comment une entrée affecte le résultat, et les effets des changements dans les valeurs d'entrée sur la sortie.
- *Justifications.* Il s'agit d'expliquer pourquoi le système a choisi le chemin de raisonnement utilisé et de justifier le choix de telle ou telle hypothèse à un moment donné.
- *Préférences.* Explications comparatives permettant de dire pourquoi une action est meilleure qu'une autre.

- *Explications sur le domaine.* Il s'agit de connaissances supplémentaires sur le domaine qui ne sont pas liés à la résolution même du problème, mais qui permettent de mieux expliquer à l'utilisateur le domaine en général.
- *Définition de terminologie.* Un « dictionnaire » du domaine.

Faible niveau de travail supplémentaire (en anglais: *low construction overhead*). Le travail nécessaire pour donner des explications par rapport à un système plus simple ne doit pas être trop important et ne doit pas conduire à une acquisition de connaissances trop longue ou une programmation trop difficile.

Efficacité. Avant tout, il ne faut pas que les performances de résolution soient dégradées, mais le temps nécessaire à la production des explications ne doit pas être trop long non plus.

Les systèmes experts présentés au chapitre précédent permettent de fournir des explications qui satisfont de mieux en mieux à ces exigences. Par contre, ces critères sont malheureusement un peu trop focalisés sur le langage naturel comme seul moyen d'explication. Les ordinateurs permettent une grande multitude d'autres moyens d'explication, comme les images, les graphes, les films, le son, etc. Les critères proposés donnent néanmoins pour l'instant une base de comparaison entre des systèmes explicatifs.

2. Explications dans StocDech

StocDech a été utilisé comme système d'aide à la décision pour le problème de choix d'un site de stockage de déchets industriels ultimes dans la région Rhône-Alpes [Främling & Graillot, 1994]. Il s'agit d'un problème très délicat où il est indispensable d'expliquer les recommandations du SAD au décideur afin de lui fournir toutes les informations pouvant motiver sa décision finale. Les explications fournies par le SAD doivent également permettre au décideur d'expliquer et de motiver cette décision auprès des autres acteurs du problème. Cet aspect est une des originalités de StocDech.

La présence de deux types d'approches d'aide à la décision, le système expert et l'analyse multicritère, nécessite également deux approches d'explication. Les explications données sur les résultats du système expert sont, malgré tout, du type classique pour un système expert de

première génération. Pour les méthodes d'analyse multicritère, il est nécessaire d'utiliser une approche assez différente, car le raisonnement n'est pas du tout le même.

2.1 Explications des résultats de la base de règles de StocDech

L'approche adoptée dans StocDech permet à l'utilisateur soit d'exploiter l'arbre d'inférence entier, soit de demander une explication sur la valeur déduite d'un concept intermédiaire. Le choix du concept intermédiaire à expliquer se fait par l'interface graphique, à l'aide d'une souris. La Figure 2-1 montre un exemple d'explication donnée par StocDech pour tout l'arbre d'inférence de la Figure 1-1.

1222.12 est très bon

Le/la résultat final a été trouvé(e) très bon parce que les critères suivants:

- Nuisances vis à vis la population est faible
- Valeurs esthétiques est favorable
- Accès est bon
- Géologie globale est bonne
- Hydrographie est bonne

Le/la Nuisances vis à vis la population a été trouvé(e) faible parce que les critères suivants:

- Localités traversées est une
- Parcours en agglomération est 400

Le/la Valeurs esthétiques a été trouvé(e) favorable parce que les critères suivants:

- Valeur écologique est faible
- Exposition est importante à moyenne

Le/la Accès a été trouvé(e) bon parce que les critères suivants:

- Proximité fournisseurs est favorable

Le/la Géologie globale a été trouvé(e) bon parce que les critères suivants:

- Géologie globale est geolep 7
- Proximité de marnes est moins de 5 km

Le/la Hydrographie a été trouvé(e) bon parce que les critères suivants:

- Longueur de ruisseaux est aucun

- Nombre de sources privées est 0.01

Figure 2-1. Exemple d'une explication fondée sur la base de règles, produite par StocDech.

Ce sont des explications qui répondent aux questions « comment » et « pourquoi ». Le niveau linguistique pourrait certainement être amélioré, ce qui n'empêche pas les explications d'être bien compréhensibles. Il est également possible d'obtenir une définition de chaque critère et de chaque concept intermédiaire.

2.2 Explications des résultats des méthodes d'analyse multicritère

Il y a très peu d'informations disponibles pour construire des explications sur les résultats donnés par les méthodes de somme pondérée et d'Electre I. Les seules informations disponibles sont les poids des critères et les fonctions d'utilité données par le décideur. Cette information peut sembler très pauvre par rapport à celle de la base de règles. Il est malgré tout possible de donner des explications en utilisant uniquement ces deux types d'information, comme nous allons le voir dans les sections suivantes.

2.2.1 Importance et utilité comme éléments d'explication

Le poids d'un critère correspond directement à l'importance accordée à un critère par le décideur. La fonction d'utilité permet de déterminer si la valeur d'un critère est bonne ou mauvaise. Les poids et les utilités sont des valeurs quantitatives, qui sont difficiles à comprendre et, pour cette raison, difficiles à utiliser dans les explications. Pour résoudre ce problème, StocDech effectue une transformation quantitative-qualitative de l'importance et des valeurs d'utilité par des intervalles de valeurs. Pour effectuer la transformation de l'importance, il est d'abord nécessaire de normaliser les poids par l'opération:

$$pn_i = p_i / \sum_{j=1}^N p_j \quad (13)$$

où pn_i est le poids normalisé du critère i , p_i et p_j sont les poids des critères i et j et N est le nombre de critères de sélection. Les intervalles de valeurs sont définis par rapport à cette valeur normalisée (Tableau 2-1).

Pour réaliser la transformation quantitative-qualitative de l'utilité, les valeurs d'utilité sont d'abord normalisées dans l'intervalle $[0,1]$. La normalisation se fait par l'opération:

$$gn_j(a_i) = \frac{g_j(a_i) - \text{minval}}{\text{maxval} - \text{minval}} \quad (14)$$

où $gn_j(a_i)$ est la valeur d'utilité normalisée de l'action i pour le critère j , $g_j(a_i)$ est la valeur d'utilité de l'action i pour le critère j , minval est la valeur d'utilité minimale et maxval la valeur d'utilité maximale. Les intervalles de valeurs utilisés pour les valeurs d'utilité sont indiqués dans le Tableau 2-1.

Le choix des limites des intervalles a une grande influence sur les explications données par le système. Pour cette raison, il faut que les limites soient plus ou moins paramétrables pour correspondre aux besoins de chaque application. Dans la plupart des cas il suffit d'utiliser des valeurs par défaut. Le choix de valeurs qualitatives pour l'importance pose plus de problèmes, car les intervalles devraient être au moins ajustés en fonction du nombre de critères. La détermination de cette procédure d'ajustement reste encore à étudier et à préciser en fonction des expériences pratiques.

<i>Importance qualitative</i>	<i>Intervalle d'importance</i>	<i>Utilité qualitative</i>	<i>Intervalle d'utilité</i>
Pratiquement déterminant	[0.75, 1.0]	Très bonne	[0.95, 1.0]
Presque déterminant	[0.5, 0.75[Bonne	[0.80, 0.95[
Vraiment très important	[0.25, 0.5[Assez bonne	[0.60, 0.80[
Très important	[0.15, 0.25[Mauvaise	[0.30, 0.60[
Important	[0.09, 0.15[Très mauvaise	[0, 0.30[
Assez important	[0.05, 0.09[
Peu important	[0.04, 0.05[
Très peu important	[0.03, 0.04[
Sans importance	[0, 0.03[

Tableau 2-1. Intervalles de valeurs utilisés pour la transformation quantitative-qualitative de l'importance et de l'utilité dans StocDech.

2.2.2 Explications sur les résultats de la somme pondérée

Un exemple d'explication s'appuyant sur la méthode de la somme pondérée est présenté dans la Figure 2-2. Les intervalles de valeurs et les valeurs qualitatives du chapitre précédent ont été utilisées pour la transformation quantitative-qualitative des importances et des utilités.

Cette explication montre la différence entre la valeur d'un critère et sa valeur d'utilité. Ainsi le critère « Aptitude agricole » dont la valeur est « Excellente » offre une valeur d'utilité « mauvaise ». La différence est encore plus claire pour les critères ayant des valeurs quantitatives.

L'explication donnée concerne le même site (1222.12) que l'explication à partir de la base de règles fournie dans la Figure 2-1. Il est difficile de dire laquelle des deux explique mieux le résultat obtenu, un résultat qui est d'ailleurs un peu différent pour les deux méthodes. Il est

vrai, certes, que l'explication utilisant les règles est plus structurée grâce aux concepts intermédiaires. Il est néanmoins possible d'utiliser les mêmes concepts intermédiaires dans l'explication de la somme pondérée, comme indiqué dans le Chapitre 3.3.

Le secteur 1222.12 est assez bon(ne) (note 73 sur 100).

Le secteur a une très bon(ne) valeur pour les critères:

Valeur écologique (Faible) , qui est un critère *important* (9.50%)

Parcours RN-Secteur (3.1) , qui est un critère *assez important* (5.00%)

Longueur de ruisseaux (Aucun) , qui est un critère *peu important* (4.50%)

Proximité fournisseurs de déchets (Favorable) , qui est un critère *peu important* (4.50%)

Routes à construire. (Aucun) , qui est un critère *peu important* (4.00%)

Proximité de marnes (Moins de 5 km) , qui est un critère *très peu important* (3.50%)

Le secteur a une bon(ne) valeur pour les critères:

Localités traversées (Une) , qui est un critère *important* (12.50%)

Parcours en agglomération (400) , qui est un critère *important* (10.00%)

Le secteur a une mauvais(e) valeur pour les critères:

Aptitude agricole (Excellente) , qui est un critère *très important* (15.00%)

Proximité d'un chemin de fer (Aucun raccordement) , qui est un critère *assez important* (5.00%)

Le secteur n'a pas de très mauvais(e)s valeurs.

Les valeurs des deux critères *les plus importants* (Aptitude agricole et Localités traversées) sont mauvais(e) et bon(ne)

Figure 2-2. Exemple d'une explication sur la somme pondérée d'un site suisse, produite par StocDech. Les mots en italique indiquent l'importance et les mots soulignés indiquent l'utilité.

Malheureusement, le problème majeur reste la justification du poids ou de la fonction d'utilité d'un critère, car c'est une information que le décideur a donnée et qu'il est le seul à connaître. Ceci est un problème du même type que celui évoqué pour les systèmes experts - comment justifier les connaissances introduites par le décideur quand il n'est plus disponible?

2.2.3 Explications sur les résultats de la méthode Electre I

La méthode Electre I (et d'autres versions d'Electre) permet de fournir deux types d'explications. Premièrement, il y a les explications sur les raisons de surclassement ou de

non-surclassement d'une action potentielle. Deuxièmement, il y a les explications comparatives entre deux actions potentielles.

1222.12 est surclassé par 7 secteurs (1184.01, 1202.03, 1203.02, 1222.13, 1223.01, 1242.07, 1242.09).

Le surclassement est en particulier due aux valeurs suivantes:

Le secteur a une mauvais(e) valeur pour les critères:

Aptitude agricole (Excellente) , qui est un critère *très important* (15.00%)

Proximité d'un chemin de fer (Aucun raccordement) , qui est un critère *assez important* (5.00%)

Le secteur n'a pas de très mauvais(e)s valeurs.

Figure 2-3. Exemple d'une explication sur le surclassement ou non-surclassement d'un site de stockage de déchets Suisse par Electre I, produite par StocDech. Les mots en italique indiquent l'importance et les mots soulignés indiquent l'utilité.

L'explication donnée par la méthode Electre I est identique à celle construite pour la somme pondérée; la seule information supplémentaire concerne le surclassement de l'action. Pour un secteur qui n'est pas surclassé, il est plus intéressant de donner une explication fondée sur les critères pour lesquels les valeurs sont bonnes, mais pour un secteur surclassé c'est le cas contraire. La structure retenue pour ce type d'explication est la suivante:

- Classification par la méthode Electre I. Si le secteur est surclassé, les secteurs par lesquels il est surclassé sont affichés. Si le secteur n'est pas surclassé ce sont les secteurs surclassés par le secteur qui sont affichés.
- Énumération des critères qui ont une très bonne ou une très mauvaise valeur par ordre d'importance selon que le secteur soit surclassé ou pas.
- Énumération des critères qui ont une bonne ou une mauvaise valeur par ordre d'importance selon que le secteur soit surclassé ou pas.

L'explication obtenue pour le même site¹⁰ 1222.12 est montré dans la Figure 2-3. La notion de surclassement utilisée par les méthodes Electre permet de donner des explications pour comparer deux actions. Ceci est particulièrement intéressant pour les problèmes de choix de site, pour lesquels les effets bien connus de NIMBY (Not In My BackYard), PIITBY (Put It In Their BackYard) et NIMEY (Not In My Election Year) sont très forts. Il est donc souvent nécessaire d'expliquer aux représentants d'une commune les raisons qui ont conduit à sa sélection et non pas à celle d'autres communes. L'explication comparative de la Figure 2-4 est encore assez peu développée et montre deux informations relativement peu intéressantes et peu compréhensibles pour un décideur non initié aux méthodes Electre.

1222.12 a surclassé 1242.02 (indice de concordance = 0.950, indice de discordance = 0.167)
 1242.02 n'a pas surclassé 1222.12 (indice de concordance = 0.570, indice de discordance = 0.333)

Figure 2-4. Explication comparative entre deux sites de stockage de déchets potentiels en Suisse, fondée sur la relation de surclassement de la méthode Electre et produite par StocDech.

2.2.4 Stratégies d'explication

Le contenu d'une explication peut varier fortement en fonction de la stratégie adoptée. La stratégie utilisée pour produire les explications sur la somme pondérée, par exemple, est la suivante:

¹ Il s'agit en réalité de « secteurs » plutôt que de « sites ». Le mot « site » est utilisé uniquement pour des raisons de simplicité et clarté. Dans le cadre de ce document les deux mots sont à considérer de manière équivalente.

Affichage de la valeur de préférence calculée par la somme pondérée dans l'ordre des actions potentielles.

Enumération des critères qui ont une très bonne valeur par ordre d'importance.

Enumération des critères qui ont une bonne valeur par ordre d'importance.

Enumération des critères qui ont une mauvaise valeur par ordre d'importance.

Enumération des critères qui ont une très mauvaise valeur par ordre d'importance.

Affichage des valeurs des deux critères les plus importants.

Cette stratégie pourrait s'appeler « explication par valeur d'utilité » versus « explication par importance », puisque la valeur d'utilité détermine l'ordre de présentation des critères. Dans le cas de la somme pondérée, il est normal d'utiliser une telle stratégie puisque ce sont les valeurs d'utilité qui changent en fonction de l'action, les importances restant toujours les mêmes.

Dans le cas de fonctions de préférence contextuelle, les importances des critères peuvent changer en fonction de l'action. Dans ce cas, il semble moins évident de déterminer quelle stratégie d'explication utiliser. Cela est encore moins évident s'il existe des concepts intermédiaires. Le choix de la stratégie d'explication à adopter est traité plus en détail au Chapitre 4.

2.3 Conclusion

Une explication doit permettre de convaincre le décideur que la recommandation du SAD est bonne. Dans StocDech, les explications permettent également au décideur de déterminer la méthode d'aide à la décision à laquelle il fait le plus confiance.

Les explications fournies par StocDech selon les deux méthodes d'analyse multicritère (somme pondérée et Electre I) prouvent qu'il est possible d'obtenir des explications utiles et compréhensibles à partir de ces méthodes. Les explications présentent une bonne fidélité. La compréhensibilité et la complétude des explications sont à peu près du même niveau que pour le système expert utilisé. La quantité de travail supplémentaire est petite et l'efficacité est bonne, même si certaines explications à partir de la méthode Electre I sont longues à obtenir.

Nous pouvons donc constater que le niveau d'explication offert par le système expert à base de règles utilisé pour cette application ne constitue pas vraiment un avantage explicatif par rapport aux méthodes d'analyse multicritère. Ceci est également vrai pour le SIAD utilisant les réseaux de neurones, à condition de trouver des équivalences aux concepts d'importance et d'utilité.

3. Explication des résultats obtenus par réseaux neuronaux

Les réseaux de neurones sont souvent vus comme des « boîtes noires », qui produisent des résultats qu'il est impossible d'expliquer. Il est vrai qu'il est très difficile de comprendre le résultat d'un réseau neuronal en étudiant les valeurs de ses poids ou ses valeurs d'activation et de sortie. Plus le réseau neuronal devient grand, plus cette interprétation devient difficile.

C'est pour cette raison que l'approche proposée ici consiste à étudier le comportement du réseau neuronal « de l'extérieur » pour en extraire certaines caractéristiques intéressantes. Une telle approche correspond aux tendances actuelles dans le domaine des explications fournies par les systèmes experts, où les modules de raisonnement et d'explication sont de plus en plus séparés.

Il y a néanmoins une lacune grave dans les explications des systèmes experts actuels. Même s'il existe un grand nombre d'approches variées pour traiter l'incertitude au niveau du module de raisonnement, l'incertitude est souvent mal traitée dans le module explication. Les seules approches proposées consistent à afficher les coefficients de certitude ou probabilités des réponses. La seule explication sur ces informations consiste à montrer le calcul ayant permis de les obtenir, ce qui est généralement trop compliqué ou sans intérêt pour le décideur. Il est encore plus difficile de connaître les raisons du choix des coefficients de certitude que celles du choix des règles.

Dans un contexte d'aide à la décision multicritère, l'incertitude constitue généralement un élément essentiel du raisonnement et, donc, de l'explication. Le vocabulaire utilisé concerne des importances et des utilités, plutôt que des coefficients de certitude ou des probabilités. Les concepts d'importance et d'utilité contextuelles introduits dans ce chapitre permettent

d'obtenir des explications du même type que celles de StocDech, tout en permettant l'utilisation de fonctions de préférence contextuelle. En principe, ces concepts permettent de produire des explications à partir de n'importe quel type de raisonnement. Ils ne sont donc pas limités au réseau neuronal INKA, mais ils sont applicables à d'autres systèmes (autres réseaux neuronaux, systèmes utilisant la logique floue, ...).

L'explication par similarité constitue une autre possibilité d'explication. Le principe de ce type d'explication rejoint un peu celui du raisonnement à base de cas [Schank et al., 1994]. Il s'agit d'utiliser une explication existante pour une action similaire.

3.1 Importance contextuelle

L'importance d'un critère de sélection peut être considérée comme constante si elle ne change pas en fonction des valeurs d'autres critères. Dans ce cas, il est assez simple de construire des explications, comme nous avons vu dans le Chapitre 1 pour les résultats issus des méthodes de la somme pondérée et Electre I. Dans le cas de fonctions de préférence contextuelle du type défini au Chapitre 1.3 la situation est différente, car il n'y a pas de correspondance directe entre les paramètres définissant la fonction de préférence et l'importance d'un critère. Les poids d'un réseau neuronal n'ont a priori rien à voir avec les importances des critères et il est très difficile à analyser et à identifier le lien.

Il est néanmoins possible de redéfinir la notion d'importance pour l'appliquer aux fonctions de préférence contextuelle. Une façon de définir l'importance d'un critère consiste à considérer la variation observée dans la valeur de préférence d'une action en faisant changer les valeurs du critère. Si la variation de la valeur de préférence entre les deux valeurs extrêmes est plus grande pour un critère que pour un autre, le premier est plus important. Cette définition se traduit par l'expression suivante [Främling, 1992]:

$$ID_j(a_i) = \frac{dynmax_j(a_i) - dynmin_j(a_i)}{absmax - absmin} \quad (15)$$

où $ID_j(a_i)$ est l'importance contextuelle du critère j pour l'action i , $dynmax_j(a_i)$ est la valeur de préférence la plus grande et $dynmin_j(a_i)$ la plus petite trouvée en faisant varier la valeur du même critère j . Les valeurs des autres critères sont déterminées par l'action a_i , qui définit le contexte étudié. Les variables $absmax$ et $absmin$ sont la valeur maximale et la valeur

minimale pour la valeur de préférence. La valeur $absmax - absmin$ est ainsi l'intervalle de variation maximale de la valeur de préférence, estimable en faisant varier les valeurs de tous les critères simultanément.

Les valeurs de $dynmax_j(a_i)$, $dynmin_j(a_i)$, $absmax$ et $absmin$ sont faciles à identifier dans l'analyse graphique décrite au Chapitre 11 et donnée dans la Figure 3-1. Les valeurs $absmax$ et $absmin$ définissent l'échelle sur l'axe des valeurs de préférence et les valeurs $dynmax_j(a_i)$ et $dynmin_j(a_i)$ correspondent aux valeurs extrêmes de la fonction de préférence visualisée.

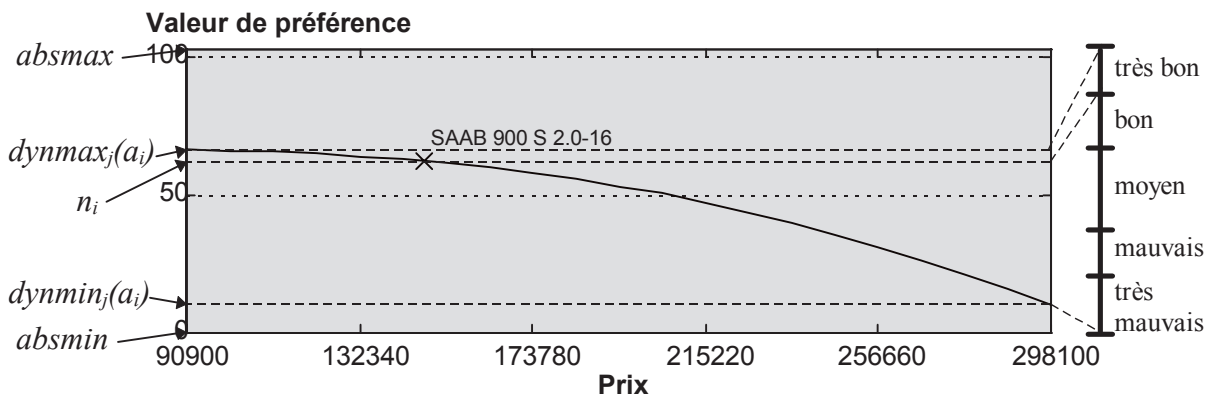


Figure 3-1. Illustration du calcul de l'importance et de la valeur d'utilité contextuelle. Dans ce cas $a_i = \text{« SAAB 900 S 2.0-16 »}$ et le critère $j = \text{« Prix »}$.

Cette définition est également valable pour des importances constantes. Dans le cas d'une méthode par somme pondérée, par exemple, la somme des poids est habituellement égale à la valeur de préférence maximale (celle de l'action idéale). La valeur de $absmin$ est donc égale à zéro, la valeur de $absmax$ égale à la somme des poids et $(dynmax_j - dynmin_j)$ est égal au poids affecté à chaque critère j , p_j . La somme des $(dynmax_j - dynmin_j)$ de tous les critères j est donc égal à $(absmax - absmin)$.

Ceci n'est pas le cas pour les importances contextuelles. Dans le cas extrême, $(dynmax_j - dynmin_j)$ peut être égal à $(absmax - absmin)$ même pour chaque critère. Nous nous retrouvons alors dans le cas des bases de règles et d'autres systèmes de raisonnement symbolique. Dans ce type de systèmes, ID_j est souvent soit un, soit zéro en fonction de la prise en compte ou non du critère j dans une règle utilisée au cours du raisonnement. Si la valeur du critère j change suffisamment pour être incluse ou exclue du raisonnement, son importance peut d'un coup varier de zéro à un ou inversement. Un tel comportement n'est bien sûr pas souhaitable, car la non-continuité numérique du raisonnement symbolique classique devient davantage marquée.

Pour fournir des explications en langue naturelle, il faut traduire la valeur numérique de ID en une valeur symbolique, comme c'était le cas dans StocDech (voir le Tableau 2-1). La définition des valeurs symboliques consiste à effectuer une partition de l'intervalle $[absmin, absmax]$ en des sous-intervalles correspondant aux valeurs « important », « peu important », etc.

Un exemple issu du choix de site qui illustre l'utilisation des importances contextuelles est la règle suivante: *Si la valeur du critère proximité autoroute est très bonne alors l'importance du critère proximité d'un chemin de fer est très faible*. Cet exemple illustre l'importance contextuelle, car l'importance du critère « proximité d'un chemin de fer » serait plus grande sans la présence d'autoroute. L'utilité de la valeur du critère proximité autoroute est une utilité contextuelle, définie suivant les principes de l'utilité contextuelle.

3.2 Utilité contextuelle

Dans le cas de fonctions de préférence contextuelle, les utilités des valeurs de critères de sélection peuvent changer en fonction du contexte, comme l'importance d'un critère. Ceci signifie qu'une valeur d'un critère de sélection n'a pas toujours la même utilité. Pour une voiture a , par exemple, la couleur jaune peut être très mauvaise, alors qu'elle est parfaite pour la voiture b . La définition de l'utilité contextuelle d'un critère se traduit par l'expression suivante [Främling, 1992]:

$$UD_j(a_i) = \frac{n_i - dynmin_j(a_i)}{dynmax_j(a_i) - dynmin_j(a_i)} \quad (16)$$

où $UD_j(a_i)$ est la valeur d'utilité contextuelle du critère j pour l'action i , n_i est la valeur de préférence de l'action i , $dynmax_j(a_i)$ est la valeur de préférence la plus grande $dynmin_j(a_i)$ la plus petite obtenue en faisant varier la valeur du critère j . Comme dans le calcul de l'importance contextuelle, les valeurs des autres critères sont déterminées par l'action i , qui définit le contexte étudié. Cette définition est également valable pour des méthodes additives, comme la méthode de somme pondérée. Le lien est fort entre la formule (16) et la formule (14) de normalisation de l'utilité dans StocDech.

La notion de valeur d'utilité contextuelle est illustrée dans la Figure 3-1. La valeur d'utilité est calculée par rapport au même intervalle $[dynmin_j(a_i), dynmax_j(a_i)]$ que dans l'expression

(15). La définition des valeurs symboliques pour la valeur d'utilité consiste à effectuer une partition de cet intervalle en sous-intervalles correspondant à des concepts de type « bon », « moyen », « mauvais ». Dans la Figure 3-1 l'action i a une valeur très bonne pour le critère en question.

Les concepts et les intervalles de valeurs utilisés pour la valeur d'utilité peuvent être différents pour chaque critère. Ceux qui sont présentés dans la Figure 3-1 sont bien adaptés pour les critères « vitesse maximale » ou « géologie globale », que pour des critères comme « prix » ou « consommation d'essence ». Une échelle comme « très cher », « cher », « moyennement cher », « peu cher » et « très cher » serait mieux adapté pour le critère « prix ». Comme nous l'avons déjà constaté au Chapitre 1.3, il est important d'utiliser un vocabulaire approprié pour pouvoir fournir des bonnes explications.

3.3 Concepts intermédiaires dans les réseaux de neurones

Les définitions d'importance et d'utilité contextuelles permettent d'accéder au même type d'explications que pour les méthodes d'analyse multicritère de StocDech. Cependant, cela ne suffit pas toujours pour atteindre le même niveau d'explications que celui des systèmes experts, qui permettent des explications de plusieurs niveaux. Ces niveaux sont définis par les concepts intermédiaires (ou hypothèses) qui permettent de structurer le domaine. Ils sont utilisés pour la mise en place d'un graphe de règles, d'un réseau sémantique ou d'autres méthodes de raisonnement (« frames », raisonnement à base de cas, ...).

Au niveau général, un concept intermédiaire est utilisé pour regrouper des critères de sélection ou d'autres concepts intermédiaires, soit pour faciliter la conception de l'outil de décision, soit pour mieux comprendre les décisions prises. Ces concepts intermédiaires donnent ainsi la possibilité d'étudier le domaine selon différents points de vue, ce qui permet également d'expliquer les résultats au niveau de détail souhaité.

Dans les explications des méthodes d'analyse multicritère de StocDech, il n'y a pas de concepts intermédiaires, bien qu'il soit possible d'intégrer cette notion. Dans le domaine de choix de voiture, par exemple, il suffit de dire que le critère « performances » se décompose en « puissance maximale », « poids », « vitesse maximale » et « 1000m départ arrêté ». Ces quatre critères sont appelés des *sous-critères* du concept intermédiaire « performances ». Dans le cas d'une méthode comme la somme pondérée, l'importance de ce concept intermédiaire

est directement la somme des importances de chaque critère individuel. Pour trouver la valeur d'utilité correspondante, il suffit de calculer la somme pondérée pour les quatre sous-critères et de la diviser par la somme de leurs poids.

3.3.1 Importances et utilités contextuelles des concepts intermédiaires

Dans le cas d'importances et d'utilités contextuelles la définition précédente change. L'importance d'un concept intermédiaire peut être très différente de la somme des importances des sous-critères. Pour trouver l'importance contextuelle d'un concept intermédiaire, il est donc nécessaire de refaire le calcul (15), tout en faisant varier les valeurs de tous les sous-critères (et éventuellement de leurs sous-critères jusqu'au niveau de critères de sélection de base). La définition de l'importance contextuelle d'un concept intermédiaire est donc exactement la même que pour les critères de sélection individuels. Cette définition reste toujours applicable pour des méthodes comme la somme pondérée.

Une petite remarque concernant le cas extrême peut être intéressante pour mieux comprendre l'importance des concepts intermédiaires. Il est même possible de calculer l'importance du concept intermédiaire « valeur de préférence » pour la valeur de préférence, qui devrait être 100%. Le calcul de cette importance revient à faire varier les valeurs de tous les critères de sélection. Les valeurs $dynmax(a_i)$ et $dynmin(a_i)$ devrait, dans ce cas, être égales aux valeurs $absmax$ et $absmin$.

Le principe de calcul de l'utilité contextuelle est le même que pour l'importance contextuelle. Puisque la valeur n_i est connue et que les valeurs $dynmax(a_i)$ et $dynmin(a_i)$ du concept intermédiaire ont été calculées pour trouver l'importance contextuelle, il suffit d'appliquer le calcul (16) pour trouver l'utilité de la valeur du concept intermédiaire. Cette définition reste, elle aussi, applicable pour des méthodes comme la somme pondérée.

L'explication concernant la valeur de préférence et le classement d'une action peut donc se décomposer en concepts intermédiaires. Ceci introduit un nouveau besoin, le besoin de justifier leurs valeurs, ce qui revient en pratique à justifier leurs valeurs d'utilité. L'explication de la valeur d'un concept intermédiaire peut se construire exactement de la même façon qu'une explication sur la valeur de préférence. Pour le faire, il faut que les importances des sous-critères soient indiquées par rapport au concept intermédiaire.

Chapitre 3. Explication des résultats obtenus par réseaux neuronaux

Il s'agit donc d'une *importance relative*, à la différence de l'*importance globale* qui est définie par rapport à la valeur de préférence. Pour cette raison, le calcul de l'importance d'un sous-critère pour un concept intermédiaire consiste à utiliser les valeurs $dynmax(a_i)$ et $dynmin(a_i)$ du concept intermédiaire comme des valeurs *absmax* et *absmin* dans la formule (15). Les importances des sous-critères sont par la suite calculées par rapport à cet intervalle.

A partir de ces définitions, il est possible de déduire certaines limites logiques concernant les importances globales des concepts intermédiaires et de leurs sous-critères:

L'importance globale d'un concept intermédiaire est toujours supérieure à l'importance globale individuelle de chacun de ses sous-concepts. Faire varier la valeur d'un sous-concept peut seulement produire des combinaisons de valeurs qui sont un sous-ensemble des combinaisons de valeur de tous les sous-concepts. Ceci est également vrai pour des sous-concepts qui sont des concepts intermédiaires eux-mêmes. Il est donc impossible qu'un sous-critère ait une importance relative supérieure à cent pour-cent.

L'importance globale d'un concept intermédiaire peut être supérieure ou inférieure à la somme d'importances de tous ses sous-concepts. La relation d'infériorité a déjà été discutée, mais celle de supériorité peut sembler plus illogique. Il est cependant possible qu'une combinaison de valeurs des sous-critères puisse donner des valeurs de préférence différentes de celles trouvées en faisant varier la valeur d'un seul sous-concept. Ce cas demeure assez exceptionnel.

Si l'une ou l'autre de ces conditions n'est pas respectée, il est d'abord nécessaire de vérifier la fonction de préférence apprise, puis les valeurs extrêmes $dynmax(a_i)$ et $dynmin(a_i)$. Ces valeurs extrêmes sont censées être représentées par les centroïdes des neurones, principe de base de la méthode de recherche. Cette méthode permet néanmoins d'éviter les problèmes d'explosion combinatoire provoqués par le nombre de combinaisons de valeurs à tester.

3.3.2 Problème d'explosion combinatoire

L'explosion combinatoire provoquée par le nombre de combinaisons de valeurs à tester pose des problèmes pour le calcul des deux valeurs $dynmax_i(a_i)$ et $dynmin_i(a_i)$ des concepts intermédiaires. Il est particulièrement aigu pour les critères qualitatifs où, déjà avec quatre critères ayant cinq valeurs possibles, il y a $4^5 = 1024$ combinaisons possibles. Un réseau de

neurones permet d'effectuer rapidement le nombre de calculs requis. La vitesse de calcul dépend de la taille du réseau de neurones, du simulateur et de l'ordinateur utilisé, mais l'augmentation exponentielle du nombre de calculs à effectuer rend vite le temps de calcul trop long.

Dans le cas de critères quantitatifs ayant des valeurs continues, il est possible d'utiliser des méthodes de gradient ou autre pour rechercher les valeurs maximale et minimale de la fonction de préférence. Même dans ce cas, les temps de calcul risquent d'être longs et il est difficile d'être sûr de trouver les minima et maxima globaux.

INKA permet d'effectuer une recherche beaucoup plus rapide et fiable que la plupart des autres réseaux de neurones. L'algorithme d'apprentissage de INKA crée des nouveaux neurones pour les exemples ayant des valeurs de préférence maximales ou minimales. Il suffit donc de remplacer les valeurs des critères étudiés par des valeurs des centroïdes des neurones cachés pour avoir une bonne approximation de $dynmax(a_i)$ et de $dynmin(a_i)$. Pour que cette méthode reste fiable, il faut que la valeur du paramètre d_i^2 soit correcte pour éviter des oscillations excessives dans la fonction apprise.

Même avec un réseau neuronal ayant des centaines de neurones, le calcul reste relativement rapide. Cette fonctionnalité de INKA constitue un avantage par rapport à la plupart des autres réseaux neuronaux, pour lesquels les temps de calcul deviennent vite trop longs. Un autre type d'explication qui ne pose pas de problèmes au niveau du temps de calcul est l'explication par similarité. Par contre, ce type d'explication présente l'inconvénient de ne pas être applicable dans toutes les situations.

3.4 Explications par similarité

L'explication par similarité est une autre façon de fournir des explications. Il s'agit d'utiliser l'explication déjà connue sur une action potentielle qui « ressemble » à l'action à expliquer. Cette action déjà connue peut être un des exemples d'apprentissage fournis par le décideur. Pour être capable de déterminer la similarité, il est nécessaire d'avoir une mesure de distance entre deux actions. Ce type de distance est déjà défini dans la plupart de méthodes de classification numérique. Dans les systèmes de classification symbolique, il est plus difficile d'identifier de telles mesures de distance. Le raisonnement à base de cas [Schank et al., 1994] offre certaines fonctionnalités de ce type.

Une méthode qui permet une mesure simple de la similarité entre actions sont les cartes auto-organisatrices de Kohonen présentées au Chapitre 6.1. Si deux actions appartiennent à la même classe, elles sont par définition assez similaires. Même si elles appartiennent à deux classes voisines, elles peuvent être assez similaires pour permettre d'appliquer l'explication de l'une à l'autre.

Malgré les notions de classe et de voisinage, il reste utile d'indiquer la proximité entre les actions. Cette distance sert à indiquer à quel point l'explication correspond à l'action expliquée. Si la distance est de zéro, l'explication devrait normalement être applicable à cent pour-cent. Plus la distance augmente, plus la similarité diminue. Le calcul approprié de cette distance reste encore à déterminer (distance euclidienne ou autre), mais la notion de classe doit normalement suffire pour déterminer l'applicabilité de l'explication d'une action sur une autre.

L'intérêt des explications par similarité est qu'elles sont souvent mieux acceptées par le décideur que les explications fondées sur l'importance et l'utilité des valeurs. Ceci est particulièrement vrai si les explications concernent des exemples d'apprentissage introduits par le décideur lui-même. Malheureusement, le faible nombre d'exemples d'apprentissage fait qu'il n'est pas toujours possible d'identifier une action de référence dont l'explication serait utilisable. Les explications par similarité restent donc des explications complémentaires..

L'intérêt d'utiliser une explication par similarité dépend du degré auquel l'explication la plus « proche » s'applique au cas traité. La détermination de cette « proximité » revient à estimer un niveau de crédibilité pour l'explication. L'estimation d'un tel niveau de crédibilité est importante pour tout type d'explication, les explications par importance et utilité comprises. Ceci est le sujet du chapitre suivant.

3.5 Niveau de crédibilité dans les explications

Les coefficients de certitude présentés au Chapitre 11.2 peuvent être utilisés pour calculer un niveau de crédibilité pour les explications de la même façon que le coefficient de certitude moyen de l'ensemble d'actions potentielles. Le niveau de crédibilité dans une explication est calculé comme le coefficient de certitude moyen pour l'ensemble d'actions utilisées dans la recherche des valeurs de $dynmax_j(a_i)$ et $dynmin_j(a_i)$. D'autres opérations, différentes de la

moyenne arithmétique, sont envisageables (valeur maximale ou minimale ou d'autres calculs pour traiter les coefficients de certitude).

Le niveau de crédibilité reste faible si le nombre d'exemples d'apprentissage est faible, comme c'est le cas dans les applications étudiées. L'introduction de pseudo-exemples est un excellent moyen d'augmenter ce niveau de crédibilité, tout en augmentant le niveau et la précision des explications. Puisque les pseudo-exemples sont normalement introduits pour des valeurs de critères extrêmes, ils fournissent beaucoup d'information concernant notamment les valeurs de préférence minimales et maximales.

Le niveau de crédibilité de l'explication d'une action potentielle est normalement inférieur au coefficient de certitude de sa valeur de préférence. Ceci est tout à fait logique, car expliquer une décision demande toujours beaucoup plus d'information que de la prendre. Le niveau de crédibilité des explications aide à savoir si le nombre d'exemples d'apprentissage est suffisant pour fournir des explications.

4. Construction d'une explication

Pour fournir une bonne explication il est rarement suffisant d'en produire une seule. Le même type d'explication n'est pas toujours bon. Les facteurs qui déterminent le type d'explication à utiliser sont l'action expliquée et l'utilisateur. Chaque utilisateur n'a pas le même niveau de connaissances du domaine et il est nécessaire d'ajuster le niveau de l'explication pour correspondre à ce niveau de connaissances.

Comme constaté dans [Bouri et al., 1990b], la prise en compte de l'incertitude offre de nouvelles possibilités d'explications. Il devient notamment possible d'augmenter le nombre de façons de construire les explications. Les principes d'explication employés doivent être valables pour des situations de certitude aussi bien que d'incertitude. Les explications ne doivent pas être trop liées à la méthode de raisonnement car, si c'est le cas, un changement de méthode de raisonnement nécessite un changement de module d'explication.

Les stratégies paramétrables présentées dans ce chapitre permettent le traitement de l'incertitude grâce aux notions d'importance et d'utilité contextuelles et aux concepts

intermédiaires présentés au chapitre précédent. La nature générale de ces notions les rend applicables quelle que soit la méthode de raisonnement. Il faut seulement que les temps de calcul restent raisonnables et que toutes les combinaisons de valeurs des critères de sélection soient suffisamment bien couvertes.

L'application utilisée comme exemple pour illustrer les stratégies possibles est le cas suisse de choix des sites de stockage de déchets industriels ultimes [Rey et al., 1988]. Cette application est la seule pour laquelle une base de règles ait été développée. Il est donc possible de comparer les explications obtenues à celles de la Figure 2-1.

L'ensemble d'apprentissage utilisé comporte 12 exemples choisis à l'aide d'une carte auto-organisatrice de Kohonen, dont 2 exemples d'idéal et anti-idéal. Vingt-six pseudo-exemples ont été ajoutés à cette base d'exemple initiale, qui sont tous identiques au secteur « 1222.12 », à un critère près. La valeur d'un critère a été remplacée par la valeur minimale ou maximale du critère dans chaque pseudo-exemple. Les valeurs de préférence sont calculées par la méthode de somme pondérée.

L'apprentissage de la fonction de préférence utilisée est assez simple. Puisque tous les pseudo-exemples utilisent le secteur 1222.12 comme base, la fonction est définie avec une précision très élevée pour ce secteur. L'objectif recherché ici n'est pas de juger les capacités d'apprentissage du réseau neuronal, mais uniquement de comparer les explications obtenues en utilisant l'importance et l'utilité contextuelles avec celles obtenues en utilisant d'autres méthodes.

4.1 Paramètres déterminant le contenu des explications

Un module d'explication expérimental qui a été développé¹¹ permet la définition de différentes stratégies d'explication par l'intermédiaire de quelques paramètres. Les paramètres actuellement utilisés sont les suivants:

¹¹ Ce module a été développé avec l'outil de calcul matriciel « Matlab », le même outil qui a été utilisé pour la programmation du réseau neuronal INKA et pour produire la plupart des résultats de ce document (graphes, classements, valeurs de préférence, ...).

- Valeur booléenne qui indique si les concepts intermédiaires sont utilisés ou pas.
- Profondeur maximale de l'arbre de concepts intermédiaires utilisé dans les explications.
- Type de parcours de l'arbre de concepts intermédiaires, en profondeur ou en largeur d'abord.
- Valeur booléenne qui indique si les importances des critères doivent être affichées.
- Valeur booléenne qui indique si les valeurs *ID* doivent être affichées.
- Classes d'importance à afficher et leur ordre.
- Valeur booléenne qui indique si les importances des critères doivent être groupées sous un titre indiquant l'importance.
- Valeur booléenne qui indique si les utilités des valeurs des critères doivent être affichées.
- Valeur booléenne qui indique si les valeurs *UD* doivent être affichées.
- Classes d'utilité à afficher et leur ordre.
- Valeur booléenne qui indique si les utilités des critères doivent être groupées sous un titre indiquant l'utilité.

Tous ces paramètres ne sont qu'un sous-ensemble des paramètres imaginables pour préciser et modifier le contenu des explications. Il y a également d'autres paramètres plus implicites, comme le vocabulaire du domaine qui exprime s'il s'agit du choix d'une voiture ou d'un site de stockage. Les structures de concepts intermédiaires utilisées sont également déterminantes pour le contenu des explications. Comme évoqué précédemment, les définitions des limites et du vocabulaire pour des classes d'importance et d'utilité peuvent radicalement modifier les explications obtenues.

Les stratégies qu'il est possible d'obtenir en modifiant ces paramètres peuvent être définies sous des noms comme « concepts intermédiaires d'abord », « importance d'abord » ou « utilité d'abord ». Une explication de type « concepts intermédiaires d'abord » correspond aux explications obtenues avec les bases de règles. Certaines combinaisons de paramètres sont interdites afin d'éviter des explications illogiques ou contenant trop d'information.

Il peut s'avérer inutile d'utiliser des critères sans importance, ou avec très peu d'importance, dans les explications. Si l'importance d'un critère est vraiment faible, ce critère a peu d'influence sur le résultat expliqué.

4.2 Explications de type « base de règles »

Une explication obtenue pour le secteur Suisse « 1222.12 » est indiquée dans la Figure 4-1. Elle est similaire à l'explication fournie par la base de règles de StocDech (Figure 2-1). Son mécanisme de construction est définie de la manière suivante:

- Recours aux concepts intermédiaires.
- Profondeur maximale de concepts intermédiaires.
- Recherche par profondeur d'abord.
- Aucun affichage d'importances.
- Afficher les utilités de façon groupée.

Le niveau d'explication obtenu correspond bien à celui obtenu avec la base de règles de StocDech, mais il reste quelques petites différences. Premièrement, la fonction de préférence est celle de la somme pondérée et pas celle de la base de règles, ce qui explique par exemple le fait que l'accès soit moyen ici, tandis qu'il est bon dans l'explication donnée par la base de règles.

Il est important de noter que les utilités affichées dans l'explication par la base de règles sont directement les valeurs qualitatives des critères. Seules les utilités calculées sont affichées dans l'explication de INKA, ce qui n'empêche pas, bien sûr, d'afficher les valeurs aussi. Dans un SAD à base de règles classique il est difficile d'utiliser des utilités contextuelles. L'utilité contextuelle risque d'être incohérente avec la valeur qualitative des critères, constituant les prémisses des règles.

Cependant, l'avantage des explications obtenues à partir d'une base de règles par rapport aux explications obtenues en utilisant les importances et les utilités contextuelles est le suivant: dans une base de règles, l'arbre d'inférence est construit en fonction des besoins d'inférence et, en conséquence, seuls les concepts intermédiaires effectivement utilisés sont expliqués. Par contre, dans une explication obtenue par importances et utilités contextuelles, l'arbre de

concepts intermédiaires est fixé d'avance. En conséquence, les concepts utilisés dans les explications sont les mêmes pour toutes les actions. Comme évoqué au chapitre précédent, il est possible de modifier ce comportement en imposant une importance minimale pour les concepts inclus dans les explications. En respectant ces contraintes, il est donc possible d'obtenir un arbre de concepts intermédiaires identique à l'arbre d'inférences de la base de règles.

Le Secteur 1222.12 est bon, valeur de préférence 63.7 sur 100, parce que:

- Nuisances vis à vis la population a une bonne valeur.
- Valeurs esthétiques a une bonne valeur.
- Accès a une moyenne valeur.
- Géologie globale a une bonne valeur.
- Hydrographie a une très bonne valeur.
- Aptitude agricole a une très mauvaise valeur.

Nuisances vis à vis la population a une bonne valeur. La valeur de ce critère dépend des valeurs des critères:

- Longueur du parcours en agglomération a une très bonne valeur.
- Nombre de localités traversées a une bonne valeur.

Valeurs esthétiques a une bonne valeur. La valeur de ce critère dépend des valeurs des critères:

- Exposition du secteur a une moyenne valeur.
- Valeur écologique a une très bonne valeur.

Accès a une moyenne valeur. La valeur de ce critère dépend des valeurs des critères:

- Proximité des fournisseurs de DS a une très bonne valeur.
- Proximité de chemin de fer a une très mauvaise valeur.
- Longueur à construire ou à élargir a une très bonne valeur.
- Longueur parcours RN-Secteur a une très bonne valeur.

Géologie globale a une bonne valeur. La valeur de ce critère dépend des valeurs des critères:

- Proximité d'un utilisateur de marnes a une très bonne valeur.
- Géologie globale a une bonne valeur.

Hydrographie a une très bonne valeur. La valeur de ce critère dépend des valeurs des critères:

- Nombre de sources privées a une très bonne valeur.
- Longueur de ruisseaux a une très bonne valeur.

Figure 4-1. Exemple d'une explication de type base de règles pour un site suisse, produite automatiquement (d'où les fautes de langue) en utilisant les concepts intermédiaires et les utilités contextuelles. Les mots en italique indiquent l'importance et les mots soulignés indiquent l'utilité.

4.3 Explications de type « analyse multicritère »

Une explication obtenue pour le secteur Suisse « 1222.12 » est indiquée dans la Figure 4-2. Elle est similaire à l'explication fournie par StocDech pour la somme pondérée (Figure 2-2). Son mécanisme de construction est définie de la manière suivante:

- Ne pas utiliser les concepts intermédiaires.
- Afficher les importances et les pourcentages d'importance.
- Grouper les critères par valeur d'utilités.
- Limiter les utilités affichées aux deux classes d'utilité les meilleures et aux deux classes d'utilité les plus mauvaises.

Le Secteur 1222.12 est bon, valeur de préférence 63.7 sur 100, parce que:

Les critères ayant une très bonne valeur sont:

- Longueur du parcours en agglomération, qui est *assez important* (0.10).
- Valeur écologique, qui est *peu important* (0.10).
- Longueur parcours RN-Secteur, qui est *peu important* (0.05).
- Longueur de ruisseaux, qui est *très peu important* (0.05).
- Proximité des fournisseurs de DS, qui est *très peu important* (0.05).
- Longueur à construire ou à élargir, qui est *très peu important* (0.04).
- Proximité d'un utilisateur de marnes, qui est *très peu important* (0.04).
- Nombre de sources privées, qui est *très peu important* (0.03).

Les critères ayant une bonne valeur sont:

- Nombre de localités traversées, qui est *assez important* (0.13).
- Géologie globale, qui est *assez important* (0.12).

Les critères ayant une très mauvaise valeur sont:

- Aptitude agricole, qui est *assez important* (0.15).
- Proximité de chemin de fer, qui est *très peu important* (0.05).

Figure 4-2. Exemple d'une explication de type somme pondérée pour un site suisse, produite automatiquement en utilisant les importances et les utilités contextuelles. Les mots en italique indiquent l'importance et les mots soulignés indiquent l'utilité.

Cette explication est pratiquement identique à celle de la Figure 2-2. Il y a néanmoins certaines différences. La première vient de l'utilisation de fonctions d'utilité continues pour les critères quantitatifs, au lieu des fonctions d'utilité par intervalles de valeurs utilisées dans StocDech. Il y a également des différences au niveau des importances et utilités, qui sont dues à des classes et des intervalles différents. Les classes et les intervalles d'importance et d'utilité utilisés sont indiqués dans le Tableau 4-1.

<i>Importance qualitative</i>	<i>Intervalle d'importance</i>	<i>Utilité qualitative</i>	<i>Intervalle d'utilité</i>
Très peu important	[0, 0.05[Très mauvaise	[0, 0.20[
Peu important	[0.05, 0.1[Mauvaise	[0.20, 0.40[
Assez important	[0.1, 0.2[Moyenne	[0.40, 0.60[
Important	[0.2, 0.35[Bonne	[0.60, 0.80[
Extrêmement important	[0.35, 0.50[Très bonne	[0.80, 1.0]
Presque déterminant	[0.50, 0.80[
Pratiquement déterminant	[0.80, 1.0]		

Tableau 4-1. Classes et intervalles d'importance et d'utilité utilisés dans les explications par importances et utilités contextuelles.

Ces intervalles d'importance ne sont pas très bien adaptés pour l'exemple d'explication donné, car il n'y a aucun critère dont l'importance qualitative soit supérieure à « assez important ». Ces intervalles sont assez adaptés pour les sous-concepts des concepts intermédiaires, car leur nombre est plus petit que le nombre de critères de sélection. Chacun d'entre eux a donc une importance relative plus grande. Pour remédier à ce problème, il faudrait modifier les classes et les intervalles en fonction du nombre de sous-concepts. Une autre possibilité pourrait être de les normaliser par rapport à l'importance du sous-concept le plus important.

4.4 Explications « mixtes »

Les explications de type « base de règles » et « somme pondérée » prouvent qu'il est possible d'obtenir des explications de même niveau qu'à l'aide des autres méthodes. Le grand nombre de paramètres et la faible connexion entre le module de raisonnement et le module d'explication permettent en plus de construire des explications plus flexibles et variées. Un exemple d'explication qui combine les aspects base de règles et analyse multicritère est donné dans la Figure 4-3.

Le Secteur 1222.12 est bon, valeur de préférence 63.7 sur 100, parce que:

- Nuisances vis à vis la population, qui est *assez important* (0.19) a une bonne valeur.
- Valeurs esthétiques, qui est *assez important* (0.18) a une bonne valeur.
- Accès, qui est *assez important* (0.14) a une moyenne valeur.
- Géologie globale, qui est *assez important* (0.12) a une bonne valeur.
- Aptitude agricole, qui est *assez important* (0.15) a une très mauvaise valeur.
- Hydrographie, qui est *peu important* (0.06) a une très bonne valeur.

Nuisances vis à vis la population, qui est un critère *assez important* (0.19) a une bonne valeur. La valeur de ce critère dépend des valeurs des critères:

- Nombre de localités traversées, qui est *presque déterminant* (0.67) a une bonne valeur.
- Longueur du parcours en agglomération, qui est *presque déterminant* (0.54) a une très bonne valeur.

Valeurs esthétiques, qui est un critère *assez important* (0.18) a une bonne valeur. La valeur de ce critère dépend des valeurs des critères:

- Exposition du secteur, qui est *presque déterminant* (0.62) a une moyenne valeur.
- Valeur écologique, qui est *presque déterminant* (0.54) a une très bonne valeur.

Accès, qui est un critère *assez important* (0.14) a une moyenne valeur. La valeur de ce critère dépend des valeurs des critères:

- Proximité de chemin de fer, qui est *extrêmement important* (0.37) a une très mauvaise valeur.
- Longueur parcours RN-Secteur, qui est *extrêmement important* (0.37) a une très bonne valeur.
- Proximité des fournisseurs de DS, qui est *important* (0.33) a une très bonne valeur.
- Longueur à construire ou à élargir, qui est *important* (0.29) a une très bonne valeur.

Géologie globale, qui est un critère *assez important* (0.12) a une bonne valeur. La valeur de ce critère dépend des valeurs des critères:

- Géologie globale, qui est *pratiquement déterminant* (0.97) a une bonne valeur.
- Proximité d'un utilisateur de marnes, qui est *important* (0.28) a une très bonne valeur.

Hydrographie, qui est un critère *peu important* (0.06) a une très bonne valeur. La valeur de ce critère dépend des valeurs des critères:

- Longueur de ruisseaux, qui est *presque déterminant* (0.73) a une très bonne valeur.
- Nombre de sources privées, qui est *presque déterminant* (0.55) a une très bonne valeur.

Figure 4-3. Exemple d'une explication mixte pour un site suisse, produite automatiquement en utilisant les concepts intermédiaires, les importances et les utilités contextuelles. Les mots en italique indiquent l'importance et les mots soulignés indiquent l'utilité.

La stratégie d'explication « mixte » utilisée est la suivante:

- Utiliser les concepts intermédiaires.
- Afficher les importances et les pourcentages d'importance en ordre décroissant.

Les importances des sous-concepts dépendent du concept intermédiaire. Dans l'exemple présenté, ces pourcentages sont un peu trop élevés, puisque leur somme devrait être égale à 1

dans le cas d'une fonction de préférence linéaire. Ceci indique qu'il y a encore des améliorations à apporter à l'algorithme de recherche de valeurs minimales et maximales de la fonction de préférence.

5. Explications pour l'application « choix de voiture »

Les explications sur les secteurs/sites suisses illustrent le type d'explications qu'il est possible d'obtenir pour des fonctions de préférence linéaires. Par contre, l'explication des résultats obtenus par une fonction de préférence contextuelle pose certains problèmes. Un premier problème concerne la logique même de l'explication, qui est moins évidente que dans le cas linéaire. Un deuxième problème concerne le grand nombre d'exemples nécessaire pour atteindre une précision suffisamment grande pour les besoins des explications.

5.1 Explications pour des fonctions de préférence contextuelle

Les explications sont étudiées pour une voiture spécifique, la SAAB 900 S 2.0-16 pour bien illustrer l'effet d'une fonction de préférence non-linéaire. Les pseudo-exemples du Chapitre 12.2 ont été remplacés par des pseudo-exemples construits à partir des données sur cette voiture. Dans chaque pseudo-exemple, la valeur d'un critère de sélection a été remplacée soit par la valeur minimale, soit par la valeur maximale possible du critère.

# Critère	1	2	3	4	5	6	7	8	9	10	11	12	13
ID (%)	58	11	6	5	6	6	4	10	10	39	8	5	10
Poids linéaire	12	11	6	5	6	6	4	10	10	10	8	5	10
UD	0.94	0.28	1.00	0.00	0.74	0.60	0.62	0.33	0.44	0.90	0.66	0.99	0.45
Utilité linéaire	0.72	0.29	1.00	0.00	0.76	0.61	0.63	0.33	0.45	0.61	0.67	1.00	0.45

Tableau 5-1. Tableau des valeurs ID, des poids initiaux, des valeurs UD et des valeurs d'utilité théoriques pour des fonctions d'utilité linéaires. Les colonnes des critères « prix » et « consommation moyenne » sont indiquées en gras.

Chapitre 5. Explications pour l'application « choix de voiture »

Un tel ensemble d'entraînement signifie que la fonction de préférence est définie avec une très grande précision pour la SAAB 900 S 2.0-16, mais beaucoup moins pour des voitures peu similaires. Les comparaisons avec les poids de départ des méthodes d'analyse multicritère deviennent ainsi plus faciles.

L'algorithme d'apprentissage créait 35 neurones pour 38 exemples disponibles. Ce nombre relativement élevé permet d'atteindre la grande précision souhaitée pour les estimations des importances et des utilités contextuelles. Une telle précision simplifie les comparaisons avec les poids et les utilités « linéaires ». Les valeurs calculées pour *ID* et *UD* sont indiquées dans le Tableau 5-1. Pour les critères qui sont linéaires, nous retrouvons les poids et les valeurs d'utilité d'origine. Par contre, dans le cas des critères « prix » et « consommation moyenne », *ID* et *UD* sont largement supérieures aux valeurs linéaires. Ceci est tout à fait logique.

La voiture SAAB 900 S 2.0-16 est moyenne, valeur de préférence 57.3 sur 103, parce que:

- Prix, qui est *presque déterminant* (0.58) a une très bonne valeur (147800).
- Consommation, qui est *extrêmement important* (0.39) a une très bonne valeur (10.2).
- Performances, qui est *important* (0.33) a une mauvaise valeur.
- Equipement, qui est *assez important* (0.10) a une moyenne valeur (50/83).
- Transmission, qui est *peu important* (0.06) a une très bonne valeur (manuelle).
- Motrices, qui est *peu important* (0.05) a une très mauvaise valeur (traction avant).
- Dimension, qui est *peu important* (0.06) a une bonne valeur (11.4).
- Coffre, qui est *peu important* (0.06) a une bonne valeur (451).
- Esthétique, qui est *peu important* (0.08) a une bonne valeur (belle).
- Image marque, qui est *très peu important* (0.05) a une très bonne valeur (très bon).

Performances, qui est un critère *important* (0.33) a une mauvaise valeur.

La valeur de ce critère dépend des valeurs des critères:

- Puissance, qui est *important* (0.33) a une mauvaise valeur (130).
- Accélération, qui est *important* (0.30) a une moyenne valeur (31.8).
- Vitesse, qui est *important* (0.30) a une mauvaise valeur (200).
- Poids, qui est *assez important* (0.12) a une bonne valeur (1285).

Figure 5-1. Explication produite automatiquement pour la SAAB 900 S 2.0-16. Les mots en italique indiquent l'importance et les mots soulignés indiquent l'utilité.

Le concept intermédiaire « performances » se décompose en sous-concepts « puissance maximale », « poids », « vitesse maximale » et « 1000m départ arrêté ». Les valeurs *ID* et *UD* obtenues pour « performances » sont 33% et 0.36. L'explication correspondante, qui a été obtenue avec des paramètres permettant un affichage maximal d'information, est indiquée dans la Figure 5-1.

Dans les explications obtenues pour le cas linéaire de choix de sites de stockage, une addition pondérée des bonnes et des mauvaises valeurs montrait que la conclusion finale était logique. Ce n'est pas le cas ici, car la valeur de préférence n'est que moyenne même si les valeurs des critères « prix » et « consommation moyenne » sont très bonnes, ces deux critères étant de loin les plus importants. L'explication donnée reste malgré tout correcte.

Il est normal que l'importance du prix correspond à la valeur de préférence de la voiture. Cependant, il est impossible d'améliorer le résultat en diminuant le prix, mais on peut atteindre une valeur de préférence de zéro en augmentant le prix. Pour expliquer cela, il faudrait encore trouver d'autres stratégies d'explication. L'explication suivante semble déjà plus logique¹² [Främling & Graillot, 1995]:

« La SAAB 900 S 2.0-16 est moyenne, 57.3 sur 103 (69^{ème} de 113 voitures). Les avantages de cette voiture sont qu'elle est peu chère (147800 Francs) et qu'elle a une faible consommation d'essence (10.2 l/100km), qui sont des critères très importants, 58 % et 39%. Par contre, sa valeur est mauvaise pour le critère performances, qui est un critère important, 33%. La puissance maximale et la vitesse maximale, qui ont une mauvaise valeur, sont respectivement des critères importants ou très importants pour établir les performances. »

Une telle explication ne satisfait pas pleinement le décideur. Elle donne encore envie de poser la question suivante: *Pourquoi la SAAB 900 S 2.0-16 est-elle moyenne même si elle n'est pas chère et elle a une faible consommation?* La réponse à une telle question consisterait à expliquer pourquoi la SAAB 900 S 2.0-16 n'est ni bonne, ni très bonne:

Même si le prix est très bon et la consommation petite, il faudrait absolument améliorer les performances. Si les performances étaient moyennes, par exemple, la voiture serait bonne. Avec des performances encore meilleures, elle serait très bonne. Si, en plus, le niveau d'équipement était bon, la SAAB 900 S 2.0-16 serait la meilleure voiture de toutes.

¹² Cette explication est tout à fait logique pour notre décideur « modèle », qui dispose de 200.000 francs et qui veut s'acheter une voiture à essence et à quatre portes. Cela ne correspond bien sûr pas à la situation de tout le monde, pour qui l'explication n'est peut-être pas aussi logique toute de suite.

Chapitre 5. Explications pour l'application « choix de voiture »

Une telle réponse présente un certain dynamisme de type « et si », qui se concentre sur les effets des changements hypothétiques dans les valeurs des critères de sélection. Le besoin de questions et de réponses de ce type n'a jamais été identifié avant parce que les importances et les utilités contextuelles n'ont jamais été utilisées. Ce type d'argumentation est absolument nécessaire pour pouvoir convaincre le décideur dans des situations de fonctions de préférence contextuelle.

5.2 Nombre d'exemples d'apprentissage nécessaire

Dans le chapitre précédent, les exemples d'apprentissage étaient choisis de façon à obtenir une fonction de préférence aussi précise que possible pour une voiture particulière. Malheureusement, le nombre d'exemples d'apprentissage nécessaire devient vite trop grand pour atteindre une telle précision pour toutes les voitures.

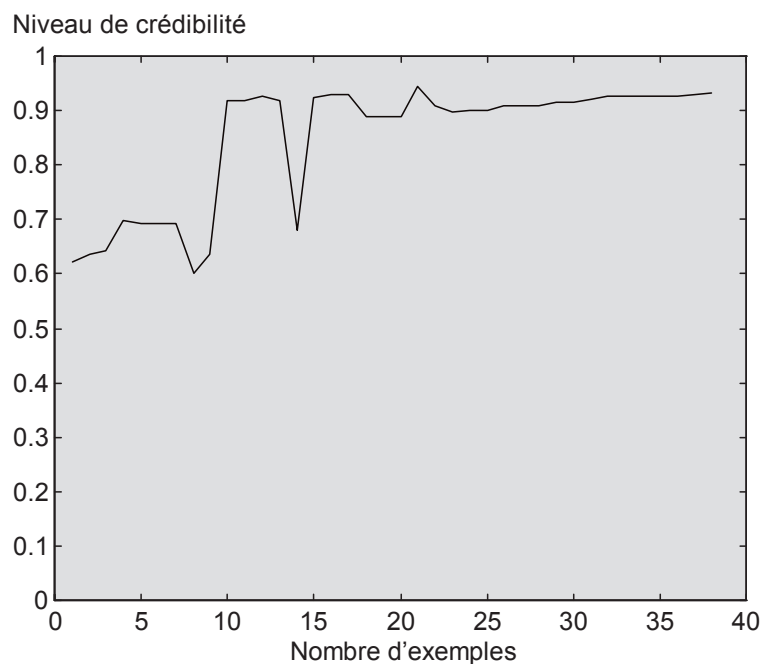


Figure 5-2. Évolution du niveau de crédibilité pour les explications concernant la SAAB 900 S 2.0-16 en fonction du nombre d'exemples.

Le niveau de crédibilité défini au Chapitre 3.5 est utile afin de pouvoir déterminer quand il y a assez de pseudo-exemples pour telle ou telle voiture. Ce niveau de crédibilité exprime le degré auquel INKA couvre les valeurs des critères de sélection utilisées pour déterminer les valeurs *ID* et *UD*.

Le niveau de crédibilité est estimé pour ID et UD de chaque critère de sélection et l'indicateur final est la moyenne de ces estimations. L'évolution de cet indicateur en fonction du nombre d'exemples pour la SAAB 900 S 2.0-16 est indiquée dans la Figure 5-2. Il est normal que le niveau de crédibilité augmente après l'introduction du dixième exemple, qui est la SAAB en question. Il reste assez stable jusqu'à l'introduction du premier pseudo-exemple, après quoi la courbe commence à se stabiliser.

L'évolution du niveau de crédibilité des explications de trois autres voitures, la BMW 316i Pack, la Volvo 850 T5 et la SAAB 900 SE 2.0 Turbo, est indiquée dans la Figure 5-3. La BMW 316i Pack appartient à la même classe donnée par la carte auto-organisatrice que la SAAB 900 S 2.0-16. La Volvo 850 T5 et la SAAB 900 SE 2.0 Turbo appartiennent elles aussi à la même classe, cette classe étant choisie pour être le plus différente possible de la première. Il est important de noter que la Volvo 850 T5 et la SAAB 900 S 2.0-16 sont des exemples d'apprentissage.

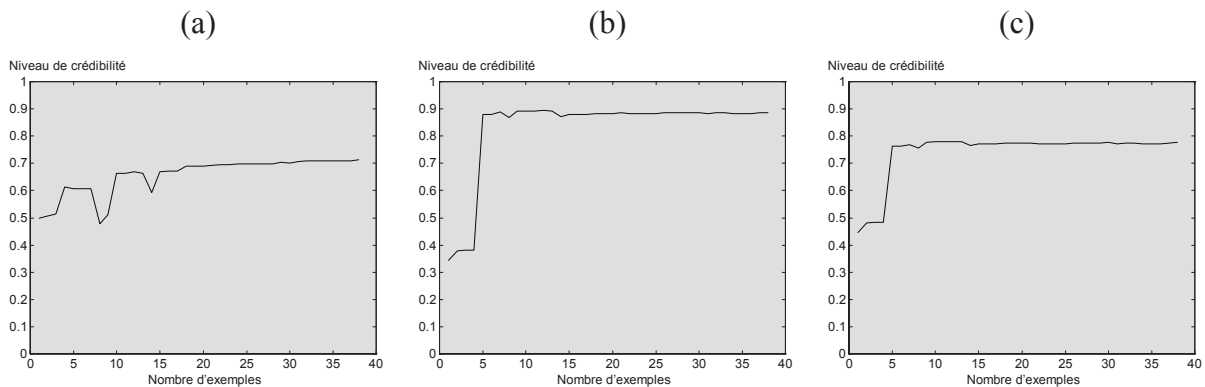


Figure 5-3. Évolution du niveau de crédibilité des explications concernant (a) la BMW 316i Pack, (b) la Volvo 850 T5 et (c) la SAAB 900 SE 2.0 Turbo en fonction du nombre d'exemples.

L'interprétation de ces courbes est assez difficile. Pour la BMW 316i Pack, qui est la voiture la plus proche de la SAAB 900 S 2.0-16, le niveau de crédibilité des explications est le plus petit des trois. L'évolution de cet indicateur est pratiquement identique pour la SAAB 900 S 2.0-16, ce qui confirme la similarité des deux voitures. Le niveau de crédibilité des explications concernant la Volvo 850 T5 augmente très fortement dès son introduction comme exemple d'apprentissage, mais se stabilise malgré les pseudo-exemples ajoutés. Grâce à la similarité, l'évolution est la même pour la SAAB 900 SE 2.0 Turbo.

Chapitre 5. Explications pour l'application « choix de voiture »

Au lieu d'utiliser le niveau de crédibilité absolu comme indicateur, il semble qu'il serait mieux d'examiner sa variation en fonction des pseudo-exemples introduits. L'utilisation de valeurs variables pour le paramètre d_i^2 de INKA pourrait également rendre la mesure plus significative.

Il faut constater qu'il est en général assez difficile d'estimer le degré de pertinence d'une explication. Cela revient à estimer la précision des valeurs maximales et minimales calculées pour la fonction de préférence. Malheureusement, il est impossible de savoir si ces derniers ne vont pas changer radicalement avec une combinaison de valeurs des critères de sélection qui n'est pas encore intégré dans l'ensemble d'apprentissage.

D'une manière générale, il est facile d'accepter une explication bien formulée et bien argumentée, même si elle est complètement fausse. Il faudra un exemple qui contredit l'explication donnée pour comprendre que l'explication était fausse. D'ailleurs, ceci est certainement la raison pourquoi ce n'est pas toujours la meilleure décision qui est prise dans notre société, c'est plutôt la décision qui est soutenue par les meilleurs démagogues.

# Critère	1	2	3	4	5	6	7	8	9	10	11	12	13
Poids linéaire	12	11	6	5	6	6	4	10	10	10	8	5	10
ID BMW	44	7	6	2	7	5	3	7	7	31	8	9	7
ID Volvo	27	22	15	3	17	6	10	20	21	12	20	35	9
ID SAAB	33	20	13	2	15	7	6	18	18	14	15	20	8
UD BMW	0.91	0.12	1.00	1.00	0.62	0.77	0.96	0.26	0.44	0.97	0.76	1.00	0.67
UD Volvo	0.60	0.98	1.00	0.90	0.93	0.99	0.92	1.00	0.99	0.80	1.00	1.00	0.99
UD SAAB	0.70	0.78	1.00	0.00	0.86	0.99	0.96	0.92	0.94	0.98	1.00	1.00	1.00
Utilités	0.74	0.09	1.00	0.50	0.54	0.57	0.76	0.18	0.33	0.82	0.67	1.00	0.57
linéaires des	0.29	0.96	1.00	0.00	0.80	0.60	0.47	1.00	0.97	0.34	1.00	1.00	0.80
trois	0.39	0.68	1.00	0.00	0.76	0.61	0.47	0.83	0.90	0.57	1.00	1.00	0.75

Tableau 5-2. Valeurs de ID, de UD et d'utilités linéaires calculées pour les voitures BMW 316i Pack, Volvo 850 T5 et SAAB 900 SE 2.0 Turbo.

Les explications construites à partir des valeurs ID et UD des trois voitures (Tableau 5-2) illustrent le cas où les explications peuvent être pertinentes et erronées à la fois. Une explication sur la BMW 316i Pack construite à partir de ces valeurs serait encore à peu près correcte. En ce qui concerne les deux autres, les erreurs sont trop grandes sur plusieurs critères pour pouvoir fournir une explication correcte. Une amélioration de la situation nécessiterait l'introduction d'autres pseudo-exemples les concernant.

La détermination d'une règle d'arrêt est dans ce cas plus difficile que pour le SIAD de la partie précédente. Le niveau de crédibilité permet au décideur d'avoir une idée de l'effet obtenu par l'ajout de pseudo-exemples, mais l'interprétation reste difficile. Le fait de fournir des explications textuelles permet au moins au décideur de détecter des incohérences ou des fautes directes. Si le décideur n'est pas du tout d'accord avec les explications qui sont censées être plus ou moins les siennes, il sait qu'il y a un problème. Dans le cas de décideurs abstraits,

Chapitre 5. Explications pour l'application « choix de voiture »

pour qui les explications sont encore plus importantes, il est plus difficile de détecter ce type d'erreurs.

Le besoin potentiel d'un grand nombre de pseudo-exemples peut rendre leur saisie longue et difficile. Dans un grand nombre d'applications il n'est pas toujours nécessaire de fournir des explications sur toutes les actions potentielles. Le faire seulement pour les actions qui sont en tête du classement peut être suffisant. Dans ce cas, les actions en question appartiennent normalement aux mêmes classes données par la carte auto-organisatrice car elles sont, par défaut, assez similaires. L'effort nécessaire de la part du décideur reste ainsi limité et raisonnable. La similarité des actions évoquée à plusieurs reprises nous amène aux explications par similarité, déjà mentionnées au Chapitre 3.4, qui peuvent également permettre de diminuer le nombre d'exemples nécessaire.

5.3 Explications par similarité

Les explications concernant les exemples d'apprentissage peuvent servir de « prototypes » pour expliquer les valeurs de préférence d'actions similaires. Ces explications « prototypes » sont définies directement par le décideur au moment de la sélection de l'exemple d'apprentissage. Une autre possibilité est d'utiliser les explications fournies à partir d'INKA dont on sait qu'elles sont suffisamment correctes, comme l'explication sur la SAAB 900 S 2.0-16 de la Figure 5-1. L'explication sur la BMW 316i Pack indiquée à la Figure 5-4 est construite à partir de cette explication.

La BMW 316i Pack est considérée similaire à la SAAB 900 S 2.0-16. Elles sont moyennes toutes les deux. Les raisons pour que la SAAB 900 S 2.0-16 soit moyenne sont:

« Répétition partielle de l'explication sur la SAAB 900 S 2.0-16 de la Figure 5-1 ».

La BMW 316i Pack est néanmoins meilleure que la SAAB 900 S 2.0-16 (valeur de préférence 59.2 au lieu de 57.3) car elle présente des différences significatives au niveau de traction (traction arrière contre traction avant), capacité du coffre (435 l contre 451 l), poids (1190 kg contre 1285 kg) et équipement en série (57 points contre 50 points), malgré des valeurs moins bonnes au niveau de puissance maximale (102 ch contre 130 ch).

Figure 5-4. Explication par similarité pour la BMW 316i Pack construite à partir d'une explication sur la SAAB 900 S 2.0-16.

Cette explication est une explication comparative pour deux actions qui sont assez similaires. La similarité permet de diminuer la quantité d'information à utiliser. Seuls les critères pour lesquels il y a des différences assez significatives (plus de 0.15 entre les valeurs *UD* des deux voitures) sont inclus dans l'explication.

La Volvo 850 T5 est une voiture très performante, spacieuse et confortable. Elle est également bien équipée et son image de marque est très bonne. Malheureusement, son prix est assez élevé, ce qui est un grand inconvénient. D'autres inconvénients sont une consommation élevée et la traction avant combinée avec une très grande puissance maximale.

La SAAB 900 SE 2.0 Turbo est considérée similaire à la SAAB 900 S 2.0-16. Toutes les deux sont bonnes. Les raisons pour que la Volvo 850 T5 soit bonne sont:
« Répétition partielle de l'explication sur la Volvo 850 T5 ».

La SAAB 900 SE 2.0 Turbo est néanmoins moins bonne que la Volvo 850 T5 (valeur de préférence 69.7 au lieu de 70.0) car elle présente des différences significatives au niveau de puissance maximale (185 contre 225), malgré une valeur plus bonne au niveau de la consommation moyenne (10.40 l/100 km contre 11.70 l/100km).

Figure 5-5. Explication fournie par le décideur sur la Volvo 850 T5 (gauche) et appliquée pour construire l'explication par similarité de la SAAB 900 SE 2.0 Turbo (droite).

Chapitre 5. Explications pour l'application « choix de voiture »

Cette explication contient d'ailleurs une erreur, car la capacité du coffre de la SAAB est plus grande que celle de la BMW. La SAAB a ainsi une meilleure valeur pour ce critère que la BMW, même si les valeurs de *UD* indiquent le contraire. Pour le décideur, il est facile de détecter une telle situation et de la corriger par des nouveaux pseudo-exemples s'il trouve que cette erreur est trop gênante.

Les explications de la Figure 5-5 montrent l'utilisation d'une explication sur la Volvo 850 T5, fournie par le décideur au moment de son choix comme exemple d'apprentissage. Dans ce cas, l'explication prototype est reprise telle quelle pour expliquer le résultat de la SAAB 900 SE 2.0 Turbo. La partie comparative a été ajoutée au début et à la fin de l'explication. Il serait certainement avantageux de mieux analyser la structure de l'explication « prototype », afin de mieux l'intégrer dans l'explication construite.

Seulement deux types d'explication ont été évoqués jusqu'ici: Les explications par importance et utilité contextuelles et les explications par similarité. Les concepts intermédiaires permettaient de construire des explications avec plusieurs niveaux de détail. Le chapitre suivant présentera comment mieux utiliser ces concepts intermédiaires en les intégrant dans d'autres modèles de représentation de connaissances. Ils sont également utilisés pour d'autres applications que l'aide à la décision multicritère.

6. Modèles conceptuels pour un dialogue explicatif

La notion de concept intermédiaire, telle qu'elle était définie au Chapitre 3.3, permet à l'utilisateur de définir ses propres concepts à tout moment. Cela représente un avantage important par rapport aux concepts intermédiaires des systèmes experts, qui sont définis dans une structure fixe (base de règles, réseau sémantique, ...). La définition d'un nouveau concept intermédiaire dans un système expert nécessite une modification de la base de connaissances, ce qui peut difficilement être fait par un utilisateur ordinaire.

Dans la méthodologie d'explication développée, le module de raisonnement et le module d'explication sont complètement indépendants l'un de l'autre. Il est donc possible d'utiliser le nombre de modules d'explication souhaité pour satisfaire les besoins d'explication. Il est, par exemple, possible de définir une structure de concepts intermédiaires qui soit plus détaillée

pour les aspects techniques d'une voiture que pour les autres aspects. Une telle structure conviendra donc à une personne orientée « technique ». Pour une personne orientée plutôt « esthétique », il sera possible d'utiliser une structure de concepts intermédiaires différente et plus adaptée.

Les réseaux sémantiques sont les modèles conceptuels les plus utilisés dans le domaine de l'explication. Le chapitre suivant montre les avantages obtenus si ce type de modèle conceptuel est associé à l'approche neuronale.

6.1 Réseaux sémantiques

Les réseaux sémantiques permettent souvent d'exprimer des connaissances plus variées que les bases de règles [Kassel, 1995]. Le principe consiste à définir une typologie de liens entre les concepts du réseau: « est-un », « fait-partie-de », « dépend-de », etc. Les bases de règles classiques utilisent uniquement le lien « implique-que ».

Les concepts intermédiaires peuvent s'intégrer dans les réseaux sémantiques pour profiter de toute la richesse d'expression offerte par cette technique. Ceci est illustré par un petit exemple de diagnostic de panne de voiture. Le système de diagnostic est illustré dans la Figure 6-1.

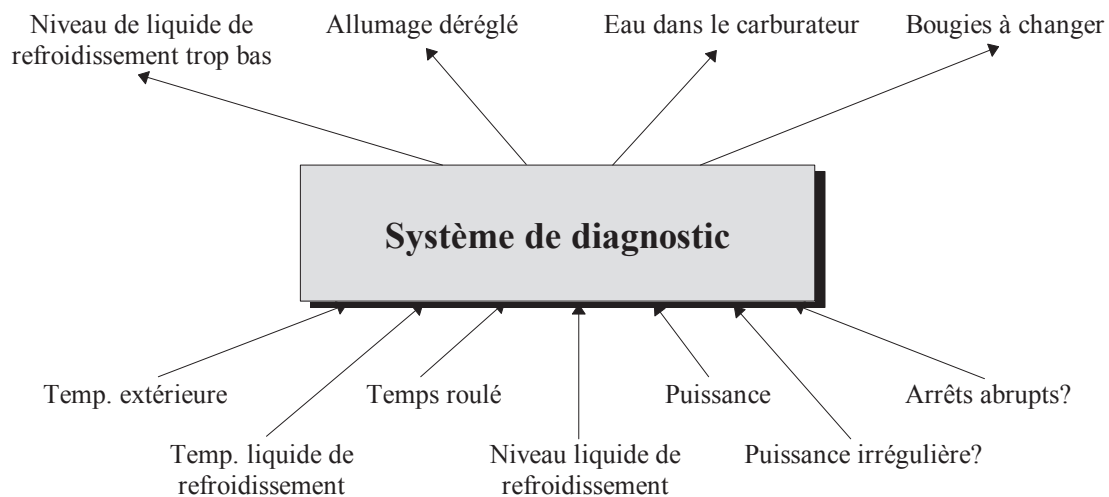


Figure 6-1. Système de diagnostic de panne de voiture.

Les valeurs d'entrée du système sont des mesures quantitatives (température extérieure, température du liquide de refroidissement, temps roulé, niveau de liquide de refroidissement, niveau de puissance) et des indicateurs booléens (présence d'arrêts abrupts, puissance

irrégulière). Les valeurs de sortie sont des valeurs de certitude ou de probabilité des diagnostics différents (niveau de liquide de refroidissement trop bas, allumage déréglé, eau dans le carburateur, bougies à changer). Le système de diagnostic peut être une base de règles, un réseau neuronal, un réseau sémantique, etc. Nous étudions le cas d'un réseau sémantique en premier.

6.1.1 Diagnostic et explication par réseau sémantique

La partie du réseau sémantique concernant le diagnostic « eau dans le carburateur » est illustrée dans la Figure 6-2. Quand ce réseau sémantique est utilisé pour établir un diagnostic, il utilise seulement les liens « implique ». Les autres liens représentent de l'information supplémentaire sur le domaine, qui peut être utilisée dans les explications. Le module de raisonnement (le système de diagnostic dans ce cas) de ce réseau sémantique présente certaines limitations, qui sont essentiellement dûs à sa nature symbolique:

- Le raisonnement se fait normalement de façon booléenne, c'est-à-dire qu'à chaque nœud correspond une valeur soit « vrai », soit « faux ». Des coefficients de certitude ou des probabilités peuvent être associées aux nœuds et liens du réseau sémantique, mais il devient alors nécessaire d'expliquer leur raison d'être et leur propagation. Il est donc difficile de prendre en compte l'incertitude dans les explications.
- Le traitement des valeurs quantitatives se fait par des intervalles de valeurs, ce qui fait que le changement d'état entre vrai et faux se fait de façon abrupte avec de petits changements de valeur. Les ensembles flous peuvent être utilisés, ce qui implique plusieurs chemins d'inférence à expliquer avec leurs coefficients de certitude. Les valeurs quantitatives sont donc difficiles à traiter et leur traitement par ensembles flous (par exemple) pose des problèmes d'explication.
- Le module de raisonnement et le module d'explication sont confondus.

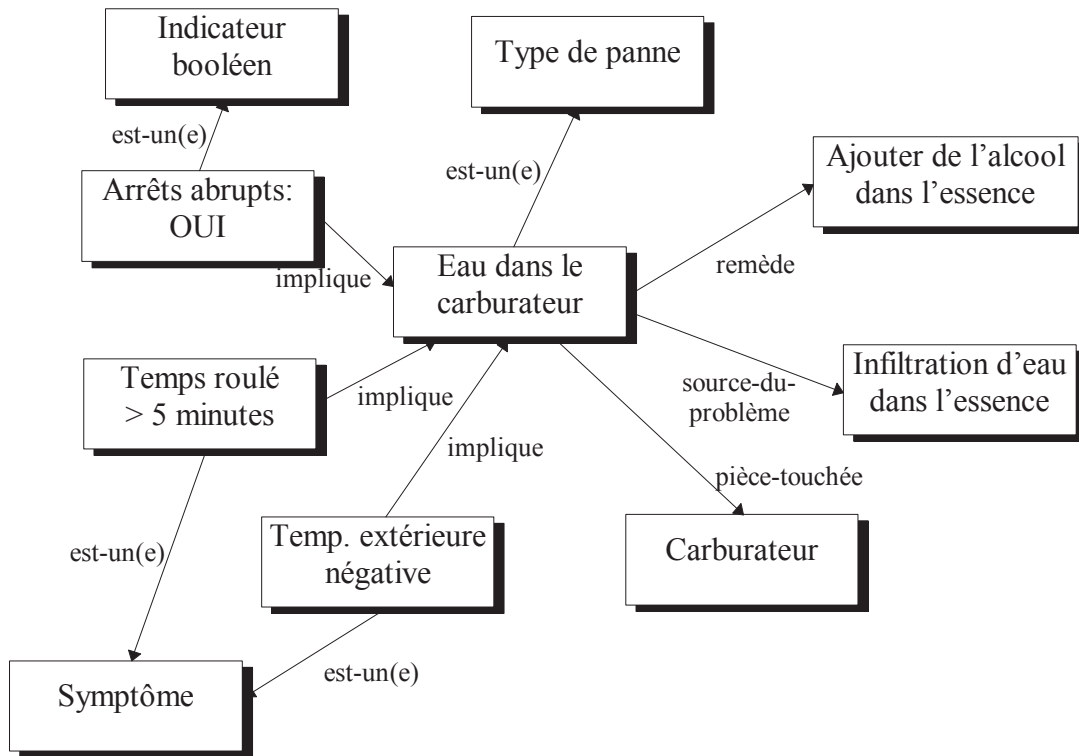


Figure 6-2. Exemple d'un réseau sémantique pour le diagnostic « eau dans le carburateur ».

6.1.2 Diagnostic par réseau neuronal et explication par un réseau sémantique associé

Si le système de diagnostic est un réseau neuronal, chaque sortie du réseau neuronal correspond à une certitude ou une probabilité que le diagnostic soit le bon. Cette valeur de sortie détermine donc le degré d'activation des nœuds correspondant aux différents diagnostics du réseau sémantique. Le réseau sémantique utilisé (Figure 6-3), présente certaines différences à celui de la Figure 6-2. Les différences principales sont les suivantes:

- Chaque entrée du système de diagnostic est connectée à chaque diagnostic par des liens « peut-impliquer », dont la force de connexion dépend du contexte.
- Les nœuds correspondant aux entrées du système de diagnostic sont directement leurs noms au lieu de correspondre à certains intervalles de valeurs d'entrée.
- Le module de diagnostic est complètement séparé du réseau sémantique.

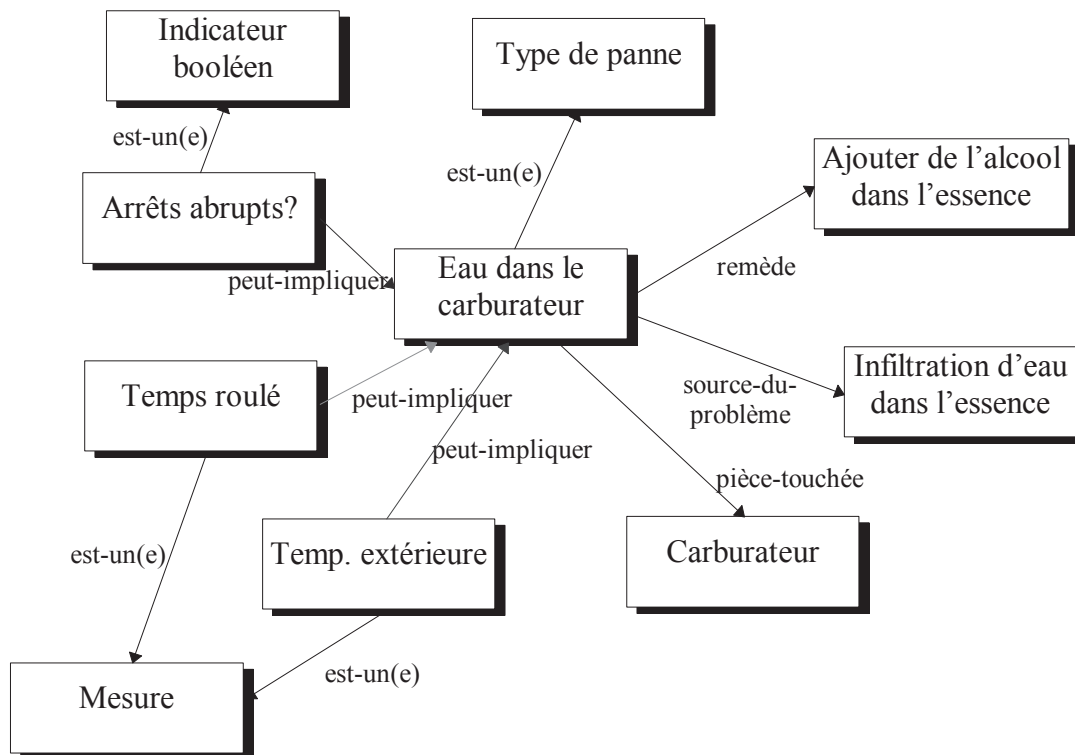


Figure 6-3. Exemple d'un réseau sémantique pour expliquer le diagnostic « eau dans le carburateur » utilisant les concepts intermédiaires « neuronaux ».

Le premier point vient du fait que les liens entre les symptômes et les résultats du diagnostic sont découverts par l'apprentissage. Tous les symptômes sont donc a priori connectés à tous les diagnostics et leur degré de connexion est déterminé par l'apprentissage. Ce degré de connexion change en fonction du contexte traité. L'importance contextuelle sert à estimer le degré de connexion.

Prenons comme exemple une situation où le temps roulé est de 3 minutes 45 secondes, la température extérieure -5° et il y a des arrêts abrupts. Supposons que l'importance contextuelle soit de 45%, 75% et 85% pour ces trois mesures par rapport au diagnostic « eau dans le carburateur » (flèches grisées). Les importances contextuelles estimées pour les liens du réseau sémantique permettent de produire l'explication suivante:

Le diagnostic est qu'il y a de l'eau dans le carburateur avec une probabilité de 87%. Ce diagnostic est avant tout motivé par la présence d'arrêts abrupts et la température extérieure. Le temps roulé est un indicateur supplémentaire.

Les utilités contextuelles ne sont pas utilisées dans cette explication. Elles pourraient servir pour expliquer si une température extérieure de -5° est plutôt un indicateur favorable ou

défavorable (et à quel degré) pour le diagnostic. Le réseau sémantique de la Figure 6-2 ne peut pas fournir de telles explications. D'abord, il ne détectera pas du tout ce diagnostic avant que le temps roulé dépasse 5 minutes, limite pour laquelle la certitude dépasse un certain seuil (75% par exemple). Même s'il est capable d'estimer un coefficient de certitude pour ce diagnostic, il ne pourra pas le justifier de façon convaincante.

Le deuxième point signifie qu'il n'y a pas besoin de définir des intervalles de valeurs pour les besoins du diagnostic. Les intervalles de valeurs n'interviennent que dans les explications, où ils servent à traduire les valeurs quantitatives en valeurs qualitatives. Le réseau sémantique de la Figure 6-3 est donc plus facile à définir et à maintenir que celui de la Figure 6-2.

La connexion entre le module de raisonnement et le module d'explication (le réseau sémantique) se fait uniquement par les importances et les utilités contextuelles. Les importances contextuelles déterminent la « force » des liens « peut-impliquer » et donc les variables d'entrée à utiliser dans les explications. Les utilités contextuelles déterminent si la valeur actuelle d'un symptôme est plutôt favorable ou défavorable pour le diagnostic.

Cet aspect contextuel présente malheureusement un inconvénient. On ne peut expliquer un diagnostic qu'en fonction du contexte, il n'est pas possible d'identifier les connaissances indépendantes du contexte. Il est bien connu que pour diagnostiquer la présence de l'eau dans le carburateur, il faut généralement prendre en compte les trois indicateurs (présence d'arrêts abrupts, température extérieure basse et le temps roulé). Pour le faire avec le réseau sémantique de la Figure 6-3, il faudrait établir des statistiques sur les contextes dans lesquels le diagnostic est établi. Il serait donc possible d'ajouter des liens de type « dépend-souvent-de » avec une indication de la fréquence de dépendance. De tels liens permettent de préserver la séparation entre le module de raisonnement et le module d'explication.

La séparation du module de raisonnement et du module d'explication présente également l'avantage de pouvoir utiliser plusieurs modules d'explication en fonction de l'utilisateur. Ces modules peuvent être des réseaux sémantiques, des arbres de concepts ou des combinaisons des deux. Le traitement de l'incertitude à l'aide des notions d'importances et d'utilités contextuelles devrait même permettre d'augmenter leurs performances. Il est donc possible d'utiliser ce type de modèle conceptuel pour la gestion des dialogues homme-machine, la représentation de connaissances, etc. Spécialement le dialogue homme-machine est un des plus grands défis de la communauté scientifique qui travaille sur les explications.

6.2 Dialogues

Une bonne explication doit offrir des possibilités de dialogue avec l'utilisateur pour lui permettre d'obtenir les explications qui correspondent à ses besoins. Convaincre l'utilisateur est encore plus difficile que d'expliquer une recommandation. Il est souvent nécessaire de revenir en arrière et de ré-expliquer. Il faut donc posséder plusieurs approches différentes pour expliquer le même résultat ou concept.

L'utilisation de fonctions de préférence contextuelle et la prise en compte de l'incertitude permettent de diversifier les approches explicatives. Il devient notamment possible de mieux répondre aux questions de l'utilisateur.

6.2.1 Types de questions

Dans un dialogue explicatif naturel il y a un grand nombre de questions que l'utilisateur est susceptible de poser. Il existe a priori deux types de questions principales. Le premier type concerne les connaissances du domaine et le deuxième concerne les justifications d'une recommandation. Dans [Greboval, 1994] ces deux types sont groupés comme des questions de type QUOI et POURQUOI. Des exemples de questions du type QUOI sont:

- Que signifie le concept X? Réponse: une définition du concept style dictionnaire.
- Quel est le lien entre le concept X et le concept Y? Réponse: explication du lien entre les concepts, s'il y en a un.
- Comment expliquer le concept X avec d'autres mots? Réponse: présentation des synonymes et/ou des liens avec d'autres concepts du domaine.
- Comment a le système de résolution été construit? Réponse: une explication de la démarche utilisée pour l'identification des décideurs, acquisition de connaissances, etc.

Ce type de questions peut être posé à un système qui est complètement indépendant du système de résolution et a des liens forts avec le domaine d'*environnements interactifs d'apprentissage par ordinateur* (EIAO) [Baron et al., 1993]. Tout en étant fortement lié au problème d'explication d'un résultat, ce domaine reste un domaine scientifique qui ne sera pas développé dans ce cadre. Ce type de module ne peut être que bénéfique pour le dialogue

et l'interaction homme-machine en général. Ce sont donc plutôt les questions du type POURQUOI qui seront traitées ici:

Pourquoi l'action X est-elle bonne? Réponse: mettre en évidence les critères de sélection ayant des valeurs favorables pour le choix de l'action. Dans [Greboval, 1994], ce type de réponse s'appelle une justification positive.

Pourquoi l'action Y est-elle mauvaise? Réponse: mettre en évidence les critères de sélection ayant des valeurs défavorables pour le choix de l'action. Dans [Greboval, 1994], ce type de réponse s'appelle une justification négative.

Pourquoi l'action X est-elle meilleure que l'action Y? Réponse: mettre en évidence les critères de sélection pour lesquels X a une meilleure valeur que Y et comparer leur importance par rapport aux critères pour lesquels Y a une meilleure valeur que X.

Que se passe-t-il si la valeur du critère Z change à y au lieu de x pour l'action X?

Réponse: recalculer la valeur de préférence de l'action X, ce qui donne un nouveau classement, qui demande une nouvelle explication.

Quelle valeur faut-il donner au critère Z de l'action Y pour que Y devienne meilleure que X? Réponse: effectuer une analyse graphique du type présentée dans le Chapitre 11, ce qui permet de déterminer directement la réponse à cette question. L'analyse graphique peut également servir en tant que réponse partielle à la question précédente.

Les deux premières questions sont assez classiques et pratiquement toute méthode de résolution permet de donner une réponse, comme cela est montré dans l'application StocDech. La troisième question est déjà moins classique, ce qui vient probablement du fait que les applications classiques des systèmes experts, comme le diagnostic médical, n'en ont pas forcément besoin. Par contre, dans le domaine de l'analyse multicritère, ce type de justification est fondamental. Les explications comparatives de StocDech fondées sur la méthode Electre sont un exemple simple de justifications de ce type, qui peut encore être largement développé.

La réponse à la quatrième question ne pose pas de problèmes dans la plupart des cas. Il suffit de remplacer la valeur en question, refaire le raisonnement et expliquer le nouveau résultat. La réponse à la cinquième question demande une analyse plus approfondie, qu'il est facile d'effectuer dans le cas de modèles numériques. Dans le cas de bases de règles ou d'autres

méthodes de raisonnement non-continues, la précision de cette analyse risque cependant d'être faible étant donné les changements abrupts de la valeur de préférence en fonction des changements de valeur des critères.

Une partie importante de l'apport au domaine d'explication de ce travail de thèse vient de la possibilité de répondre aux trois dernières questions, en plus des deux premières. Plus de questions possibles implique plus de cheminements possibles dans le dialogue. La gestion du dialogue se fait par le modèle d'utilisateur, qui se charge de suivre le dialogue et l'évolution des connaissances de l'utilisateur.

6.2.2 Modèles d'utilisateur

Un *modèle d'utilisateur* est un ensemble d'information sur le comportement typique d'un utilisateur ou d'une catégorie d'utilisateurs pendant le dialogue [Cawsey, 1992]. Il permet d'optimiser ce dialogue de façon à faciliter la compréhension de la justification. Certains utilisateurs préfèrent des illustrations graphiques plutôt que des explications en langage naturel ou des explications par similarité plutôt que des explications sur le raisonnement utilisé. Le modèle de l'utilisateur peut se décomposer en deux parties, qui sont le suivi d'un dialogue spécifique et le suivi de l'utilisateur pendant plusieurs sessions de dialogue.

Mener un dialogue intelligent nécessite au moins de mémoriser ce qui a déjà été expliqué.

Une première raison pour le faire est d'éviter de justifier la même chose plusieurs fois. Il est impossible de savoir à l'avance comment un dialogue va évoluer et il est fort possible qu'il y ait des circuits provoquant des explications identiques aux précédents. Dans ce cas, il faut soit rappeler une explication déjà fournie, soit expliquer la même chose d'une manière différente. Le choix peut dépendre du temps parcouru depuis la dernière explication sur le sujet, pour effectuer le rappel d'une explication « oubliée ».

Le suivi de l'utilisateur pendant plusieurs sessions de dialogue doit permettre d'obtenir des données statistiques sur le succès de tel ou tel type d'explication. Si l'utilisateur met en moyenne plus de temps pour accepter une explication qui utilise des importances que pour en accepter une utilisant les utilités, il vaut mieux commencer par des explications utilisant les utilités. Le modèle d'utilisateur peut modifier les paramètres du module d'explication pour obtenir le type d'explication souhaité.

L'élaboration de modèles d'utilisateurs de ce type n'entre pas dans le cadre de ce travail, mais demeure un projet intéressant. Un très grand nombre de techniques développées pour les systèmes experts sont directement utilisables avec les méthodes développées ici.

Les explications produites à partir de réseaux neuronaux peuvent donc satisfaire les besoins d'explication au moins pour les problèmes d'aide à la décision multicritère. Avec la méthodologie SIAD neuronale développée dans la partie précédente, nous pouvons envisager comment mieux résoudre des problèmes comme le choix de site. Ce sont des problèmes où les décisions doivent être négociées et où l'aspect explication est primordial.

7. Perspectives: Des explications pour les décisions négociées

Dans une application réelle, comme le choix d'un site de stockage de déchets industriels ultimes, effectuer un choix ou proposer un classement n'est pas suffisant. Il est, en plus, nécessaire de réussir à convaincre les autres acteurs du problème que la décision prise est bonne à l'aide d'argumentations, de justifications et de stratégies d'explication. La décision finale est donc le résultat d'un processus de négociation. Or, le grand nombre d'acteurs rend impossible une négociation directe. Il est inimaginable de réunir les centaines d'élus, les experts, les industriels et la population concernée dans une salle de réunion pour aboutir à une action consensuelle.

L'apprentissage des fonctions de préférence permet d'identifier les préférences des groupes de décideurs (décideurs abstraits), un des objectifs principaux de ce travail de thèse. Le processus de négociation peut être largement accéléré par l'utilisation des modèles de préférence des décideurs abstraits, qui permettent de trouver plus facilement des solutions de compromis. Le projet « *SPREAD - Societal preference modelling and decision support for location problems* » est fondé sur ces possibilités.

Il s'agit d'un projet défini par trois partenaires, qui sont l'École des Mines de Saint-Étienne (Laboratoire Ingénierie de l'Environnement), l'Université Catholique de Louvain (CESAME) et l'Université de Technologie de Helsinki (Laboratoire d'Analyse de Systèmes). SPREAD

était soumis à une évaluation par la Commission Européenne pour l'obtention d'une aide financière dans le cadre du programme « ENV 2 - Environnement et Climat » de 1995.

7.1 Objectifs du projet

L'objectif de SPREAD est d'étudier comment améliorer l'acceptabilité de décisions concernant des problèmes de choix de site en général. Les problèmes de choix de site traités ne concernent pas uniquement des problèmes de stockage de déchets. Ils concernent également le choix de site pour d'autres installations d'utilité publique, comme des centres de commerce, des banlieues et des espaces de récréation. L'incompatibilité entre l'avantage social et les intérêts de la population du site est une caractéristique typique pour ces problèmes.

« Maximiser les avantages de la société » consiste à prendre en compte les aspects techniques, économiques et environnementaux de chaque site. Les critères de sélection concernant ces aspects sont pratiquement les mêmes pour chaque problème de choix de site. Les décideurs représentant les intérêts de la société peuvent être appelés des *décideurs de premier niveau*.

Augmenter l'acceptabilité concerne avant tout la population touchée par la décision. Les décideurs représentant les intérêts de la population locale peuvent être appelés des *décideurs de second niveau*. Augmenter l'acceptabilité consiste à identifier les moyens d'incitation dont disposent les décideurs de premier niveau pour compenser les inconvénients des décideurs de second niveau dans le processus de négociation.

La première étape consiste d'abord à sélectionner un ensemble initial de sites par les décideurs de premier niveau. Une *commission de concertation* est créée par la suite pour établir le contact avec la population concernée, ce qui est très difficile en pratique. Dans le cas d'un échec, il devient difficile ou impossible de mettre la décision en œuvre. Les délais introduits par ce type de problèmes coûtent cher et peuvent même être dangereux pour l'environnement.

La solution finale sera, pour cette raison, choisie en fonction des réactions de la population et des résultats des négociations [Gailly et al., 1993]. La nature des moyens d'incitation utilisés est un critère permettant d'évaluer la réussite des négociations. Les moyens les moins souhaitables sont les ordonnances, les décrets, le recours à la force publique, etc. Le processus

de négociation devrait notamment permettre d'éviter ce genre de situations. Ceci est l'objectif de la méthodologie proposée dans le projet SPREAD.

7.2 Méthodologie proposée

SPREAD devrait permettre de trouver et de définir des approches plus ou moins standards pour des problèmes de choix de site. Le choix d'un site de stockage de déchets industriels ultimes dans la région Rhône-Alpes en est un. Les expériences déjà acquises en la matière montrent que l'interaction entre les acteurs du problème est l'élément essentiel pour la réussite d'un tel choix, peu importe le SAD ou SIAD utilisé.

L'approche SPREAD contient un ensemble de connaissances sur la conduite de projets de choix de site, d'outils et de données permettant une mise à jour et une amélioration constante de l'approche utilisée et de ses composants. Ceci est nécessaire pour prendre en compte les futurs changements dans la technologie, la société et l'environnement. La mise à jour se fait à l'aide d'un suivi des cas traités pour estimer dans quel sens l'approche proposée doit être modifiée (Figure 7-1).

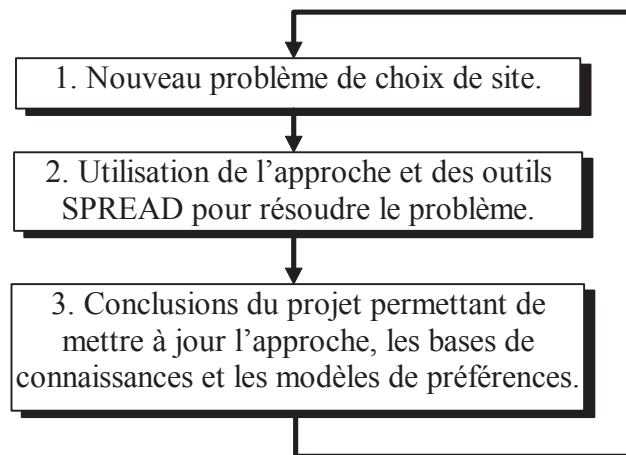


Figure 7-1. L'approche SPREAD.

La phase n° 2 est présentée plus en détail dans la Figure 7-2. Les expériences des cas déjà traités permettent d'identifier les acteurs, les structures décisionnelles et d'initialiser les modèles de préférence, ce qui permet d'aborder le problème plus efficacement. L'étape suivante est l'interaction avec les décideurs de premier niveau pour identifier et améliorer les modèles de préférence correspondants. Les modèles de préférence permettent par la suite

d'utiliser un SAD ou un SIAD pour choisir un ensemble initial de solutions. Ce choix peut déjà nécessiter un processus de négociation.

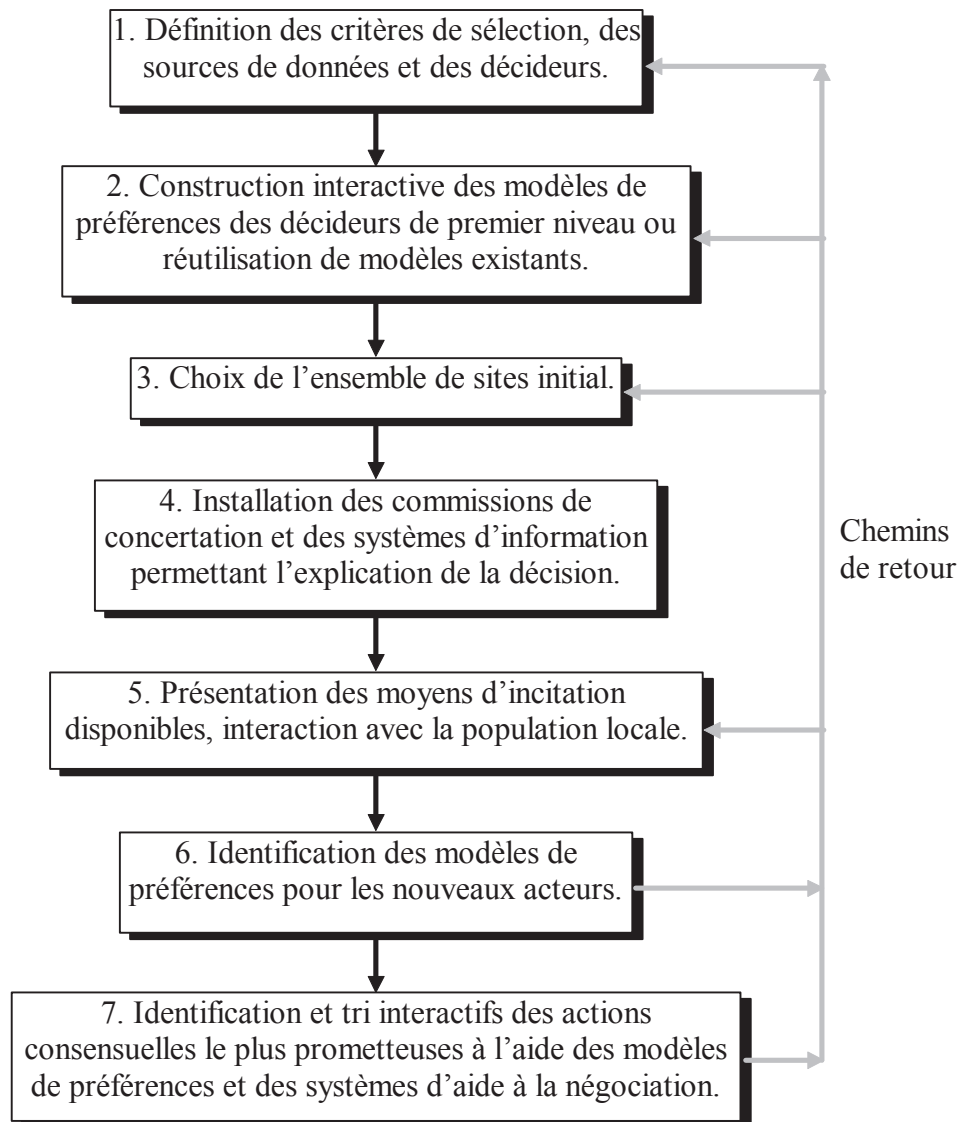


Figure 7-2. Procédure de décision pour le choix de site.

Après avoir identifié l'ensemble initial des solutions, il faut procéder à la publication de ces solutions. Il est très important de justifier toute de suite le choix effectué aussi bien que possible, ce qui implique une installation rapide des commissions locales et des systèmes d'information permettant la justification du choix. Il a été prévu d'utiliser des moyens de communication récents comme le réseau Internet et des services WWW (*World Wide Web*) pour faciliter l'installation et diminuer le coût de ces systèmes d'information.

L'interface directe avec la population locale par l'intermédiaire des systèmes d'information est destinée à permettre une acquisition directe de leurs opinions et préférences. Ceci est

préférable plutôt que de s'en apercevoir à partir d'autres sources d'information, comme des manifestations ou des articles défavorables dans les journaux. L'information acquise doit servir à la construction de modèles de préférence des décideurs de bas niveau, utilisables dans la phase n° 7.

La procédure de négociation de cette étape est facilitée par les modèles de préférence de tous les acteurs, car ils permettent à tous les décideurs d'étudier et de mieux comprendre les motivations des autres. L'analyse de préférences permettra également d'étudier l'effet potentiel des moyens d'incitation sur les opinions, tout dans l'objectif d'identifier ceux qui sont les plus efficaces.

Le rôle des méthodes d'aide à la décision et d'explication présentées dans le cadre de ce travail est de mieux comprendre les préférences des acteurs, mais également de faciliter la tâche de négociation. Le logiciel ADELAIDE est un outil qui intégrera la totalité de ces méthodes et qui permettra de traiter les modèles de préférence et les explications correspondantes. Ce logiciel s'intégrera dans la méthodologie SPREAD pour les parties SIAD, explication et négociation.

7.3 Implémentation informatique

ADELAIDE est l'abréviation de « Aide à la DÉcision par L'Apprentissage et l'Intégration D'Explications », (en anglais: « *Automated DEcisions by Learning And Integrated Decision Explanation* »). ADELAIDE est un logiciel partiellement fondé sur StocDech qui regroupe l'ensemble des méthodes développées dans ce travail.

7.3.1 Rôle d'ADELAIDE dans la méthodologie SPREAD

L'utilisation d'ADELAIDE se situe dans les étapes 2, 3, 5, 6 et 7 de la méthodologie de SPREAD (Figure 7-2), ce qui montre qu'il est un outil essentiel de la méthodologie. Dans l'étape 2, il est utilisé comme un SIAD pour identifier les fonctions de préférence des décideurs de premier niveau. Les fonctions de préférence apprises sont utilisées pour aider les décideurs à identifier l'ensemble de sites initial.

Les besoins d'explication interviennent avant tout dans l'étape 5, où il faudra commencer à justifier le choix auprès de la population locale. Dans cette étape ADELAIDE devrait être

Chapitre 7. Perspectives: Des explications pour les décisions négociées

utilisé en parallèle avec d'autres systèmes d'information pour permettre une campagne d'information aussi efficace que possible. Ce moment d'interaction avec la population locale est le moment idéal pour essayer de comprendre le mieux possible leurs préférences, ce qui est l'objectif de l'étape 6. Pour le faire, il n'est pas possible d'utiliser directement l'interface d'ADELAIDE. L'interaction se fera certainement par l'intermédiaire des enquêtes ou d'hommes d'étude afin de convertir l'information à un format compréhensible par ADELAIDE.

Dans le processus de négociation de l'étape 7, les modèles de préférence appris par ADELAIDE seront utilisés pour simuler les préférences des acteurs. Dans le cas idéal, le processus de négociation peut être largement accéléré, car il est possible d'identifier automatiquement les actions ayant les meilleures possibilités d'être consensuelles.

Un tel outil doit être capable de traiter un grand nombre de données différentes et de les présenter à un grand nombre d'utilisateurs. Les chapitres suivants montrent une première proposition d'architecture informatique pour permettre à ADELAIDE de répondre à ces besoins.

7.3.2 Architecture informatique

Un premier objectif de cet outil est d'être modulable et facilement extensible à de nouvelles méthodes d'aide à la décision. L'interface et la présentation des résultats doivent être cohérents, quelle que soit le SAD ou le SIAD. Un deuxième objectif consiste à inclure les fonctionnalités méthodologiques nécessaires à l'apprentissage de modèles de préférence des décideurs, et à intégrer l'interface nécessaire à l'acquisition de ces préférences. Les modules envisageables dans ADELAIDE sont au moins les suivants:

- Base de données et interface correspondante.
- Module de présentation et d'analyse des résultats.
- Module d'explication.
- Acquisition des paramètres des méthodes (acquisition de connaissances), comme les poids, les exemples d'apprentissage, etc.
- Modules d'apprentissage.

Il peut être assez difficile de trouver une seule interface utilisateur pour les méthodes « classiques » et pour les techniques neuronales. Au niveau de technique de programmation, une méthodologie orientée objet semble être la plus adaptée, car les objets permettent de « cacher » les différences internes des méthodes d'aide à la décision sous une couche commune [Goldberg & Robson, 1983], [Goldberg, 1984], [Främling, 1990] et [Främling & Hammarström, 1993a, b].

7.3.3 Bases de données

Les données à traiter sont très variées:

- Valeurs brutes des critères de sélection sur les actions potentielles.
- Valeurs d'utilité des critères de sélection pour les actions potentielles.
- Valeurs de préférence (résultats) des actions potentielles, en fonction de la méthode.
- Exemples d'apprentissage, comprenant les valeurs des critères et les valeurs de préférence correctes.
- Paramètres propres à chaque méthode utilisée.

Il est difficile de « stocker » les paramètres propres à chaque méthode sous un seul format. Il devient, pour cette raison, difficile d'utiliser l'approche de StocDech, où il y avait plusieurs bases de données, une pour les données des actions potentielles et certains résultats, une pour les valeurs d'utilité, une pour les résultats de la méthode Electre et une base de règles. Le nombre de base de données deviendrait trop grand et leur cohérence serait difficile à maintenir. Une approche utilisant une base de données bien structurée sous forme d'un seul fichier semble pour cette raison être la meilleure solution.

Les données doivent être accessibles à l'aide de fonctions suffisamment générales pour correspondre aux besoins de chaque méthode utilisée. Ceci revient à normaliser les données d'une façon cohérente qui convienne aux méthodes d'analyse multicritère, aux bases de règles et aux réseaux de neurones.

7.3.4 Acquisition des paramètres des méthodes d'aide à la décision

La conception d'une interface de saisie unique est difficile à cause la grande disparité des paramètres propres à chaque méthode. Il est possible de regrouper des fonctions communes à chacune des méthodes, comme l'interface de saisie des poids et des fonctions d'utilité. L'acquisition d'autres paramètres s'effectuera à l'aide de modules supplémentaires, spécifiques à chaque méthode.

En ce qui concerne la saisie d'exemples d'apprentissage, l'interface graphique du réseau neuronal INKA a déjà été présentée. Ce type d'interface est valable pour toute méthode numérique (réseaux neuronaux), elle ne l'est pas pour les méthodes symboliques. Il peut difficilement être utilisé directement pour l'interaction avec la population concernée par un choix de site, par exemple, à condition de développer des interfaces encore plus conviviales.

7.3.5 Présentation et analyse des résultats

Les différences de présentation des résultats concernent avant tout les explications générées par les méthodes différentes. Les concepts d'importance et d'utilité contextuelles permettent d'utiliser la même approche pour la plupart des méthodes, mais une explication à partir d'une base de règles, par exemple, restera plus facile à construire utilisant les règles. Il est également difficile de trouver des moyens d'analyse graphique des résultats d'une base de règles.

La gestion de l'interaction avec les acteurs non initiés aux méthodes d'analyse multicritère demeure un problème à résoudre. Les fonctionnalités de dialogue devraient pour cette raison être beaucoup développées encore. Le nombre de questions et la quantité d'information nécessaire pour bien réussir cette tâche nécessitera une intégration avec d'autres systèmes d'information environnementaux. Ceci est peut-être l'aspect d'ADELAIDE le plus important pour pouvoir mettre en œuvre la méthodologie de SPREAD.

8. Conclusion

Les réflexions conduites dans cette troisième partie montrent l'intérêt des explications dans les SIAD, en particulier pour des applications à grande échelle, où le nombre de décideurs et/ou d'actions est grand. Les explications sont même nécessaires dans le cas où le modèle de préférence a été appris par le SIAD pour que la pertinence du modèle appris puisse être vérifiée.

L'objectif de cette partie est avant tout de montrer les possibilités d'expliquer les résultats des réseaux de neurones. Les notions d'importance et d'utilité contextuelles proposées définissent une base uniforme d'explication, non seulement pour les réseaux de neurones, mais aussi pour d'autres types de systèmes d'aide à la décision. Les explications par similarité et l'analyse graphique constituent d'autres possibilités d'explication.

Toutefois, ceci ne suffit pas pour obtenir un système explicatif complet. Un tel système demanderait encore d'inclure d'un grand nombre de fonctionnalités, comme un meilleur module de langage naturel, un modèle d'utilisateur et un gestionnaire de dialogues. Ce sont néanmoins des composants qui sont assez indépendants du modèle de raisonnement utilisé et qui peuvent fonctionner de la même façon que dans d'autres systèmes d'explication, comme les systèmes experts de deuxième génération. La flexibilité des explications proposées peut, par ailleurs, augmenter la qualité de ces modules.

La possibilité d'obtenir des explications permet d'utiliser des SIAD pour des nouveaux types de problèmes. Il est, par exemple, possible d'apprendre les préférences des consommateurs pour les analyser et pour déterminer les améliorations optimales d'un produit, un site de stockage de déchets, une centrale nucléaire, une entreprise, etc. Pour augmenter les ventes d'une voiture, par exemple, il peut parfois être plus efficace d'augmenter le niveau d'équipement plutôt que diminuer le prix, car le rapport coût/bénéfice est meilleur.

Les nouvelles méthodes ou techniques qui ont été développées dans cette troisième partie de ce travail sont donc :

Chapitre 7. Perspectives: Des explications pour les décisions négociées

- Les fonctionnalités d'explication développées pour les SAD, fondées sur les notions d'importance et d'utilité contextuelles, permettant d'expliquer les résultats des fonctions de préférence contextuelle.
- Explication des résultats des réseaux de neurones par ces mêmes notions d'importance et d'utilité contextuelles.
- Extension de ces notions aux concepts intermédiaires, qui permettent de structurer les explications et d'obtenir des explications de plusieurs niveaux de détail.
- Moyens d'utiliser les modèles de préférence comme des acteurs dans les processus de négociation, où les fonctions de préférence contextuelle permettent de modéliser les préférences des décideurs mieux qu'avant. Dans un tel processus, les aspects d'explication et de justification sont primordiaux.
- Analyse des préférences des décideurs (consommateurs, par exemple) pour déterminer les moyens optimaux pour un producteur d'actions potentielles d'augmenter les possibilités de choix de certaines actions.

Enfin, les principes d'explication ne sont pas limités aux problèmes d'aide à la décision multicritère. Il est tout à fait imaginable de les utiliser dans d'autres domaines d'application, comme le diagnostic médical. Il s'agit, dans ce cas, d'un problème de classification où le module de raisonnement détermine une probabilité de vérité pour chaque diagnostic potentiel du cas traité. La probabilité (ou une autre mesure de certitude) est une valeur numérique qui peut être expliquée de la même façon qu'une valeur de préférence.

Nous pouvons constater que les méthodes d'explication décrites ici sont, en principe, applicables à toute méthode ayant des résultats numériques, y compris les probabilités, les certitudes, les valeurs de sortie des systèmes de contrôle, etc. Ces aspects restent encore à étudier plus en détail.

Conclusion générale

L'objectif de ce travail était d'étudier et de montrer les possibilités d'utilisation des réseaux de neurones dans le domaine de l'aide à la décision multicritère. Pour atteindre cet objectif, il a été nécessaire d'étudier non seulement les réseaux de neurones, mais aussi un grand nombre d'autres outils scientifiques, comme l'analyse multicritère, l'analyse de données, l'identification et l'approximation de fonctions, les systèmes experts et les dialogues homme-machine. Les méthodes relevant de ces domaines ont leurs avantages, mais aucune ne se distingue par ses capacités à proposer une solution optimale pour résoudre un problème complexe d'aide à la décision. L'approche neuronale trouve une place naturelle dans le traitement de problèmes à grande échelle caractérisés par le grand nombre de décideurs et/ou d'actions potentielles, mais elle présente également des avantages pour d'autres problèmes de décision.

Le SAD neuronal décrit dans la première partie montre que l'approche neuronale est réaliste. Les résultats obtenus sont aussi bons que ceux obtenus par les SAD plus classiques, sans nécessiter un plus grand travail de la part du décideur. Exprimer ses préférences comme des poids et des fonctions d'utilité, des règles ou d'autres moyens d'expression directs, plutôt que par des exemples d'apprentissage, dépend du problème traité et du décideur. Par contre, les fonctions de préférence contextuelle sont plus facilement exprimables à l'aide d'exemples d'apprentissage. L'expression explicite des préférences contextuelles à l'aide d'une fonction mathématique ou d'une base de règles est plus difficile et se complique davantage dans le cas de décideurs abstraits (les groupes d'acteurs, l'environnement, l'économie, ...). La possibilité d'utiliser des fonctions de préférence contextuelle doit permettre de traiter un grand nombre de problèmes de décision d'une façon plus réaliste.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

Les exemples d'apprentissage permettent d'obtenir des informations aussi bien sur les préférences des décideurs abstraits que sur les préférences des décideurs « réels ». Le choix d'exemples d'apprentissage est important pour les résultats de l'apprentissage. La méthode de sélection des exemples à l'aide d'une carte auto-organisatrice permet de guider le décideur dans le choix de ces exemples.

La méthodologie d'un SIAD neuronal présentée dans la deuxième partie permet au décideur de vérifier que la fonction de préférence apprise est bonne. Si ce n'est pas le cas, il peut la corriger de façon interactive par l'introduction de pseudo-exemples, qui sont faciles et rapides à introduire par une interface graphique. Les indicateurs de précision et de stabilité utilisés permettent au décideur d'estimer le besoin et l'utilité de pseudo-exemples supplémentaires et de proposer une approche réaliste au niveau du nombre d'exemples d'apprentissage nécessaire.

Les méthodes présentées dans la troisième partie permettent de produire des explications sur les résultats d'un SAD multicritère et d'un réseau neuronal qui sont de qualité égale ou supérieure à celles produites par les systèmes experts. Les notions d'importance et d'utilité contextuelles définies dans cette partie sont applicables à pratiquement tout système d'aide à la décision. Elles peuvent également être combinées avec les modèles conceptuels existants (comme les réseaux sémantiques). Un des plus grands obstacles pour l'utilisation des réseaux de neurones en tant qu'outil d'aide à décision est donc désormais pratiquement franchi. Les limitations principales concernent d'une part le nombre d'exemples d'apprentissage nécessaire et, d'autre part, les temps de calcul sensiblement diminués dans le cas du réseau neuronal INKA.

Pour pouvoir estimer la valeur générale de l'ensemble de méthodes développées pendant ce travail de thèse, il faudrait les appliquer à d'autres problèmes de choix. Une telle estimation rend nécessaire le développement et l'implémentation d'une architecture d'intégration. Ceci est l'objectif du logiciel ADELAIDE présenté dans la troisième partie. Il serait également intéressant d'appliquer les méthodes développées aux problèmes de commercialisation, où l'analyse des préférences des consommateurs peut permettre aux entreprises de mieux établir leur stratégie de marché en comprenant mieux les besoins de leurs clients et en ciblant mieux leurs produits.

Il reste malgré tout un grand nombre d'améliorations envisageables. Les capacités d'identification et d'approximation de fonctions du réseau neuronal INKA peuvent encore être améliorées. La partie où il reste certainement le plus de travail à faire concerne la partie explications. Le travail présenté traite avant tout les capacités d'explication des réseaux de neurones par rapport à celles d'autres méthodes, mais les aspects langage naturel, gestion des dialogues et interface utilisateur, entre autres, restent à approfondir.

Références bibliographiques

ANDREWS, R., GEVA, S. Rule extraction from a constrained error back propagation MLP. *Proceedings of the 5th Australian Conference on Neural Networks, Brisbane Queensland, 1994.* pp 9-12.

ANDERSON, J. A., SILVERSTEIN, J. W., RITZ, S. A., JONES, R. S. Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological Review*, 1977, Vol. 84, N° 5, pp. 413-451.

AYEL M., ROUSSET M.-C. *La coherence dans les bases de connaissances.* Toulouse: Cepadues-editions, 1990. 106 p.

BARON, Monique, GRAS, Régis, NICAUD, Jean-François (Ed.). *Environnements Interactifs d'Apprentissage avec Ordinateur.* Actes des Troisièmes journées EIAO de Cachan, 3-5 Février 1993. Paris: Eyrolles, 1993. 264 p.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

BAXT, William G., WHITE, Halbert. Bootstrapping Confidence Intervals for Clinical Input Variable Effects in a Network Trained to Identify the Presence of Acute Myocardial Infarction. *Neural Computation*, 1995, Vol. 7, N° 3. pp. 624-638.

BEAUNE, Philippe. *Acquisition interactive de la connaissance*. Thèse d'université: U.S.T.L. Montpellier, 1992. 215 p.

BENAYOUN, R., DE MONTGOLFIER, J., TERGNY, J., LARICHEV, O. Linear programming with multiple objective functions: STEP Method (STEM). *Mathematical Programming*, 1971, Vol. 1. pp. 366-375.

BERTIER, P., BOUROUCHE, J.M. *Analyse des données multi-dimensionnelles*. Paris: Presses universitaires de France, 1975. 270 p.

BLAYO, François, TOUZET, Claude, MAGNIER, Jeanine (Ed.). *IIème école de printemps: Neurosciences et sciences de l'ingénieur NSI'93*. Pont-du-Gard, France, Mai 1993. 250 p.

BOURI, Mounia. *Les explications dans les systèmes experts: Étude synthétique. Extension au raisonnement incertain: Le système SERIN*. Thèse d'université: Université Paul Sabatier, Toulouse, 1990. 195 p.

BOURI, Mounia, DIENG, Rose, KASSEL, Gilles, SAFAR, Brigitte. Vers des systèmes experts plus explicatifs. *Actes des 3e journées nationales PRC-GDR Intelligence artificielle. LAFORIA, Paris, France, 1990*. pp. 340-355.

BOYCE, D. E., FARHI, A., WEISCHEDEL, R. *Optimal Subset Selection - Multiple Regression, Interdependence and Optimal Network Algorithms*. Berlin: Springer-Verlag, 1974. 187 p.

BRANS, J.-P., MARESCHAL, B., VINCKE, Ph. PROMETHEE: a New Family of Outranking Methods in Multicriteria Analysis. In: *Operational Research '84*. Edited by J.-P. Brans. Amsterdam: Elsevier Science Publishers B.V., 1984. pp. 408-421.

BREZILLON, P. (Ed.). *Proceedings of the ECAI-92 Workshop W15 « Improving the Use of Knowledge-Based Systems with Explanations »*, Vienna, Austria, August 4, 1992. 134 p.

BRODHAG, Christian. Le développement soutenable. *Annales des Mines*, 1992, Juillet-Août. pp. 22-26.

BUCHANAN, Bruce G., SHORTLIFFE, Edward H. *Rule-Based Expert Systems - The MYCIN Experiments of the Stanford Heuristic Programming Project*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1984. 748 p.

CATHERIN, Jean-Yves. Le flou japonais passe aux commandes. *Mesures*, 1991, N° 636. pp. XXIV-XXVIII.

CAWSEY, Alison. *Explanation and Interaction*. Cambridge, Massachusetts: MIT Press, 1992. 232 p.

CELEUX, G., DIDAY, E., GOVAERT, G. LECHEVALLIER, Y., RALAMBONDRAIN, H. *Classification automatique des données*. Paris: Dunod, 1989. 285 p.

CHANDRASEKARAN, B. Generic tasks in knowledge-based reasoning. *IEEE Expert*, 1986, Vol. 1, N° 3. pp. 23-30.

CLANCEY, W. The epistemology of a rule-based expert system: A framework for explanation. *Artificial Intelligence*, 1983, Vol. 20, N° 3. pp. 215-251.

COLLETTE, S., INSTALLÉ, M. Tools for Multicriteria Decision Making in a Multiactor Context that may be Applied to a Hazardous Plant Location Problem. *Proceedings of the 5th IFIP Conference on Computer Support for Environmental Impact Assessment, Como, Italy, 6-8 October 1993*. pp. 169-175.

DE JONG, Kenneth Alan. *An Analysis of the Behaviour of a class of Genetic Adaptive Systems*. Thèse de doctorat, University of Michigan, Ann Arbor, Michigan, 1975. 266 p.

DE KLEER, J. Choices without backtracking. *Proc. of the Amer. Assoc. for Artificial Intelligence Conf. (AAAI-84), Austin, 1984*. p. 79-85.

DE KLEER, J. An assumption-based truth maintenance system. *Artificial Intelligence*, 1986, N° 28. pp. 127-162.

DE KLEER, J. Extending the ATMS. *Artificial Intelligence*, 1986, N° 28. pp. 163-196.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

DECAMP, Erik, AMY, Bernard. *Neurocalcul et réseaux d'automates*. Nanterre: EC2, 1988. 198 p.

DECO, Gustavo, EBMEYER, Jürgen. Coarse Coding Resource-Allocating Network. *Neural Computation*, 1993, Vol. 5, N° 1. pp. 105-114.

DEMUTH, Howard, BEALE, Mark. *Neural Network Toolbox for Use With Matlab User's Guide*. Massachusetts: MathWorks Inc., 1993. 122 p.

DEVILLERS J., KARCHER W. (Ed.). *Applied Multivariate Analysis in SAR and Environmental Studies*. Dordrecht: Kluwer academic publishers, 1991. 530 p.

DIEDERICH, J. Explanation and artificial neural networks. *International Journal of Man-Machine Studies*, 1992, Vol. 37. pp. 335-357.

DONG, Cao. *Developpement d'outils d'aide à la décision dans des environnements multicritères, dynamiques et incertains: Application à des problèmes de planification régionale*. Thèse Doct. Ing.: Université Catholique de Louvain, Louvain-la-Neuve, Belgique, 1992. 169 p.

DOYLE, J. A thruth maintenance system. *Artificial Intelligence*, 1979, N° 12. pp. 231-272.

DREYFUS Hubert L. From Socrates to Expert Systems: The Limits and Dangers of Calculative Rationality. *Arpakannus*, 1991, N° 3. Finnish Artificial Intelligence Society. pp. 17-25.

FRANTZIS I. E.I.A. In the Landfill Sites Selection Process. *Proceedings of the Sardinia 91 Third International Landfill Symposium, Vol. 2*. Imola: Grafiche Galeati, 1991. p. 1611-1614.

FRÄMLING, Kary. *Using object oriented models for the scheduling of production in petrochemical industries*. MSc. thesis : Université de technologie à Helsinki, 1990. 86 p.

FRÄMLING, Kary. *Les réseaux de neurones comme outils d'aide à la décision floue*. Mémoire de D.E.A. (Diplôme d'Etudes Approfondies), Institut National de Sciences Appliquées de Lyon, Ingénierie Informatique, 1992. 55 p.

FRÄMLING, Kary. L'apport des réseaux de neurones dans un système d'aide à la décision. *Actes des cinquièmes journées annuelles de l'ACTH: VALGO'93 Larnas, France, 25-29 Mai 1993.* pp. 58-65.

FRÄMLING, Kary, HAMMARSTRÖM, Leif. Object-oriented tool for modeling, simulation and production planning in petrochemical industries. *Actes du congrès Representation Par Objets (RPO'93), La grande motte, France, 17-18 June 1993.* pp. 169-180.

FRÄMLING, Kary, HAMMARSTRÖM, Leif. A Distributed Heuristic Expert System for Simulation and Production Planning in Petrochemical Industries. *Actes du workshop "Knowledge-based Production Planning, Scheduling and Control" du congrès IJCAI'93. Chambéry, France, 29 August - 3 September 1993.* pp. 149-158.

FRÄMLING, Kary, GRAILLOT, Didier. *Système d'Aide à la Décision pour le choix de centres de stockage de déchets - StocDech, Guide d'installation, Guide de l'utilisateur, Guide de l'administrateur, Description des méthodes d'aide à la décision utilisées.* Saint-Etienne: Ecole Nationale Supérieure des Mines, France, 1994. 91 p.

FRÄMLING, Kary, GRAILLOT, Didier. Extracting Explanations from Neural Networks. *Actes du congrès ICANN '95, Vol. 1, Paris, France, 9-13 Octobre, 1995.* Paris: EC2 & Cie, 1995. pp. 163-168.

FRÄMLING, Kary, GRAILLOT, Didier, INSTALLÉ, Michel. Apprentissage interactif des fonctions de préférences par réseaux de neurones. *Résumés des communications du congrès FRANCORO'95.* Mons, Belgique, 11-14 Juin 1995. p. 41.

FU, L. M. Rule generation from neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 1994, Vol. 28, N° 8. pp. 1114-1124.

GAILLY, B., BOULIER, F., INSTALLÉ, M. Decision support tools using hierarchical multiperson multicriteria model. General procedure and application to a problem of rural development. *Proceedings of the 12th IFAC World Congress, Vol. 10, Sydney 19-23 July 1993.* pp. 119-121.

GALLANT, Stephen I. *Neural Network Learning.* Cambridge, Massachusetts: MIT Press, 1993. 365 p.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

GENESERETH, Michael R., NILSSON, Nils J. *Logical Foundations of Artificial Intelligence*. Los Altos, California: Morgan Kaufmann Publishers Inc., 1987. 401 p.

GEOFFRION, A., DYER, J., FEINBERG, A. An interactive approach for multi-criterion optimization, with an application to the operation of an academic department. *Management Science*, 1972, Vol. 19, N° 4. pp. 357-368.

GILES, C. L., OMLIN, C. W. Rule refinement with recurrent neural networks. *Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, 1993*. pp. 801-806.

GILES, C. L., OMLIN, C. W. Extraction, insertion, and refinement of symbolic rules in dynamically driven recurrent networks. *Connection Science*, 1993, Vol. 5, N° 3 and 4. pp. 307-328.

GOLDBERG, Adele. *Smalltalk-80: The Interactive Programming Environment*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1984. 516 p.

GOLDBERG, Adele, ROBSON, David. *Smalltalk-80: The language and its implementation*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1983. 714 p.

GORINEVSKY, Dimitry, CONNOLLY, Thomas H. Comparison of Some Neural Network and Scattered Data Approximations: The Inverse Manipulator Kinematics Example. *Neural Computation*, 1994, Vol. 6. pp. 521-542.

GOVINDARAJULU, Z. *The Sequential Statistical Analysis of Hypothesis testing, Point and Interval Estimation, and Decision Theory*. Columbus, Ohio: American Sciences Press Inc., 1987. 680 p.

GRABOT, Bernard. Fuzzy Knowledge Modeling in Production Management. In: *Publications of the Finnish Artificial Intelligence Society N° 3*. Helsinki: Kari Paatelainen Oy, 1990. pp. 46-53.

GRAILLOT, Didier, FRÄMLING, Kary, BUCHA, Jindrich. Waste Placement Decision Support System. *Actes du congrès HELECO '93, Vol. 2, 1-4 April 1993*. Athens: Technical Chamber of Greece, 1993. pp. 16-29.

GRAND, Jean-Michel, JULIOT, Bertrand. *Descriptor, manuel d'utilisation de la version Dos 2.2.* Vannes: Intelmark, 1990. 161 p.

GREBOVAL, Marie-Hélène. *La production d'explications, vue comme une tâche de conception: contribution au projet AIDE.* Thèse de Docteur: Université Technique de Compiègne, Contrôle des systèmes, France, 1994. 172 p.

GREFENSTETTE, J. Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 1985, Vol. 16, N° 1. pp. 122-128.

GREFENSTETTE, J. Multilevel Credit Assignment in a Genetic Learning System. *In: Proceedings of the Second International Conference on Genetic Algorithms and Their Applications.* Cambridge, MA: Lawrence Erlbaum Publishers, 1987. pp. 202-209.

HALL, Peter. *The Bootstrap and Edgeworth Expansion.* New-York: Springer-Verlag, 1992. 352 p.

HARDY, Rolland L. Multiquadric Equations of Topography and Other Irregular Surfaces. *Journal of Geophysical Research*, 1971, Vol. 76, N° 8. pp. 1905-1915.

HARMON, Paul. Precisely fuzzy - Part I. *Intelligent software strategies*, 1993, Vol. IX, N° 4. pp. 1-15.

HARMON, Paul. Precisely fuzzy - Part II. *Intelligent software strategies*, 1993, Vol. IX, N° 5. pp. 1-14.

HECHT-NIELSEN, Robert. Nearest matched filter classification of spatiotemporal patterns. *Applied Optics*, 1987, Vol. 26, N° 10, pp. 1892-1899.

HECHT-NIELSEN, Robert. *Neurocomputing.* Reading, Massachusetts: Addison-Wesley Publishing Company, 1990. 433 p.

HOLLAND, J.H. *Adaptation in Natural and Artificial Systems.* Ann Arbor, Michigan: University of Michigan Press, 1975. 183 p.

HOPFIELD, J. J. Neural Networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 1982, Vol. 79. pp. 2554-2558.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

IVAKHNENKO, A. G. (en russe) The group method of data handling - A rival of stochastic approximation. *Soviet Automatic Control*, 1968, Vol. 1. pp. 43-55.

JACQUET-LAGRÈZE, Eric. *Systèmes de décision et acteurs multiples. Contribution à une théorie de l'action pour les sciences des organisations.* Thèse d'état: Université Paris-Dauphine, 1981. 606 p.

JACQUET-LAGRÈZE, E. PREFCALC: évaluation et décision multicritère. *Revue de l'utilisateur de IBM-PC*, 1984, N° 3. pp 38-55.

JACQUET-LAGRÈZE, E., MEZIANI, R., SLOWINSKI, R. MOLP with an interactive assessment of a piecewise utility function. *European Journal of Operational Research*, 1987, Vol. 31, N° 3. pp 350-357.

JACQUET-LAGRÈZE, E., SISKOS, J. Assessing a set of additive utility functions for multicriteria decision-making; the UTA method. *European Journal of Operational Research*, 1982, Vol. 10. pp 151-164.

JASZKIEWICZ, Andrzej, SLOWINSKI, Roman. *The LBS package - a microcomputer implementation of the Light Beam Search method for multiple-objective non-linear mathematical programming - User's Manual.* Poznan: Technical University of Poznan, Institute of Computing Science, Poland, 1994. 18 p.

JOKINEN, Petri. *Continuously Learning Nonlinear Networks with Dynamic Capacity Allocation.* Thèse Doct.: Tampere University of Technology, 1991. 100 p.

KASANEN, E., ÖSTERMARK, R., ZELENY, M. Gestalt system of holistic graphics: new management support view of MCDM. *Computers and Operations Research*, 1991, Vol. 18, N° 2. pp. 233-239.

KASS, Robert. Building a User Model. *User Model and User Adapted Interaction*, 1991, Vol. 1, N° 3. pp. 203-258.

KASSEL, Gilles. *Contribution à la représentation des connaissances pour les systèmes experts de 2^{de} génération: Le projet AIDE.* Mémoire d'Habilitation: U.R.A. C.N.R.S. 817 Heudiasyc, Septembre 1995. 94 p.

KEENEY, Ralph L. Evaluation of Proposed Pumped Storage Sites. *Operations Research*, 1979, N° 27. pp. 48-64.

KEENEY, R.L., RAÏFFA, H. *Decision with Multiple Objectives: Preferences and Value Tradeoffs*. New-York: Wiley, 1976. 569 p.

KIRKPATRICK, S., GELATT, C.D. Jr., VECCHI, M.P. Optimization by simulated annealing. *Science*, 1983, N° 220. pp. 671-680.

KODRATOFF Yves, MICHALSKI Ryszard. *Machine Learning Volume III*. San Mateo California: Morgan Kaufmann Publishers, Inc., 1990. 825 p.

KOHONEN Teuvo. *Self-organization and associative memory*. Berlin: Springer-Verlag, 1984. 312 p.

KOLODNER, J.-L. Extending Problem Solver Capabilities Through Case-Based Inference. *Proceedings of the Fourth International Workshop on Machine Learning, University of California, Irvine, June 22-27, 1987*. Los Altos (CA): Morgan Kaufmann Publishers, Inc., 1987. pp. 167-178.

KORHONEN, P. Using harmonious houses for visual pairwise comparison of multiple criteria alternatives. *Decision Support Systems*, 1991, Vol. 17. pp. 47-54.

KORHONEN, P., LAAKSO, J. A visual interactive method for solving the multicriteria problem. *European Journal of Operational Research*, 1986, Vol. 24, N° 2. pp 277-287.

KOSKO, Bart. *Neural networks and fuzzy systems*. Englewood Cliffs: Prentice-Hall, 1992. 449 p.

LÉA SOMBÉ. *Raisonnements sur des informations incomplètes en Intelligence Artificielle*. Toulouse: Teknea, 1989. 221 p.

LEBART, L., MORINEAU, A., FÉNELON, J.-P. *Traitement des données statistiques*. Paris: Dunod, 1979. 510 p.

LEINONEN Lea, KANGAS Jari, JUVAS Anja. Pattern Recognition of Hoarse and Healthy Voices by the Self-Organizing Map. *ICANN-91 proceedings, Vol. 2*. Amsterdam: Elsevier, 1991. p. 1385-1388.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

LEVINE, P., POMEROL, J.-Ch. PRIAM, an interactive program for choosing among multiple attribute alternatives. *Journal of Operational Research*, 1986, Vol. 25. pp. 272-280.

LEVINE, P., POMEROL, J.-Ch. *Systèmes interactifs d'aide à la décision et systèmes experts*. Paris: Hermès, 1989. 335 p.

LORQUET, Vincent. *Etude d'un codage semi-distribué adaptatif pour les réseaux multicouches - Application au diagnostic, à la modélisation et à la commande*. Thèse Doct. Ing.: Ecole Nationale Supérieure des Télécommunications, Paris, 1992. 180 p.

LUKASIEWICZ, Jan. Philosophische Bemerkungen zu mehrwertigen Systemen des Aussagenkalküls. *Comptes-rendus des séances de la Société des Sciences et des Lettres de Varsovie, Classe III, 1930, Vol. 23.* pp. 51-77.

MANN, William C., THOMPSON, Sandra A. Rhetorical Structure Theory: Towards a functional theory of text organization. *TEXT*, 1988, Vol. 8, N° 3. pp. 243-281.

MARESCHAL, B., BRANS, J.-P. Geometrical representations for MCDA. *European Journal of Operational Research*, 1988, N° 33. pp. 69-77.

MAYSTRE, Lucien Yves, PICTET, Jacques, SIMOS, Jean. *Méthodes multicritères ELECTRE*. Lausanne: Presses polytechniques et universitaires romandes, 1994. 323 p.

McCULLOCH, W. S., PITTS, W. A Logical Calculus of the Ideas Imminent in Nervous Activity. *Bulletin of math. biophysics*, 1943, Vol. 5. pp. 115-133.

McKEOWN, Kathleen R. Discourse strategies for generating natural-language text. *Artificial Intelligence*, 1985, Vol. 27. pp. 1-41.

MEURS Elizabeth, PAYEN Catherine. Nous voulons une classe 1 *Décision environnement: supplément régional Rhône-Alpes*, 1992, N° 11. pp. 34-46.

MICHELLI, Charles A. Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions. *Constructive Approximation*, 1986, Vol. 2. pp. 11-22.

MICRO DATA BASE SYSTEMS Inc. *Guru v.2.0, manuels d'utilisateur*. Lafayette: Micro Data Base Systems Inc., 1990. 4 vol., pagination multiple.

MINSKY, M., PAPERT, S. *Perceptrons*. Cambridge, Massachusetts: MIT Press, 1969. 258 p.

MOORE, Johanna D, PARIS, Cécile L. Requirements for an expert system explanation facility. *Comput. Intell.*, 1991, Vol. 7. pp. 367-370.

NECHES, R., SWARTOUT, W. R., MOORE, J. D. Enhanced maintenance and explanation of expert systems through explicit models of their development. *IEEE Transactions on Software Engineering*, 1985, Vol. 11, N° 11. pp. 1337-1351.

NEURON DATA Inc. *Manuels Nexpert Object v.2.0*. Palo Alto, California: Neuron Data inc., 1991. 12 vol., pagination multiple.

OCDE. *SEDE-90 Proceedings of an NEA workshop: Heterogeneity of Groundwater Flow and Site Evaluation*. Paris: Lavoisier, 1991. 334 p.

ORR, Mark J. L. Regularization in the selection of Radial Basis Function Centers. *Neural Computation*, 1995, Vol. 7, N° 3. pp. 606-623.

PAELINCK, J. Qualiflex, a flexible multiple criteria method. *Economic letters*, 1978, N° 3. pp. 193-197.

PARIS, Cécile L. Tailoring object descriptions to the user's level of expertise. *Computational Linguistics*, 1988, Vol. 14, N° 3. pp. 64-78.

PAWLAK, Zdzislaw, SLOWINSKI, Roman. Decision Analysis Using Rough Sets. *International Transactions on Operational Research*, 1994, Vol. 1, N° 1. pp. 107-114.

PAWLAK, Zdzislaw, SLOWINSKI, Roman. Rough set approach to multi-attribute decision analysis. *European Journal of Operational Research*, 1994, Vol. 72. pp 443-459.

PERREARD, Sophie, SZAFNICKI, Konrad, BEAUNE, Philippe, GENOUD, Marc. Towards a computer-aided training tool for steelmaking applied to DC-electric arc furnaces. *Proceedings of the 8th IFAC Symposium on Automation in Mining, Mineral and Metal Processing, Sun City, 29-31 August, 1995*. Sun City: IFAC, 1995. pp. 102-108.

PLATT, John. A Resource-Allocating Network for Function Interpolation. *Neural Computation*, 1991, Vol. 3. pp. 213-225.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

POMEROL, Jean-Charles, BARBA-ROMERO, Sergio. *Choix multicritère dans l'entreprise.* Paris: Hermes, 1993. 391 p.

POWELL, M. J. D. Radial Basis Functions for Multivariable Interpolation: A Review. In: - *Proceedings of the IMA Conference « Algorithms for Approximation », July 1985.* Shrivvenham: The institute of mathematics and its applications conference series, 1987, N° 10, Second Edition. pp. 143-167.

PRADENC Henri. L'intelligence artificielle se renouvelle déjà. *Mesures*, 1991, N° 636, p. XV-XVII.

QUINLAN, J.R. Induction of decision trees. *Machine Learning*, 1986, N° 1. pp. 81-106.

QUINLAN, J.R. Decision Trees as Probabilistic Classifiers. *Proceedings of 4th International Workshop on Machine Learning.* Los Altos (CA): Morgan Kaufmann Publishers, 1987. pp. 31-37.

QUINLAN, J.R. Simplifying Decision Trees. *International Journal of Man-Machine Studies*, 1987, N° 27. pp. 221-234.

QUINLAN, J.R. Probabilistic Decision Trees. In: *Machine Learning: An Artificial Intelligence Approach.* Edited by Y. Kodratoff. San Mateo (CA): Morgan Kaufmann Publishers, 1990. pp. 140-142.

RAÏFFA Howard. *Decision Analysis.* Reading, Massachusetts: Addison-Wesley, 1968. 328 p.

RAÏFFA Howard. *Analyse de la décision.* Paris: Dunod, 1973. 328 p.

REY M., SCHMUTZ B., STAMPFLI E. *Recherche et définition de sites en Suisse romande.* Lausanne: C.E.A.T. (Communauté d'études pour l'aménagement du territoire), 1988. 81 p.

ROMMEL, Yves. *Apport de l'intelligence artificielle à l'aide à la décision multicritère.* Thèse Doct. Ing.: Université de Paris-Dauphine, 1989. 248 p.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 1958, Vol. 65. pp. 386-408.

ROY, B. Classement et choix en présence de points de vue multiples, la méthode ELECTRE. *Revue informatique et recherche opérationnelle*, 1968, Vol. 2, N° 8. pp. 57-75.

ROY, B. From optimization to multicriteria decision aid: three main operational attitudes. In: *Multiple Criteria Decision Making*. Edited by H. Thiriez & S. Zionts. Berlin: Springer-Verlag, 1976, N° 130. pp. 1-32.

ROY, B. ELECTRE III: un algorithme de classements fondé sur une représentation floue de préférences en présence de critères multiples. *Cahiers du CERO*, 1978, Vol. 20, N° 1., pp. 3-24.

ROY, B. *Méthodologie multicritère d'aide à la décision*. Paris: Economica, coll. « Gestion », 1985. 423 p.

ROY, B., BERTIER, P. *La méthode ELECTRE II: une méthode de classement en présence de critères multiples*. Paris: SEMA (Metra International), Direction scientifique, Note de travail N° 142, 1971. 25 p.

ROY, B., BERTIER, P. La méthode ELECTRE II: une application au media-planning. In: *Operational Research*. Edited by ROSS, M. Amsterdam: North-Holland Publishing Company, 1973. pp. 291-302.

ROY, B., BOUYSSOU, D. *Aide à la décision fondée sur une PAMC de type ELECTRE*. Paris: Université Paris-Dauphine, Document du LAMSADE, 1991, N° 69. 118 p.

ROY, B., BOUYSSOU, D. *Aide à la décision: méthodes et cas*. Paris: Economica, coll. « Gestion », 1993. 695 p.

ROY, B., HUGONNARD, J.-C. Classement des prolongements de lignes de métro en banlieue parisienne (présentation d'une méthode multicritère originale). *Cahiers du CERO*, Vol. 24, N° 2-3-4, 1982. pp. 153-171.

ROY, B., HUGONNARD, J.-C. Le plan d'extension du métro en banlieue parisienne, un cas type de l'analyse multicritère. *Les Cahiers scientifiques de la revue Transports*, 1982, N° 6. pp. 77-108.

ROY, B., SKALKA, J.M. *ELECTRE IS: aspects méthodologiques et guide d'utilisation*. Paris: Université Paris-Dauphine, Document du LAMSADE, 1985, N° 30. 125 p.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

ROYER Jean-Claude. *MANDRIN: Un système d'apprentissage pour l'aide au réglage d'un instrument.* Thèse Doct.: Institut National Polytechnique de Grenoble, 1990. 156 p.

RUMELHART, David E., McCLELLAND, James L. et al. *Parallel Distributed Processing Vol. 1.* Cambridge, Massachusetts: MIT Press, 1986. 547 p.

SAATY, Thomas L. *Decision Making for Leaders.* Belmont, California: Lifetime Learning Publications, 1981. 225 p.

SAATY, Thomas L. *Décider face à la complexité.* Paris: Entreprise moderne d'édition, 1984. 231 p.

SAFAR, B. (Prés.). *Actes des deuxièmes journées Explication du PRC-GDR-IA du CNRS, Sophia-Antipolis, 17-19 juin 1992.* Sophia-Antipolis: INRIA, 1992. 239 p.

SCHANK, Roger C., KASS, Alex, RIESBECK, Christopher (Ed.). *Inside Case-Based Explanation.* Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc., 1994. 416 p.

SCHLAIFER, Robert. *Probability and Statistics for Business Decisions.* New York: McGraw-Hill, 1959. 732 p.

SCHLAIFER, Robert. *Analysis of Decisions Under Uncertainty.* New York: McGraw-Hill, 1969. 729 p.

SCHÄRLIG, A. *Décider sur plusieurs critères. Panorama de l'aide à la décision multicritère.* Lausanne: Presses polytechniques et universitaires romandes, 1985. 304 p.

SEMEDDIRA. *Livre Blanc pour l'implantation de centres de stockage de déchets ultimes stabilisés en Rhône-Alpes.* Lyon: SEMEDDIRA, Juin 1993. 38 p.

SENOUNE, Redouane. *Acquisition de Connaissances et Apprentissage Automatique: contribution pour le développement incrémental d'un Système à Base de Connaissances pour les situations de crise - application au domaine de l'eau -.* Thèse de Docteur: Institut National des Sciences Appliquées de Lyon (École Nationale Supérieure des Mines de Saint-Etienne), France, 1995. 189 p.

SESTITO, S., DILLON, T. Automated knowledge acquisition of rules with continuously valued attributes. *Proceedings of the 12th International Conference on Expert Systems and their Applications (AVIGNON'92)*. Avignon, France, 1992. pp. 645-656.

SIMON, Herbert A. A behavioural model of rational choice. *Quarterly Journal of Economics*, 1955, Vol. 69. pp 99-118.

SIMON, Herbert A. *Administration et Processus de décision*. Paris: Economica, coll. « Gestion », 1983. 321 p.

SIMOS, JEAN. *Évaluer l'impact sur l'environnement. Une approche originale par l'analyse multicritère et la négociation*. Lausanne: Presses polytechniques et universitaires romandes, 1990. 261 p.

SLEEMAN, D. H. UMFE: A User Modelling Front End Subsystem. *International Journal of Man-Machine Studies*, 1985, Vol. 23. pp. 71-88.

SLOWINSKI, Roman. Rough Set Analysis of Multi-Attribute Decision Problems. *In: Rough Sets, Fuzzy Sets and Knowledge Discovery*. Edited by Wojciech P. Ziarko. London: Springer-Verlag, 1994. pp. 136-142.

SLOWINSKI, Roman, STEFANOWSKI, Jerzy. Rough Classification with Valued Closeness Relation. *In: New Approaches in Classification and Data Analysis*. Edited by E. Diday, Y. Lechevallier, M. Schader, P. Bertrand & B. Burtschy. Berlin: Springer-Verlag, 1994. pp. 482-489.

STEUER, R., CHOO, E. An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 1983, Vol. 26. pp 326-344.

SWARTOUT, William R., MOORE, Johanna D. Explanation in Second Generation Expert Systems. *In: Second Generation Expert Systems*. Edited by Jean-Marc David, Jean-Paul Krivine and Reid Simmons. New-York: Springer-Verlag, 1993. pp. 543-585.

TRESP, V., HOLLATZ, J., AHMAD, S. Network Structuring and Training Using Rule-based Knowledge. *Advances In Neural Information Processing*, 1993, Vol. 5. pp. 871-878.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

VANDERPOOTEN, D. The interactive approach in MCDA: a technical framework and some basic conceptions. *Mathematical Computing Modelling*, 1989, Vol. 12, N° 10/11. pp. 1213-1220.

VINCKE, Philippe. Une méthode interactive en programmation linéaire à plusieurs fonctions économiques. *Revue Française d'Informatique et de Recherche Operationnelle*, 1976, vol. 2. pp. 5-20.

VINCKE, Philippe. *L'aide multicritère à la décision*. Bruxelles: Éditions de l'Université de Bruxelles, 1989. 179 p.

WANG, J., MALAKOOTI, B. A feedforward neural network for multiple criteria decision making. *Computer and Operations Research*, 1992, Vol. 19. pp 151-167.

WEISS, Sholom M., KULIKOWSKI, Casimir A. *Computer Systems That Learn*. San Mateo: Morgan Kaufmann Publishers, Inc., 1991. 223 p.

WERBOS, Paul J. *Beyond regression: New tools for prediction and analysis in the behavioural sciences*. Thèse de Docteur: Harvard University, Mathématiques appliquées, États-unis, 1974. 223 p.

WICK, Michael R., THOMPSON, William B. Reconstructive expert system explanation. *Artificial Intelligence*, 1992, Vol. 54, N° 1-2. pp. 33-70.

WIDROW, B., HOFF, M. E. Adaptive switching circuits. *1960 IRE WESCON Convention Record*. New-York, 1960. pp. 69-104.

WIERZBICKI, A. The use of reference objectives in multiobjective optimization. In: *MCDM Theory and Application*. Edited by G. Fandel & T. Gal. Berlin: Springer-Verlag, 1980, N° 177. pp. 468-486.

WIERZBICKI, A. A mathematical basis for satisficing decision making. *Mathematical modelling*, 1982, Vol. 3. pp 391-405.

WRIGHT, Tommy. *Exact Confidence Bounds when Sampling from Small Finite Universes*. New-York: Springer-Verlag, 1991. 430 p.

WU, Dekai. Active Acquisition of User Models: Implications for Decision-Theoretic Dialog Planning and Plan Recognition. *Journal of User Model and User Adapted Interaction*, 1991, Vol. 1, N° 2. pp. 149-172.

YU, W. Aide multicritère à la décision dans le cadre de la problématique du tri: concepts, méthodes et applications. Thèse de doctorat, UER Sciences de l'organisation, Université de Paris-Dauphine, 1992. 201 p.

YU, W. *ELECTRE TRI. Aspects méthodologiques et manuel d'utilisation.* Paris: Université Paris-Dauphine, Document du LAMSADE, 1992, N° 74. 80 p.

ZADEH Lotfi A. Fuzzy Sets. *Information and control*, 1965, N° 8. p. 338-353.

ZADEH Lotfi A., FU King-Sun, TANAKA Kokichi, SHIMURA Masamichi. (Ed.). *Fuzzy Sets and Their Applications to Cognitive and Decision Processes.* New-York: Academic Press Inc., 1975. 496 p.

ZIONTS, S., WALLENIUS, J. An interactive programming method for solving the multiple criteria problem. *Management Science*, 1976, Vol. 22, N° 6. pp 652-663.

ZIONTS, S. A multiple criteria method for choosing among discrete alternatives. *European Journal of Operational Research*, 1981, Vol. 7. pp 143-147.

Annexe A

Présentation détaillée des applications utilisées

1. Introduction

Les applications utilisées comme exemple sont:

1. Le choix de voiture.
2. Le choix d'un site de stockage de déchets industriels ultimes dans la région Rhône-Alpes en France.
3. Le choix d'un site de stockage de déchets industriels ultimes en Suisse.

Les deux dernières sont pratiquement équivalentes. L'application Suisse a été utilisée à cause des problèmes de confidentialité des données pour l'application Rhône-Alpine. De l'autre côté, l'application Rhône-Alpine est plus intéressante et difficile car il y a plus de critères de sélection et d'actions potentielles.

L'application de choix d'automobile a été utilisée parce qu'elle reste suffisamment simple pour pouvoir définir des fonctions de préférence contextuelle réalistes. Il est également plus facile de juger si les résultats obtenus sont bons ou pas, puisque chacun est plus ou moins son propre décideur.

1.1 Choix de voiture

Le choix de voiture est une application qui est assez souvent utilisée comme exemple de problème de choix. Les raisons pour cette sélection sont multiples:

- Elle est bien connue par pratiquement tout le monde.
- Il y a un seul décideur (bien que ce ne soit pas toujours le cas ...).
- Le décideur est facilement accessible (soi-même, ou dans le cas décrit l'auteur de ce document).
- Il est facile d'obtenir des données exactes et objectives.
- Il est possible de comparer les classements obtenus par différentes méthodes sans trop d'effort.

L'avantage principal vient malgré tout de la facilité à définir des fonctions de préférence contextuelle, où les non-linéarités introduites correspondent à des règles naturelles et faciles à comprendre. Les données techniques, présentées dans l'Annexe D, viennent des deux revues d'automobile « L'auto-journal N° 14/15 1995 » et « Action Auto Moto, N° 2 1995 ». Ceci a permis d'identifier les critères de sélection réellement utilisés pour le choix de voiture, car les journaux du domaine savent quelles informations intéressent le plus les consommateurs.

1.2 Choix d'un site de stockage de déchets industriels ultimes dans la région Rhône-Alpes

Cette application a ses origines dans un projet traité suite à une convention entre l'École des Mines de Saint-Étienne et le bureau d'études géologiques CPGF-Horizon. Les initiateurs du projet sont les sociétés françaises SEMEDDIRA (Société d'Économie Mixte pour l'Étude de Décharges pour Déchets Industriels en Rhône-Alpes) et ADEME (Agence de l'Environnement et de la Maîtrise de l'Énergie).

La partie du projet confiée à l'École des Mines de Saint-Étienne était le développement d'un système d'aide à la décision pour le choix de sites de stockage de déchets industriels ultimes. Ce logiciel s'appelle StocDech. La collaboration proche au niveau de calculs nécessaires entre l'École des Mines de Saint-Étienne et la SEMEDDIRA permettait de bien comprendre les

problèmes liés à l'utilisation des SAD existants. Suite à cette collaboration, l'École des Mines de Saint-Étienne dispose des données réellement utilisées dans cette application.

La problématique, la démarche complète et les critères de sélection utilisés sont décrits dans le Livre Blanc de la SEMEDDIRA [SEMEDDIRA, 1993], qui est un document public. Par contre, les données complètes n'ont jamais été publiées. C'est pourquoi elles ne sont pas incluses, avec les données des autres applications, dans l'Annexe D.

1.3 Choix d'un site de stockage de déchets industriels ultimes en Suisse

Le logiciel StocDech développé dans le cadre du projet avec la SEMEDDIRA a d'abord été testé avec des secteurs et des critères de sélection identifiés en Suisse Romande [Rey et al., 1988]. Une méthode informatique similaire à StocDech avait déjà été utilisée en Suisse. Puisque les critères et les données Suisses correspondent assez bien aux critères et aux données de la région Rhône-Alpes, l'approche Suisse était utilisée comme une référence.

Les critères de sélection et les données utilisés viennent du rapport [Rey et al., 1988], qui est un document public. Il s'agit d'un problème plus limité que celui de Rhône-Alpes, puisqu'il n'y a que 13 critères de sélection et 65 actions potentielles.

2. Critères de sélection

Les critères de sélection utilisés dans les applications de choix de site ont été imposés par des décideurs complètement indépendants à ce travail de thèse. C'est également le cas des poids des critères. Les valeurs qualitatives et les intervalles de valeurs utilisés pour définir les fonctions d'utilité de StocDech ont été choisis en collaboration avec CPGF-Horizon. Uniquement le problème du choix de voiture a laissé une certaine liberté de choix.

2.1 Choix de voiture

Malgré le fait d'utiliser des critères de sélection bien acceptés par les revues d'automobile, les critères choisis ne sont pas optimaux pour un problème réel, car la plupart d'entre eux sont fortement corrélés. Le nombre de critères de sélection pourrait donc être largement réduit sans pratiquement rien perdre au niveau de la précision de décision. Il faudrait néanmoins le faire avec beaucoup d'attention, car la combinaison de deux critères fortement corrélés comme « prix » et « puissance maximale » donnerait des résultats erronés, puisque la valeur de préférence ne varie pas de la même façon en fonction des deux. D'autres critères presque inutiles sont « type de transmission » et « roues motrices », qui ont été inclus pour correspondre aux critères affrontés en réalité par le décideur.

<i>Critère</i>	<i>Poids</i>	<i>Valeurs qualitatives</i>	<i>Limites des classes</i>
1. Prix en kF français	12	Excessivement cher /Très cher /Cher /Abordable /Moyen /Assez moyen /Peu cher	250, 220, 190, 160, 130, 100
2. Puissance maximale en chevaux réels	11	Très faible /Faible /Assez moyenne /Moyenne /Bonne /Très bonne /Excellente	100, 120, 140, 160, 180, 200
3. Transmission - type	6	Non existant /auto /non existant /man.	
4. Roues motrices	5	Avant /Arrière /4x4	
5. Dimensions en m³-longueur*largeur*hauteur	6	Petite/Moyenne /Grande /Très grande	9, 10, 11.5
6. Coffre en dm³	6	Petite /Moyenne /Grande /Très grande	300, 350, 400
7. Poids en kg	4	Grand /Moyen /Petit	1200, 1000
8. Vitesse max. en km/h	10	Petite /Assez petite /Moyenne /Grande /Très grande /Excellente	180, 195, 210, 225, 235
9. 1000 m départ arrêté en secondes	10	Mauvaise /Assez mauvaise /Moyenne /Assez bonne /Bonne /Très bonne	33, 31.5, 30, 29, 28
10. Consommation moyenne (l/100km)	10	Très grande /Grande /Moyenne /Assez petite /Petite /Très petite	12.5, 12, 11, 10, 9
11. Esthétique	8	Moche /Assez moyen /Moyen /Beau /Très beau	
12. Image de marque	5	Mauvais /Moyen /Bon /Très bon	
13. Équipement en série	10	Presque rien /Limite /Moyen /Bon /Très bon /Excellent	35, 40, 50, 60, 70, 75

Tableau 2-1. Critères de sélection pour le choix de voiture.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

Les valeurs de tous les critères ont été obtenues directement à partir des revues, sauf les notes pour les critères « esthétique » et « image de marque ». L'obtention de la note pour le critère « équipement en série » est expliquée en détail dans l'Annexe D. Les poids, les valeurs qualitatives et les intervalles de valeurs (définis par les limites des classes du Tableau 2-1) sont des paramètres utilisés par StocDech.

2.2 Choix d'un site de stockage de déchets industriels ultimes dans la région Rhône-Alpes

Quinze critères de sélection étaient utilisés pour choisir le site. Tous les critères de sélection sont quantitatifs. Les critères de sélection sont de nature très variée et leurs valeurs sont très peu corrélées entre elles, comme indiqué par une analyse en composantes principales.

Les valeurs des critères sont pratiquement impossibles à comprendre sans leur définition exacte. Elles sont calculées par des formules assez complexes, compréhensibles uniquement par un expert initié au problème. Pour cette même raison les intervalles de valeurs utilisés par StocDech pour la transformation valeur quantitative \Rightarrow valeur qualitative \Rightarrow note sont assez abstraites pour le non-initié. Comme indiqué par ces intervalles, les fonctions d'utilité correspondantes sont loin d'être linéaires, ce qui complique l'apprentissage de la fonction de préférence.

Critère	Poids	Valeurs qualitatives	Limites des classes
1. Protection des captages d'alimentation en eau potable	19	Très mauvais /Très limite /Acceptable /Bon /Très bon	900, 2000, 10000, 150000
2. Caractéristiques du sous-sol	19	Très mauvais /Mauvais /Très moyen /Moyen /Assez bon /Bon /Très bon	20, 30, 40, 50, 60, 70
3. Protection des aquifères	19	Très mauvais /Mauvais /Moyen /Assez bon /Très bon	10, 40, 60, 80
4. Protection des zones agricoles labellisées ou d'Appellation d'Origine Contrôlée	5.8	Mauvais /Moyen /Bon	30, 55
5. Proximité des zones de production de déchets	5.8	Mauvais /Moyen /Bon	0.8, 1.5
6. Densité de la population	5.8	Très mauvais /Moyen /Bon /Très bon	8, 30, 55
7. Protection du patrimoine bâti	5.8	Mauvais /Moyen /Bon /Très bon	0.2, 0.3, 1.2
8. Protection des zones écologiques	5.8	Mauvais /Moyen /Bon	25, 40
9. Protection des zones touristiques	2	Mauvais /Moyen /Bon	5, 20
10. Protection des zones agricoles aménagées	2	Mauvais /Moyen /Bon	2, 20
11. Aménagement du territoire	2	Mauvais /Moyen /Bon	15, 21
12. Proximité des réseaux de transport	2	Mauvais /Moyen /Bon	2, 10
13. Nuisances liées au trafic	2	Mauvais /Moyen /Bon	10, 30
14. Protection du paysage	2	Mauvais /Moyen /Bon	2.2, 2.7
15. Protection des ressources forestières	2	Mauvais /Moyen /Bon	10, 60

Tableau 2-2. Critères de sélection pour le choix d'un site de stockage de déchets industriels ultimes dans la région Rhône-Alpes en France.

2.3 Choix d'un site de stockage de déchets industriels ultimes en Suisse

Treize critères de sélection de nature qualitative aussi bien que quantitative étaient utilisés pour choisir le site. Les critères de sélection sont peu corrélés entre eux, comme indiqué par une analyse en composantes principales.

<i>Critère</i>	<i>Poids</i>	<i>Valeurs qualitatives</i>	<i>Limites des classes</i>
1. Aptitude agricole du terrain	15.0	Excellente /Moyenne a bonne / Mauvaise	
2. Proximité d'une voie de chemin de fer	5.0	Aucun raccordement /À moins de 1200m /Présence d'une voie	
3. Proximité d'utilisateurs de marne	3.5	Plus de 10 km /Entre 5 et 10 km /Moins de 5 km	
4. Exposition visuelle	11.0	Très expose /Imp. à Moyenne /Réduite	
5. Appréciation de la géologie globale	12.0	GEOLEP 10 /GEOLEP 9 /GEOLEP 8 /GEOLEP 7 /GEOLEP 6 /GEOLEP 5	
6. Nombre de localités à traverser	12.5	Quatre ou plus /Trois /Deux /Une /Aucune	
7. Longueur de ruisseaux	4.5	Sup. à 10m/ha /Inf. à 10 m/ha /Aucun	
8. Tronçons à construire ou à élargir	4.0	À construire et(ou) élargir / À élargir /Aucun	
9. Proximité des fournisseurs de déchets spéciaux	4.5	Défavorable /Moyenne /Favorable	
10. Longueur du parcours en agglomération en mètres	10.0	Supérieure à 2 km /Comprise entre 1 et 2 km /Inférieure à 1 km /Nulle	2000, 1000, 1
11. Longueur du parcours sortie d'autoroute-secteur en km	5.0	Longue /Moyenne /Courte	9.1, 5.1
12. Densité des sources privées en nombre/ha	3.5	Grande /Petite /Aucune	0.04001, 0.00001
13. Valeur écologique	9.5	Elevée /Moyenne à élevée /Faible	

Tableau 2-3. Critères de sélection pour le choix d'un site de stockage de déchets industriels ultimes en Suisse.

3. Sélection des exemples d'apprentissage

La sélection des exemples d'apprentissage a été faite par la même méthode pour chaque application, les cartes auto-organisatrices de Kohonen. Pour le faire, les valeurs des critères étaient normalisées dans l'intervalle [0,1].

La carte auto-organisatrice est la même pour chaque application, c'est-à-dire une carte auto-organisatrice avec 10 fois 10 neurones organisés selon un voisinage en deux dimensions, taille et voisinage choisis pour obtenir une bonne répartition des actions potentielles. Les informations indiquées dans les illustrations des résultats de la carte auto-organisatrice sont:

- Le nombre d'actions potentielles dans chaque classe.
- L'ordre de sélection des exemples d'apprentissage.

Les classes des cartes auto-organisatrices sont numérotées du haut vers le bas et de gauche vers la droite. La classe du coin haut à gauche est ainsi la classe 1, la classe dans le coin gauche à bas la classe 10 et la classe dans le coin droite à bas la classe 100. Chaque carreau correspond à un neurone, qui correspond à une classe d'actions potentielles. Les chiffres dans les carreaux correspondent au nombre de secteurs appartenant à chaque classe. Les classes des exemples d'apprentissage sont indiquées par une flèche et leur ordre de sélection par le chiffre devant le nom.

3.1 Choix de voiture

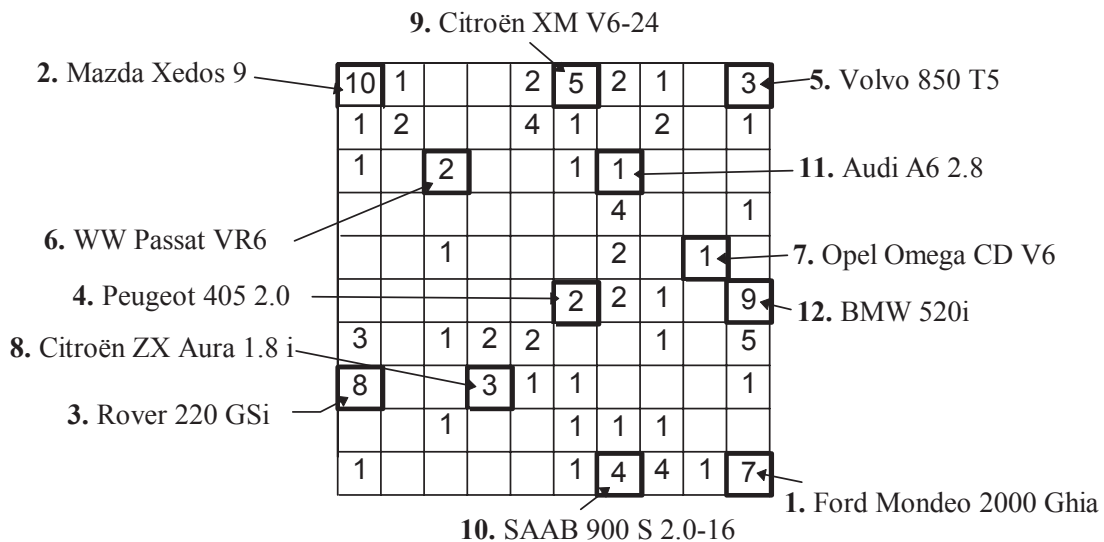


Figure 3-1. Résultats obtenus par la carte auto-organisatrice de Kohonen pour l'ensemble de voitures potentielles.

3.2 Choix d'un site de stockage de déchets industriels ultimes dans la région Rhône-Alpes

Une relation assez intéressante a été observée entre les classes de la carte auto-organisatrice et les résultats de la somme pondérée et de la méthode Electre I. Pour les actions potentielles de chaque classe de la carte auto-organisatrice, nous calculons la moyenne des valeurs de préférence obtenues par la méthode de somme pondérée. Ceci nous permet de produire une figure en trois dimensions, où les premières dimensions sont les deux dimensions de voisinage de la carte auto-organisatrice et la troisième celle de la valeur de préférence. La figure est une surface en trois dimensions avec des « montagnes » et des « vallées ». Les classes correspondant aux « montagnes » sont donc les classes contenant les secteurs les meilleurs.

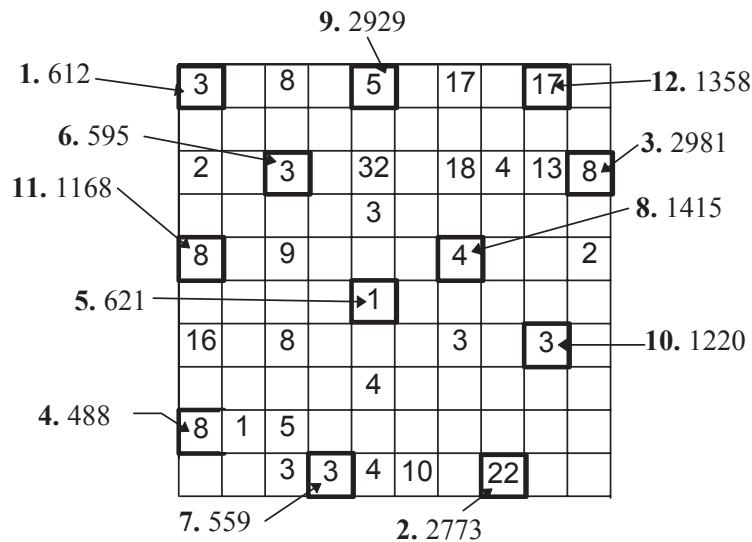


Figure 3-2. Résultats obtenus par la carte auto-organisatrice de Kohonen pour l'ensemble de secteurs potentiels pour le stockage de déchets industriels ultimes dans la région Rhône-Alpes.

La plupart des secteurs en tête du classement par somme pondérée se trouvent dans la classe qui correspond à la montagne la plus haute. Par contre, les secteurs retenus par la méthode Electre I se trouvent pas uniquement dans la classe qui correspond à la montagne la plus haute, mais également dans des classes ayant des montagnes moins hautes.

La différence entre la somme pondérée (et les méthodes additives en général) et la méthode Electre I (et d'autres méthodes du même type) est bien illustrée par ce phénomène. La notion d'incomparabilité d'actions des méthodes comparatives, comme Electre, permet de diminuer

les risques « d'oubli » de certains types d'actions à cause de poids ou d'autres paramètres mal choisis. Un changement dans les poids changerait la hauteur des montagnes, mais les actions retenues par une méthode Electre resteraient essentiellement les mêmes.

Le problème de « brillant second » des méthodes comme Electre peut s'expliquer aussi. L'action ayant la meilleure somme pondérée d'une classe réussit souvent à éliminer les autres actions de la même classe. Mais, une action éliminée de telle façon peut souvent être meilleure que toutes les actions appartenant à d'autres classes.

Cette relation entre la représentation en trois dimensions et les résultats des méthodes d'analyse multicritère fait que la carte auto-organisatrice même pourrait être utilisée comme une méthode d'analyse multicritère. Pour le faire, il faudrait d'abord calculer des valeurs de préférence par une méthode additive utilisant des poids donnés par le décideur ou des poids égaux. Le décideur pourrait étudier la surface ainsi obtenue et « tirer » certaines montagnes vers le haut ou « enfoncer » certaines vallées vers le bas. Un calcul permettrait à la suite de recalculer des poids correspondant aux changements. Les actions à retenir se trouverait à la fin aux sommets des montagnes les plus hautes. L'avantage avec une telle approche serait d'éviter les inconvénients des deux types de méthodes, les additives et les comparatives.

3.3 Choix d'un site de stockage de déchets industriels ultimes en Suisse

Pour les secteurs suisses, la carte auto-organisatrice était appliquée à un ensemble d'entraînement contenant non seulement les actions potentielles, mais également l'action idéale et anti-idéale. Les résultats de cette application et d'autres ont montré que cela ne pose pas d'avantages ni au niveau d'organisation de la carte, ni au niveau de l'interprétation des résultats. Dans un problème d'analyse multicritère classique ces deux actions sont tellement différentes ou éloignées (dans le sens vectoriel du terme) des actions potentielles qu'elles ne sont pas du tout comparables.

L'action idéale se trouve dans la classe 10 et l'action anti-idéale dans la classe 3 de la Figure 3-3. La classe la plus « proche » de l'action idéale dans le sens du voisinage est la classe 30, qui contient de très bons secteurs mais pas les meilleurs. La classe la meilleure est la classe 55. Il y a d'ailleurs plusieurs classes très bonnes dans la cinquième ligne, notamment les classes 25 et 95, où il y a néanmoins une seule action.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

Nous pouvons constater les mêmes relations entre la carte auto-organisatrice et les résultats de la somme pondérée et d'Electre I que pour les secteurs Rhône-Alpins.

Il y a deux secteurs retenus comme exemples d'apprentissage de la classe 30, ce qui n'est pas une sélection optimale. Le secteur 1222.12 a été choisi pour illustrer les explications. Ceci change peu les résultats obtenus.

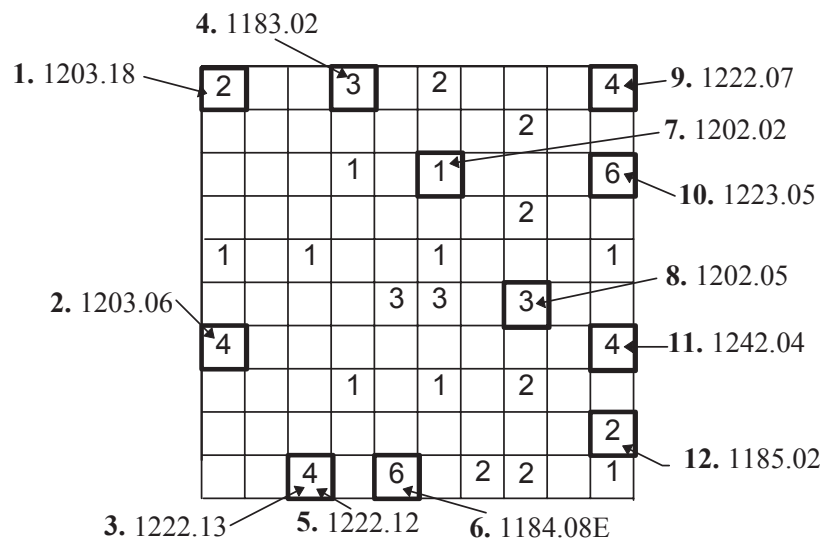


Figure 3-3. Résultats obtenus par la carte auto-organisatrice de Kohonen pour l'ensemble de secteurs potentiels pour le stockage de déchets industriels ultimes en Suisse.

4. Fonctions de préférence utilisées

Une fonction de préférence « modèle » est utilisée comme référence dans chaque application pour rendre possible une comparaison des résultats de différentes méthodes. La seule exception est la fonction de préférence obtenue par la méthode MHM pour le choix d'une voiture. Uniquement les valeurs de préférence des exemples d'apprentissage sont montrées ici. Les valeurs de préférence des autres actions potentielles se trouvent dans l'Annexe D.

4.1 Choix de voiture

La fonction de préférence modèle est une somme pondérée, modifiée par les trois règles suivantes, qui rendent cette fonction non-linéaire:

« SI le prix de la voiture > 200.000,- francs français
ALORS l'importance d'autres critères diminue pour atteindre
zéro à 300.000,- francs français »

« SI la consommation moyenne > 11L/100km ALORS l'importance
d'autres critères diminue pour atteindre zéro à 14L/100km »

« SI la puissance maximale dépasse 200ch alors qu'il n'y a
pas de traction 4x4 ALORS l'utilité de la puissance
maximale est réduite en proportion à la puissance »

Ces trois règles correspondent aux préférences d'un décideur imaginaire, qui a à sa disposition 200.000,- francs français, qu'il a prévu de mettre à l'achat d'une voiture. Même si une voiture est moins chère que ce montant, il faut qu'elle présente pratiquement les mêmes avantages qu'une voiture à 200.000,- francs pour être choisie. Si une voiture est plus chère que 200.000,- francs, par contre, notre décideur est obligé de faire des économies ailleurs ou d'emprunter de l'argent. Un prix plus élevé que 200.000,- francs rend ainsi vite peu intéressante n'importe quelle voiture. La deuxième règle concerne le même type de considérations économiques que la première.

La troisième règle est d'un autre type. Elle ne concerne pas uniquement les préférences du décideur, mais également les connaissances du domaine. Il est souvent possible de trouver des commentaires de ce type sur les problèmes de motricité des voitures à grande puissance.

Des critères d'exclusion ont également été utilisés dans le choix des automobiles à inclure dans l'ensemble d'actions potentielles. Seules les voitures à essence, à quatre portes et à un prix de plus de 100.000,- francs ont été retenues. Comme pour les règles, ce sont des critères qui sont motivés par la clientèle en question. Il est clair que les préférences et les priorités des clientèles différentes ne sont pas du tout les mêmes.

En plus des valeurs de préférence calculées par la somme pondérée modifiée, il y a également eu une saisie de poids par la méthode comparative de MHM, mais appliquée aux valeurs de préférence au lieu de poids. Les valeurs de préférence présentées dans le Tableau 7-1 sont obtenues par cette méthode à partir de comparaisons binaires avec 9 niveaux de précision, ce qui est le nombre de niveaux le plus couramment utilisé.

<i>Voiture</i>	<i>Note calculée</i>	<i>Note MHM</i>
Ford Mondeo 2000 Ghia	63.35	34.96
Mazda Xedos 9	64.65	38.88
Rover 220 GSi	60.30	26.79
Peugeot 405 2.0	62.67	32.62
Volvo 850 T5	76.40	73.64
VW Passat VR6	64.64	37.43
Opel Omega CD V6	69.15	56.48
Citroën ZX Aura 1.8 i	48.80	15.82
Citroën XM V6-24	4.93	10.01
SAAB 900 S 2.0-16	62.52	30.48
Audi A6 2.8	58.92	23.67
BMW 520i	68.88	48.20
Voiture idéale	103.00	100
Voiture anti-idéale	2.23	10

Tableau 4-1. Notes calculées par la fonction de préférence fictive et celles obtenues par la méthode MHM.

4.2 Choix d'un site de stockage de déchets industriels ultimes dans la région Rhône-Alpes

La nature de cette application rend impossible l'interaction avec les vrais décideurs du problème. Les poids des critères de sélection sont le résultat d'un processus d'acquisition de préférences auprès d'un grand nombre de décideurs. Les poids sont ainsi censés correspondre à une opinion consensuelle des décideurs.

La méthode de somme pondérée a été utilisée comme fonction de préférence « modèle ». Il est bien probable que les vraies préférences des décideurs sont contextuelles, mais il est impossible pour nous de les connaître. Par rapport à l'application choix d'automobile, il est

également assez difficile de construire de telles fonctions de préférence, car le domaine est plus complexe. Les valeurs de préférence obtenues pour les exemples d'apprentissage sont ainsi celles du Tableau 4-2.

<i>Secteur</i>	<i>Note somme pondérée</i>
612	69.8
2773	58.3
2981	62.2
488	62.2
621	68.0
595	78.1
559	54.8
1415	75.5
2929	65.1
1220	48.1
1168	64.0
1358	67.9
Idéal	100.0
Anti-idéal	23.7

Tableau 4-2. Valeurs de préférence calculées par somme pondérée pour les exemples d'apprentissage.

4.3 Choix d'un site de stockage de déchets industriels ultimes en Suisse

Cette application est pratiquement équivalente à la précédente. Le modèle de préférence utilisé dans ce cas est ainsi la méthode de somme pondérée. Les valeurs de préférence obtenues pour les exemples d'apprentissage sont celles du Tableau 4-3.

Secteur	Note somme pondérée
1203.18	62.00
1203.06	52.69
1222.13	63.98
1183.02	56.59
1222.12	63.71
1184.08E	58.10
1202.02	69.93
1202.05	53.90
1222.07	44.61
1223.05	61.25
1242.04	36.82
1185.02	61.31
Idéal	100.00
Anti-idéal	0.00

Tableau 4-3. Valeurs de préférence calculées par la somme pondérée pour les exemples d'apprentissage.

5. Pseudo-exemples utilisés

Les pseudo-exemples étaient choisis à l'aide de l'analyse graphique de la fonction de préférence apprise. La procédure suivie était toujours la même: la fonction était corrigée pour un critère de sélection après l'autre, jusqu'à ce qu'un niveau « satisfaisant » soit atteint. Ce niveau jugé satisfaisant correspond à une situation où il n'y a plus d'erreurs évidentes ou quand les erreurs évidentes sont jugées sans importance par le décideur. Pratiquement tous les pseudo-exemples sont construits par un remplacement de la valeur d'un critère de sélection par la valeur minimale ou maximale du critère pour un exemple d'apprentissage.

Les pseudo-exemples ainsi obtenus sont indiqués par une matrice d'exemple d'apprentissage versus critère, avec une indication des valeurs de remplacement. La lettre « m » indique que la valeur de remplacement est la valeur minimale du critère et la lettre « M » que c'est la valeur maximale du critère.

5.1 Choix de voiture

Les pseudo-exemples utilisés pour le choix de voiture sont indiqués dans le Tableau 5-1. Le nombre total de pseudo-exemples est 48, sans compter les actions idéale et anti-idéale. En étudiant la matrice, il devient clair que les pseudo-exemples sont le plus souvent introduits pour les cas extrêmes, qui, dans ce cas, sont la Volvo 850 T5 et la Citroën ZX Aura 1.8 i.

Voiture	1	2	3	4	5	6	7	8	9	10	11	12	13
Ford Mondeo 2000 Ghia													
Mazda Xedos 9					m			M	M				M, m
Rover 220 GSi	M									M			
Peugeot 405 2.0		M									M		
Volvo 850 T5	M, m	m			m	M, m	M, m	m	m	m	m		m
WW Passat VR6													M, m
Opel Omega CD V6	m												
Citroën ZX Aura 1.8 i	M	M						M, m	M		M		M
Citroën XM V6-24		m			m			m	m		m		m
SAAB 900 S 2.0-16										M			M, m
Audi A6 2.8						m	m						
BMW 520i	M	M											

Tableau 5-1. Matrice de pseudo-exemples pour le choix de voiture. Les deux pseudo-exemples construits à partir de la Volvo 850 T5 pour les prix de 150000 et de 200000 et le pseudo-exemple construit à partir de la Citroën XM V6-24 pour le prix de 200000 ne sont pas indiqués dans la matrice.

Certains critères demandent plus de pseudo-exemples que les autres. Il est facile de comprendre pourquoi le critère « prix » demande le plus de pseudo-exemples, puisque c'est le critère pour lequel la fonction de préférence est la moins linéaire. Le critère « équipement » demande beaucoup de pseudo-exemples lui aussi, ce qui est certainement dû à sa corrélation forte avec le prix qui nécessite d'annuler les effets de la non-linéarité du prix. En ce qui concerne les critères « puissance maximale », « accélération » et « vitesse maximale » ils sont moins corrélés avec le prix, tout en étant très corrélés entre eux. Le besoin de pseudo-exemples est pour cette raison moindre. Pour les critères « transmission » et « roues

motrices » il n’y a pas eu besoin de pseudo-exemples du tout. Ce sont des critères avec des poids et une puissance de séparation assez faibles due au petit nombre de valeurs possibles.

5.2 Choix d’un site de stockage de déchets industriels ultimes dans la région Rhône-Alpes

Secteur	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
612		M						m							
2773	M	M		m				m							
2981		M		M											
488		m		M, m			M							m	
621		m													
595		M				M	M		M	M		M		m	
559		m						m	M						
1415				M	m		M	M		M		M		m	M
2929	M	M			M	M									
1220					m			M	M	M		M	M	m	M
1168				m	m			m							
1358	m							m							

Tableau 5-2. Matrice de pseudo-exemples pour le choix d’un site de stockage de déchets industriels ultimes dans la région Rhône-Alpes.

Les pseudo-exemples utilisés pour le choix d’un site de stockage de déchets industriels ultimes sont indiqués dans le Tableau 5-2. Le nombre total de pseudo-exemples est 49, sans compter les actions idéale et anti-idéale. Comme les pseudo-exemples du choix de voiture, il y a des critères qui demandent plus de pseudo-exemples que d’autres. Il y a certains critères pour lesquels il n’était pas jugé utile de fournir de pseudo-exemples du tout. La raison est soit que les exemples d’apprentissage d’origine suffisaient, soit que les pseudo-exemples d’autres critères corrigeaient déjà la fonction de préférence.

5.3 Choix d'un site de stockage de déchets industriels ultimes en Suisse

Dans le cas de secteurs Suisses, tous les pseudo-exemples sont construits à partir du secteur 1222.12 car cette application était utilisée uniquement dans la partie sur les explications. La fonction de préférence devait pour cette raison être très exacte pour le secteur expliqué (le 1222.12), mais la précision de la fonction était beaucoup moins importante pour les autres secteurs. Il y avait donc 26 pseudo-exemples en tout, qui étaient construits à partir du secteur 1222.12 par le remplacement de la valeur de ce secteur par la valeur minimale et maximale possible un critère de sélection à la fois.

Annexe B

The Interpolating, Normalising and Kernel Allocating Network for Fast Function Approximation

ette annexe présente une étude approfondie du réseau neuronal INKA, dont le nom vient de *Interpolating, Normalising and Kernel Allocating*. Le texte est en anglais, car il vient d'un article proposé au journal *Neural Computation* en Juin 1994 avec le titre *A Dynamically Interpolating and Capacity Evaluating Network for Fast Function Approximation*, d'où venait l'abréviation DICE utilisée précédemment. L'étude des réseaux de neurones à base de fonctions radiales (*radial basis functions, RBF*) est plus approfondie ici que dans la thèse même. Les comparaisons avec les réseaux RBF classiques et le réseau de rétro-propagation permettent également de voir les avantages de INKA.

Puisqu'il s'agit à l'origine d'un article indépendant, il y a une partie « références bibliographiques » propre à cette annexe, qui a forcément certains éléments en commun avec les références bibliographiques de la thèse. Cette solution a été retenue pour garder un texte le plus indépendant que possible de la thèse.

The Interpolating, Normalising and Kernel Allocating Network for Fast Function Approximation

Kary Främling

Department for Computer Science, Mathematics, Decision Theory and Environment, Ecole Nationale Supérieure des Mines de Saint-Etienne, 158 cours Fauriel, 42023 Saint-Etienne Cedex 2, FRANCE

This paper describes the Interpolating, Normalising and Kernel Allocating (INKA) neural network that is based on the principles of radial-basis-function (RBF) networks. Modifications to the basic RBF network concern the training algorithm, the use of a non-Gaussian output function for the hidden neurons and the use of normalized inputs to the neurons of the output layer instead of absolute ones. Several tests on known functions show that these modifications give INKA learning performances which are superior to those of both multilayer backpropagation networks and usual RBF networks. These improvements concern learning speed, sensibility to local minima, precision, definition of training parameters and generalization capabilities.

1. Introduction

This paper presents a neural network (NN) that is based on the principles of the so-called *radial-basis-function* (RBF) networks. This NN approximates functions by interpolation between the peaks of the radial basis functions instead of trying to make them fit directly to the approximated function, as is usually done in RBF networks. This is the main reason why the network is called the *Interpolating*, Normalising and Kernel Allocating (INKA) network. *Kernel Allocating* means that the number of hidden neurons is adjusted in order to improve the function approximation.

Throughout this paper we will use some definitions and notations that are not yet well standardized and therefore a common source of confusion. The term *activation function* refers to the function that combines the input values of a neuron to produce the activation value of

the neuron (usually a weighted sum). The activation value is the argument for the *output function*, which gives the output value of the neuron. Linear, sigmoid and Gaussian functions are common output functions. Scalar variables are indicated in small letters (x), vectors as small letters with an arrow above (\vec{x}) and matrices in capital letters (X).

Neural networks with RBF output functions are often classified as *localized neural networks* because each hidden neuron is active i.e. has an output greater than zero, only in a very limited subspace of the whole input vector space. On the other hand, we have the *global neural networks*, where a hidden neuron may be active over the whole of the input vector space, often with great activation overlap with the other hidden neurons. The *multilayer feedforward network* using *learning by backpropagation* (BP) [Rumelhart et al., 1988] may be the best example of this kind of a neural network.

Localized neural networks are mainly used in classification tasks, but work has also been done on continuous function approximation [Klopfenstein & Sverdløve, 1983; Hartman et al., 1990; Park & Sandberg, 1991, 1993; Benaim, 1994]. Since one RBF neuron usually covers only a limited input subspace, there are often big « holes » which are not covered at all unless the shape of the RBF functions is perfectly tuned to the approximated function. In particular, multidimensional input vector spaces can easily require an impractically large number of neurons. However, the local nature of the neurons means that RBF networks can learn relatively fast [Moody & Darken, 1989; Debenham & Garth, 1991; Platt, 1991]. Even one-shot learning algorithms exist for RBF networks [Jokinen, 1991].

BP networks have been successfully applied to both function approximation and classification problems [Rumelhart et al., 1988]. As the hidden neurons have a global output function, there are usually no « holes » in the input vector space covered by the BP network. Unfortunately, the global nature of the BP network makes learning very slow. This is especially true for high-dimensional input vector spaces.

These advantages and inconveniences have been verified by the tests performed in section 5 with both RBF and BP NNs. The INKA network presented in this article combines the best aspects of the RBF network (fast and localized learning) with the best aspects of the BP network (continuous and global function approximation).

RBF networks are studied in section 2. Section 3 presents the modifications to the RBF network that improve the function approximation capabilities. Section 4 outlines the training

and optimization algorithms developed. The different NNs are tested on three different functions in section 5, followed by the conclusion in section 6.

2. RBF networks and their function approximation capabilities

The RBF network model consists of three layers as shown in Figure 4. The first layer (hereafter called the *input layer*) only distributes the input values to the neurons of the second layer (called the *hidden layer*). The neurons of the hidden layer are the *RBF neurons* (sometimes also called *kernel nodes*). Every neuron i has a centroid \bar{w}_i that is defined by its weights. The number of weights is the same as the number of inputs since no bias is used. For an input vector $\bar{x} = (x_1, x_2, \dots, x_n)$, the hidden neuron computes its activation value as the squared *Euclidean distance* between \bar{x} and \bar{w}_i as

$$r_i^2 = (\bar{x} - \bar{w}_i)(\bar{x} - \bar{w}_i)^T \quad (17)$$

where r_i^2 is the activation value of hidden neuron i and \bar{x} and \bar{w}_i are the $1 \times n$ input and weight row vectors. T means matrix transposing of $(\bar{x} - \bar{w}_i)$. Other ways of calculating the distance between \bar{x} and \bar{w}_i exist that take into account the distribution of the input data, which is usually done by using the *Mahalanobis distance* instead of the Euclidean distance [Jokinen, 1991; Carlin, 1992; Roberts & Tarassenko, 1994].

This distance is then passed to the output function of the neuron, which is usually a Gaussian function, but other functions have been proposed and used. A comparison of several output functions has been carried out in [Carlin, 1992]. The output function is usually radially symmetric around zero, but this is not always the case [Roberts & Tarassenko, 1994].

A common form of a Gaussian function for calculating the output of a hidden neuron is

$$h_i = e^{-r_i^2 / d_i^2} \quad (18)$$

where h_i is the output of hidden neuron i , r_i^2 is the activation of neuron i and d_i^2 is a positive scalar constant. The notation d_i^2 indicates that each hidden neuron has its own parameter value while d^2 is used to indicate that all hidden neurons of the network have the same parameter value.

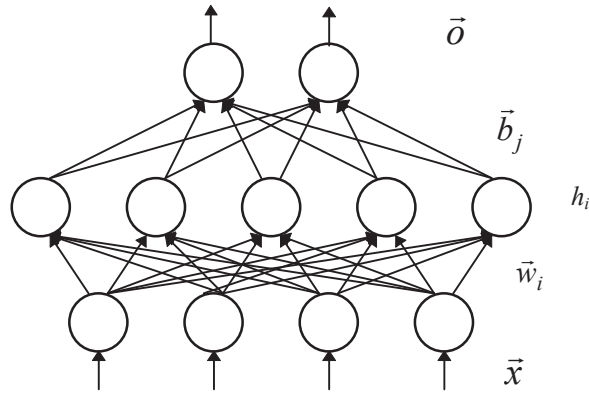


Figure 4. A radial-basis-function network. \bar{x} is the input vector, \bar{w}_i is the weight vector of hidden neuron i , h_i is the output of hidden neuron i , \bar{b}_j is the weight vector of output neuron j and \bar{o} is the output vector.

The output function used in this paper is the *Inverse MultiQuadric Equations* (IMQE) function [Hardy, 1971, 1982; Michelli, 1986; Powell, 1987; Gorinevsky & Connolly, 1994]

$$h_i = \left(1 + \frac{r_i^2}{d_i^2}\right)^{-1/2} \quad (19)$$

The neurons in the *output layer* have an activation function which is the weighted sum of their inputs. They use a linear output function. Their output is thus calculated as

$$o_j = \sum_{i=1}^k b_{j,i} h_i \quad (20)$$

where o_j is the value of network output j , h_i is the output of hidden neuron i , $b_{j,i}$ is the weight between output neuron j and hidden neuron i and k is the number of hidden neurons. A bias input may also be used in order to modify the default output value, that would otherwise be zero.

The function approximation properties of an RBF network are very dependent on the value of the parameter d^2 . Determination of the optimal value is mainly a case of trial-and-error,

Chapitre 7. Perspectives: Des explications pour les décisions négociées

becoming more and more difficult as the number of inputs increases. In the case of the IMQE function, the smaller the value of d^2 , the narrower the output function. Choosing the best value for d^2 is complicated by the fact that the distance between the centroids is not always the same and that the shape of the approximated function is not uniform. This means that d^2 has to be a compromise that may be optimized only up to a certain limit.

Several learning algorithms have been proposed for RBF networks. Some of these algorithms require that the number of hidden neurons be defined before learning, while others create hidden neurons during learning.

The learning algorithms with a predefined number of hidden neurons usually use some kind of clustering technique such as k-means clustering or competitive learning to distribute the neurons in the input vector space [Kohonen, 1984; Moody & Darken, 1988, 1989]. Once this phase is finished, the hidden neuron output function is changed into an RBF (unlike the *counter-propagation* network [Hecht-Nielsen, 1990]) and the weights of the output layer can be calculated as a matrix inversion, by gradient descent or other methods.

Learning by creation of new neurons follows a different principle. A neuron is usually created wherever the difference between the network output and the desired output exceeds a user-defined limit. To avoid over-populating some area of the input space due to excessive noise or other reasons, there is usually a constant c , which expresses a circular « influence area » of the hidden neurons. The centroid of one hidden neuron should never be inside the influence area of another neuron. The DCA (*Dynamic Capacity Allocation*) network presented in [Jokinen, 1991] is created by a training algorithm that uses these principles. This learning algorithm is very fast since it is enough to make only one pass through the training set. Unfortunately the results are extremely sensitive to the initial choice of d^2 and c . Jokinen's second algorithm with an adaptive c does not solve the problem. The number of hidden neurons can also easily get very large.

Reducing the number of hidden neurons is one of the goals of some other hidden neuron-creating learning algorithms, such as the *Resource-Allocating Network* (RAN) presented in [Platt, 1991] and the *Coarse Coding RAN* (CC-RAN) presented in [Deco & Ebmeyer, 1993]. The RAN algorithm reduces the value of c during the training, thus slowly reducing the possible distance between hidden neuron centroids. Every neuron also has its individual value for the d_i^2 parameter, which is determined by its distance to the other centroids. Both

networks use gradient descent for optimizing both the output and hidden layer weights. They both manage to significantly reduce the number of hidden neurons required compared to some other learning algorithms.

Automatically adjusting the value of the d_i^2 parameter has also been done by [Debenham & Garth, 1991] and [Roberts & Tarassenko, 1994], but they were mainly working on classification tasks. The use of Gaussian bar units instead of RBF units reported in [Hartman & Keeler, 1991] produces quite a similar behaviour to CC-RAN.

3. Modification of the RBF network to improve its interpolation capabilities

One of the biggest challenges with RBF networks is to « fill in » the holes in the input vector space that are not covered by any hidden neuron. So far nearly the only way of doing this has been adjusting the factor d^2 . The approach taken for the INKA network is to find a way of interpolating the function to approximate between the centroids of the hidden neurons.

The first modification done to the basic RBF network is to use a relative activation function in the output layer instead of using the direct weighted sum. This means that the input values h_i of the output neurons are divided by the sum of all inputs of the output neuron. The result is the normalized value $hn_{j,i}$

$$hn_{j,i} = h_i / \sum_{x=1}^m h_x \quad (21)$$

where $hn_{j,i}$ is the normalized output value of hidden neuron i for output neuron j , h_i is the output value of hidden neuron i and m is the number of hidden neurons connected to output neuron j . For the rest of the paper we will assume that all hidden neurons are connected to all output neurons, which means that $hn_{j,i}$ is the same for all j . Then the output value of output neuron j is calculated according to (6), where h_i is replaced by hn_i .

Using normalized output values for the hidden neurons has been done before [Nowlan, 1990]. These so called *Normalized Gaussian Basis Functions* (NGBF) networks were shown to be superior to normal Gaussian units for supervised classification tasks. A proof that they are

capable of universal approximation was presented in [Benaim, 1994]. They were also shown to have improved generalization properties compared to classical RBF networks, but Benaim had not found any explanation to this phenomena. Better generalization properties also usually reduce the number of hidden neurons required.

Figure 5 shows what the final network output function looks like for different hidden layer output function shapes for the case of one input, one output and two hidden neurons. The network tries to learn the two points $(2,2)$ and $(6,5)$. This means that the hidden neuron centroids will most certainly be $w_1 = 2$ and $w_2 = 6$ by using any of the currently known training algorithms for RBF networks. Then the weights of the output neuron may directly be initialized to $b_{1,1} = 2$ and $b_{1,2} = 5$.

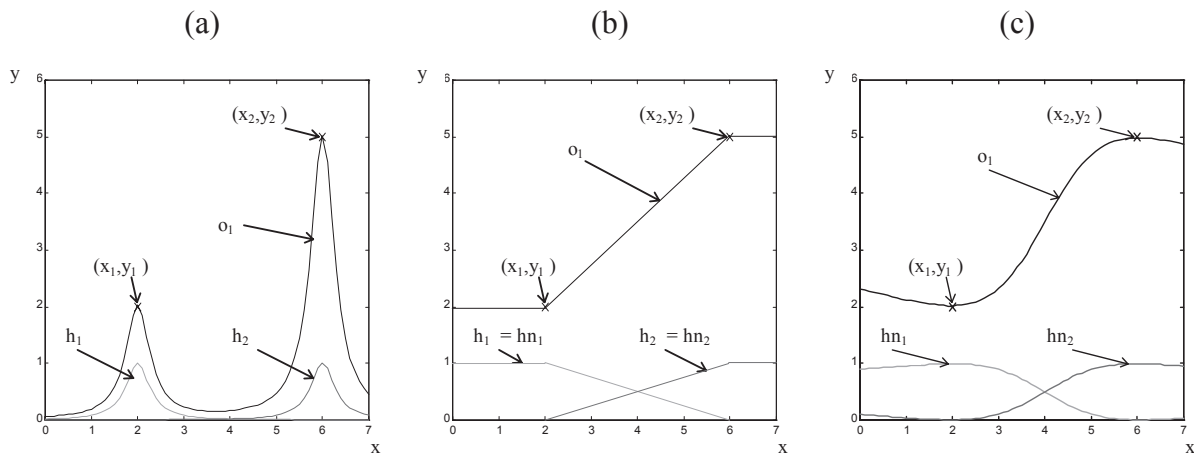


Figure 5. Network output with (a) non-normalized hidden layer outputs, (b) hidden layer outputs producing perfect linear interpolation, (c) normalized hidden layer outputs.

In the case of classical RBFs, a final network output is shown in Figure 5 (a). Since the d^2 parameter does not happen to be well tuned for the current centroid distance, we get a big « hole » in the middle. If we modify the shape of the hidden layer output functions to those shown in Figure 5 (b), we obtain a perfect linear interpolation, which is an improvement. As shown in Figure 5 (c), normalized hidden neuron output values give nearly the same shape to the hidden layer output functions as in Figure 5 (b), which results in the final non-linear interpolation shown. The normalization can be said to produce a phenomena of « dynamic width », where the RBF width changes depending on the distance and direction of the closest centroids.

One reason for using the IMQE output function instead of a Gaussian output function is that it corresponds better to the linear interpolation principle. It is also simpler and therefore faster to calculate. Furthermore, better results are reported with the IMQE than with the Gaussian output function for RBF networks in [Gorinevsky and Connolly, 1994]. This conclusion seems to be verified by the tests performed and the difference seems to be even more significant when using normalized hidden layer output values. This is the case for both approximation accuracy and sensibility to the value of the d^2 parameter.

The IMQE function is in fact less local than the Gaussian function because it goes to zero much slower when r_i^2 goes towards infinity. This means that we never obtain the ideal condition for linear interpolation indicated in Figure 5 (b) since no neuron ever has an output of exactly zero, not even when \bar{x} is exactly at the centroid of some other hidden neuron. Therefore it is usually not possible to directly use the correct output value found in the learning examples as the weight $b_{j,i}$ between output neuron j and hidden neuron i . The weight vectors \bar{b}_j are calculated by solving the matrix inversion problem

$$Y = HN * B \quad (22)$$

where Y is the matrix with the correct output values for the training set, HN is the matrix of normalized outputs of the hidden neurons for the training set and B is the output layer weight matrix to be found. This inverse matrix always exists unless H is singular, which happens if the IMQE function underflows or two centroids are identical [Powell, 1987]. The selected value of the d^2 parameter should be at least big enough to avoid this problem.

It is interesting to study the relationship between using a bias input in the output layer and using normalized hidden layer outputs. The bias weight should normally be close to the mean output value of the training examples, which is then always the default output. When using normalized hidden layer outputs we obtain the same result when the distance between \bar{x} and the closest \bar{w}_i goes towards infinity. If, however, r_i^2 stays within reasonable limits, we obtain a much more local mean value that corresponds to a kind of « moving bias ».

An approach to further localize the default output value is to limit the number of hidden neurons whose output h_i is greater than zero. This means that only the N biggest values h_i are allowed to be greater than zero, which gives greater locality since the hidden neurons with the

N biggest output values should still have the N closest centroids. In practice, this solution has not given good results.

Another idea would be to use other kinds of output functions for the hidden neurons that would correspond even more to the ones in Figure 5 (b). One idea that seems interesting would be to use so-called *multi-sigmoid unit networks* [Tomasini, 1993]. The multi-sigmoid unit networks are like the competitive layer of a feature map in the sense that they represent a Voronoï tessellation of the input vector space, but the outputs change continuously instead of being one for the winning neuron and zero for the others.

4. The learning algorithm

All NN learning algorithms have the same goal - minimizing a global error measure of some kind. For function approximation it is often the *mean square error* (MSE) or the *root mean square error* (RMSE) between the correct output and the output of the NN for the whole training or test sets.

Algorithm 1: Training

1. Choose the values for the d^2 and c parameters.
2. Start off with an empty hidden layer, which means the network output is always zero.
3. Initialize the set of resting examples R to the whole training set.
4. Select the training example in R for which the output error is the greatest, add it to the set of used examples U and remove it from the set of resting examples R .
5. If there is no hidden neuron closer than c to the example, allocate a new hidden neuron i with \bar{w}_i equal to \bar{x} .
6. Recalculate the B matrix for the examples in U .
7. Test if the global error measure is small enough. If not, go back to step 4.

The main principle of the training algorithm is to allocate new hidden neurons where the output error is the greatest, thus trying to reduce the global error as much as possible at each step. However, the distance between the centroids of two hidden neurons is not allowed to be smaller than the constant c . The algorithm presented (algorithm 1) is valid for one output only, but it is also applicable for multiple outputs.

The weight matrix B is only calculated for the examples in U instead of calculating it for all examples. This is to avoid excessive « smoothing » at an early stage of the learning when the number of hidden neurons is still small. This « smoothing » tends to lower the peaks in the NN output, which means that the hidden neurons tend to get too concentrated at the peaks. Recalculating B on the whole training set is therefore done only at the end of the training, which normally reduces the global error quite significantly. By modifying the error test in step 7 it is also possible to do incremental learning. It is, however, preferable to have an initial training set as big as possible in order to allocate as few hidden neurons as possible.

This training algorithm usually leads to local minima that are detected during training. It seems that local minima are reached once peaks of a certain amplitude have been « filled in » and before starting to fill in the peaks of the next smaller amplitude until all noise is included. It is therefore often useful to study the result at the local minima to determine the right moment to stop the training. In fact, with good values for the d^2 parameter, training could always be continued to nearly zero error on the training set, which is reached when there is one neuron per example. This is a definite advantage of this training algorithm compared to, for example, backpropagation, where local minima usually stop further learning.

Only the parameters d^2 and c need to be defined by the user in advance for training the RBF and INKA networks. In the examples used in this paper only the d^2 parameter is changed while c has been zero. The default value of zero for the c parameter is sufficient to avoid two neurons having exactly the same centroids. However, experiments have shown that it is sometimes useful to give c a non-zero value. This is the case when the training data contains a lot of noise or when the function to learn is very irregular, for example with long flat areas and abrupt valleys. This kind of training data may provoke an allocation of neurons with centroids that are sometimes very far from each other and sometimes extremely close. In this case it becomes quite difficult to choose the d^2 parameter correctly in order to get good approximation results on the whole curve.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

Some redundant neurons are nearly always generated during the training. A simple optimization algorithm (algorithm 2) may therefore be executed at the end of the training to remove them. The fact that this algorithm is slow is not a great problem since it is done on a NN that is already operational. Another optimization approach could be to modify the centroids of the hidden neurons.

The training algorithm tends to first allocate neurons for *significant training examples* (the extreme output values) [Hardy, 1971]. Combined with the optimization procedure, it is possible to be sure of not allocating much more neurons than absolutely necessary. The training algorithm is quite similar to *stepwise forward selection*, while the optimization algorithm resembles the *backward elimination* procedure discussed for instance in [Boyce et al., 1974]. The *optimal regression algorithm* presented in the same book guarantees an optimal selection of examples, but it quickly becomes too slow when the number of examples increases.

Algorithm 2: Optimization

1. Temporarily remove one hidden neuron at a time, recalculate the B matrix on the whole training set and register the resulting global error for each case.
2. Identify the neuron removal that caused the smallest error increase (the global error may even decrease with the removal of some neurons).
3. If the maximum limit for the global error is not violated by the removal of the identified neuron, the neuron is removed permanently (this means that neurons for which the global error decreases are always removed).
4. Go back to step one until no more neurons can be removed without increasing the global error too much.

Optimization of multilayer BP networks by removal of weights based on the use of second-order derivatives (or approximations of them), has been proposed in [Le Cun et al., 1990]. The algorithm is called « Optimal Brain Damage » or ODB. The INKA algorithm is, however, quite different from ODB. ODB works by removing weights in a NN with global behaviour. The optimization algorithm proposed here is for removing neurons (with all their corresponding weights) that have a local role in INKA. This algorithm is probably more

biologically plausible, since large numbers of neurons die in the brain every day, while losing connections between neurons is a phenomena that is less known.

5. Comparison of learning speed and complexity

The three tests of this chapter are function approximation tests rather than classification tests. Classification problems can, however, often be considered to be a special kind of function approximation. Therefore a NN that performs function approximation well may also be expected to be quite good in most classification tasks.

A test function with two inputs and one output is used for comparing the learning speed of INKA with that of the BP network, while a function with one input and one output is used for comparing various aspects of INKA with RBF. The reason for using « synthetic » functions is that it permits focusing on the essential points to be demonstrated, whilst remaining in a low-dimensional context, which permits a graphical representation of the function and the learning results. Other comparisons between performances of RBF and BP networks may be found in [Lee, 1991] on handwritten digit recognition.

The RBF network used is a classical RBF network, where the output of the output neurons is directly the un-biased weighted sum of the hidden neuron outputs. The training and optimization algorithms presented in section 4 have been used for allocating neurons.

The BP network used is the one found in the Matlab Neural Network Toolbox [Demuth & Beale, 1993]. BP learning was done using a momentum term as well as using an adaptive learning rate. Nguyen-Widrow initial conditions were used, which means that the input values have been normalized between $[-1,1]$.

Only one hidden layer was used for the BP network. The number of hidden neurons and values of other training parameters were decided based on the results of several test runs and previous experience. The output functions were a sigmoid function for the hidden layer and the direct weighted sum for the output layer. The sigmoid function used is:

$$h_i = \frac{2}{1 + e^{-2S_i}} - 1 \quad (23)$$

where h_i is the output of hidden neuron i and S_i is the activation value of hidden neuron i .

All the networks were implemented using Matlab v. 4.0 in its Microsoft Windows version. The computer used had an Intel 486 DX processor at 33 Mhz and 10 Mbytes of RAM. All calculation times are given for this computer configuration. This is not the optimal environment for NN simulations and the training times obtained are therefore quite long. However, this does not prevent them from being perfectly comparable.

5.1 Comparison of learning speed

The function used for comparing the learning speed of the different NNs is the « sombrero » function. This is a difficult function for all the NNs used because a large number of hidden neurons is required in order to approximate it. The function is:

$$f(x,y) = \frac{\sin(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}} \quad (24)$$

The training set consists of 300 randomly generated (x,y) pairs from the range $[-10,10]$. The test set consists of equally spaced (x,y) pairs from the range $[-10,10]$ generated using a step value of 0.7. These values are also used for the plots in Figure 6. The training set RMSE goal was 0.02.

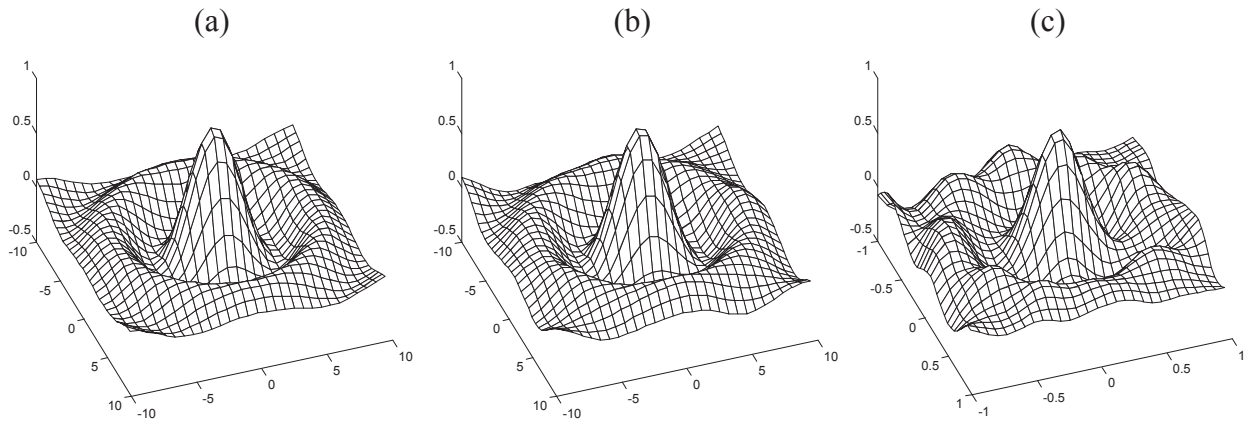


Figure 6. Three-dimensional plot of (a) the sombrero function, (b) the training result after optimization for the RBF network using $d^2 = 30$, (c) the training result after BP training with adaptive learning rate.

Several runs were performed with both BP methods in order to use parameter values that are as optimal as possible. Fifty hidden neurons were used since this amount should be sufficient for the approximated function. The number of hidden neurons was also selected close to that created by the INKA learning algorithm in order to simplify comparisons. For learning with a momentum, a learning rate of 0.0001 and a momentum of 0.95 were used. For learning with an adaptive learning rate the initial learning rate was set to 0.001.

The optimal values for d^2 were found without using the optimization phase, which significantly speeds up the procedure. Optimization is then done on a network that already performs well.

Network type	#Hidden	Training/Optimization time	Test set RMSE
BP with momentum	50	5 h. 45 min. (10000 epochs)	0.0970
BP with adaptive learning rate	50	5 h. 45 min. (10000 epochs)	0.0652
Ordinary RBF, $d^2 = 30$	39 / 36	50 s. / 15 min.	0.0234
INKA, $d^2 = 20$	45 / 38	50 s. / 25 min.	0.0227

Table 1. Comparison of learning results for multilayer network backpropagation, RBF and INKA. The number of hidden neurons after both the training and the optimization phases are indicated for the RBF and INKA networks.

The time gain compared to the BP network is quite clear. The error goal was never reached with either of the training methods. Judging by the RMSE tendency, the goal might have been reached after 20000 or 30000 epochs.

Both the INKA and RBF networks learn the sombrero function very well and using quite a small number of hidden neurons. As expected, the hidden neuron centroids formed circles in the 2-dimensional input plane that corresponded to the hills and valleys of the sombrero. However, when the constant five was added to the $f(x,y)$ values of the sombrero, the number of hidden neurons allocated was doubled for RBF, while it remained stable for INKA. This shows the sensitivity of RBF networks to output values that are not well centered around zero. As mentioned before, using a bias for the output layer neurons would help in this case.

5.2 Study of the effect of the d^2 parameter on the RBF and INKA networks

For the sombrero function it was possible to approximate the function using less hidden neurons for RBF than for INKA. However, this was the case only for a very optimal value of the d^2 parameter, that was found after several learning runs. Unfortunately, the results obtained with RBF are extremely sensitive to the value of this parameter. The minimal number of hidden neurons generated by RBF and INKA also depends on the shape of the approximated function.

In order to demonstrate this, we try to approximate the sigmoid function (23) with the exponent $(-2S_i)$ replaced with $(-10S_i)$ in order to make the step steeper. The training set consists of 200 randomly generated x values from the range $[-10,10]$. The test set consists of equally spaced x values from the same range generated using a step value of 0.7. The training set RMSE goal was 0.05. The results for minimal number of hidden neurons after optimization and minimal RMSE obtained are indicated in Table 2.

Network	Minimal test set RMSE				Minimal # hidden neurons			
	d^2	T/O #Hidd.	RMSE	T/O time (s)	d^2	T/O #Hidd.	RMSE	T/O time (s)
RBF	3	22 / 14	0.0465	11 / 155	5	35 / 12	0.0529	13 / 700
INKA	0.1	17 / 11	0.0467	24 / 65	0.5	15 / 10	0.0519	17 / 44

Table 2. Results for the goals of minimal test set RMSE and minimal number of hidden neurons. T/O means training/optimization.

In this case INKA needs less hidden neurons to reach the same approximation result. This is especially true for the training phase, where RBF allocates a lot more hidden neurons, requiring long optimization times. It also means that it is quite difficult to determine the optimal d^2 -value for RBF, since it is not the same for training and optimization.

The sensibility of RBF to changes in the d^2 parameter is clearly shown by the training results presented in Table 3 and the RMSE results plotted in Figure 7. In the case of one input it is still quite easy to find a good value by trial-and-error, but with more inputs it becomes more difficult.

d ²	RBF		INKA	
	# Hidd.	RMSE	# Hidd.	RMSE
10	51	0.0148	25	0.0436
1	19	0.0373	15	0.0330
0.1	45	0.0534	17	0.0381
0.01	102	0.246	25	0.0399
0.001	160	0.544	25	0.0436

Table 3. Number of hidden neurons created and test set RMSE as a function of the value of d^2 for RBF and INKA (without optimization). For $d^2 = 100$ the H matrix became singular and the training failed.

For the INKA network it seems that choosing a value for d^2 that is too small is not really a problem, but just means that more neurons than necessary are allocated by the training algorithm. In this case the optimization phase usually removes many neurons, which means that the final result is still quite good. No matter how much d^2 was decreased, there were always 25 hidden neurons after the training phase and the resulting RMSE did not change significantly. Therefore finding the absolutely optimal value for the d^2 parameter is not necessary for INKA.

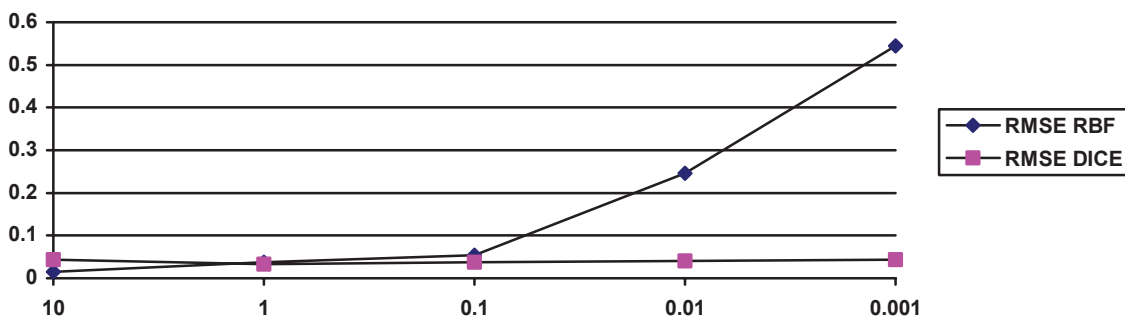


Figure 7. Comparison of test set RMSE sensibility to the value of d^2 for RBF and INKA.

It is also interesting to study the distribution of the centroids obtained by the two networks. For INKA the biggest concentration of centroids is always around the step, whilst it decreases the further we go along the flat part. For RBF, the bigger the value of d^2 , the less centroids there are on the flat part and the more around the step. The smaller the value of d^2 , the more

centroids there are on the flat part and the less around the step. This difference is clearly due to the « moving bias » effect obtained by the normalized h_i values for INKA.

5.3 Test function with 13 input variables - modeling of decider preference functions

The background for the development of INKA is to try to learn decider preference functions for various kinds of selection problems. A classical example is that of selecting a car, which will also be used in this paper, although the real problems treated with INKA come from the environmental domain. These applications are, however, too complicated and too long to describe in this context.

Thirteen selection criteria are used for determining the best car. The idea is to learn and generalize the decider preference function from a limited number of examples, which is in this case twelve. These twelve examples have been selected so that they cover the whole alternative space as well as possible by first classifying them using a feature (Kohonen) map. The test set consists of 113 cars, from which the selected car is to be identified. The small number of examples makes this application quite different from most NN applications and it also makes it quite difficult to find an accurate approximation of the preference function. Fortunately, it is not necessary to learn this function with a very high precision, since the main goal is usually only to put the cars in the correct preference order.

It is difficult to use true decider preference functions because of the great subjectivity involved. A performance measure like « decider contentment » is not very precise or easy to express numerically. Therefore the preference function used for producing the training and test data is defined as a weighted sum with some non-linearities introduced by the use of three rules, which gives « preference scores » from the range [0,100]. It is then possible to use standard error measurements such as RMSE. More details on the preference function, the criteria and the data are found in appendix 1.

In this case training time was no problem for the BP network. It was always about 4 minutes (using an adaptive learning rate, one hidden layer), compared to the 5 seconds for RBF and INKA. However, the generalization capabilities obtained were very poor, independently of the number of hidden neurons used, as can be seen from Figure 8. It is absolutely clear that a BP network can not be used for this kind of application.

INKA performs a little bit better than RBF even for optimal values of d^2 , as can be seen from Figure 8. The optimal d^2 values are quite high due to the mainly linear nature of the preference function. As usually, RBF is very sensitive to the value of this parameter, which is a big problem for this kind of application. Preference functions may take potentially any shape and be locally very linear or non-linear. The decider is usually not at all familiar with the underlying learning algorithm, which necessitates automatic procedures and a big error tolerance. It is therefore difficult to determine the optimal d^2 values, especially since the small number of examples available makes it difficult to validate the results. This is a typical case of the « small-sample » problem also known from statistics and machine learning [Weiss & Kulikowski, 1991]. Determining the number of examples required is, in fact, a domain-dependent problem whose difficulty varies greatly.

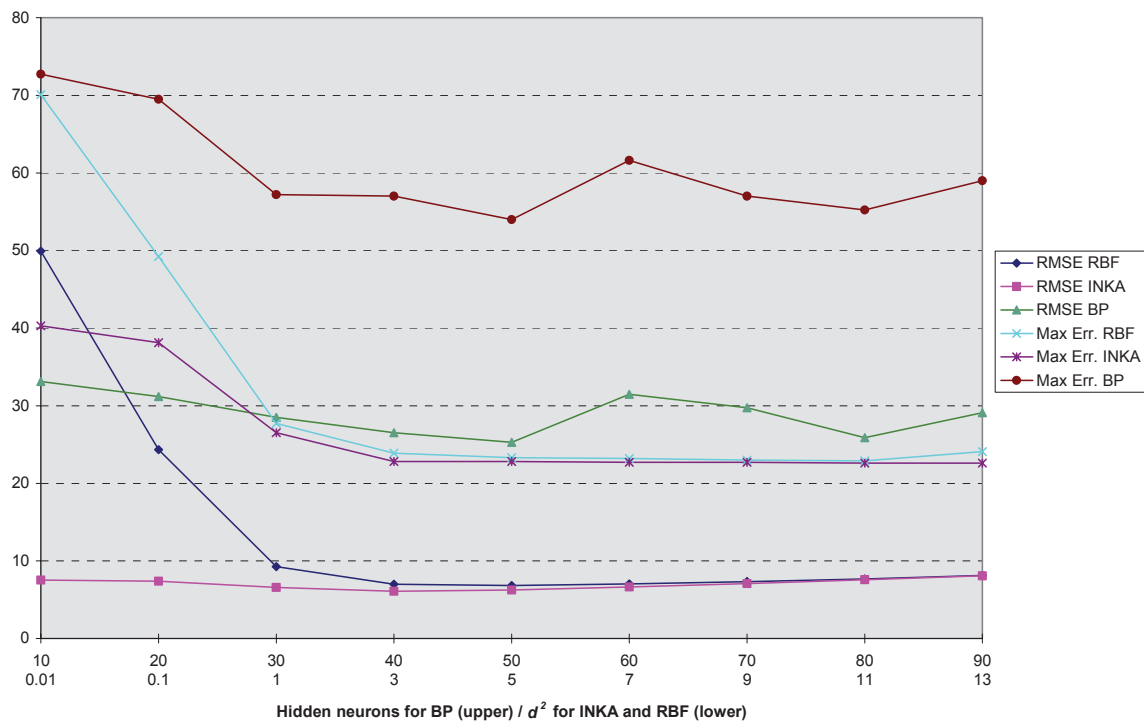


Figure 8. Comparison of training results for the three networks. Test set RMSE and maximum error are indicated for d^2 values of 0.01 to 13 for RBF and INKA, while they are indicated as a function of the number of hidden neurons for backpropagation (10 to 90).

For this application the number of hidden neurons created is naturally quite close to the number of examples. There are still some differences between RBF and INKA. The number of hidden neurons is 11 for INKA and 12 for RBF with optimal test set RMSE performance. RBF allocated 11 hidden neurons for d^2 values of 0.01 or less, but the test set RMSE is

hopelessly high. INKA only used 10 hidden neurons for $d^2 = 1$ and 9 hidden neurons for $d^2 = 0.1$ and $d^2 = 0.01$, losing hardly any generalization performance.

6. Conclusion

The results presented clearly show that BP is slow compared to the INKA training algorithm. BP networks often tend to spend a lot of time either getting over local minima or just getting stuck by them, while INKA does not have any problems of this kind. This is not compensated by better final function approximation, nor by a smaller number of hidden neurons (or weights) required. From these results it is clear that BP is not a good choice for the type of problems presented in this paper.

The advantages of INKA compared with RBF are mainly a better coverage of the input space and much less sensitivity to the value of the d^2 parameter. With less than absolutely optimal d^2 , less hidden neurons are needed for INKA than for RBF due to its interpolation capabilities. This is a very important advantage especially for applications where determining the optimal value is difficult because of the approximated function, small training sets or other factors. Therefore INKA may be used for applications where the use of NNs is still difficult, as for the modeling of decider preference functions.

Future research will mainly consider the automatic modification of the output function shapes of the hidden neurons either by locally modifying the d_i^2 parameter or by other solutions. The optimization procedure could certainly also be further improved, for instance by modifying the existing centroids instead of just removing hidden neurons.

Appendix 1. Details on the car application

The thirteen selection criteria used are: price (french francs), maximum power (hp), type of transmission (automatic/manual, scores 2/4), traction (front-/rear-/four-wheel driven, scores 1/2/3), car size (m^3), trunk size (liters), weight (kg), top speed (km/h), acceleration 1000m with zero start speed (s), fuel consumption (l/100km), esthetic score (1, 2, 3, 4 or 5), fabricant's image (1, 2, 3, 4 or 5) and score for standard equipment (continuous from 0 to 83).

Chapitre 7. Perspectives: Des explications pour les décisions négociées

These criteria are quite heterogeneous due to the presence of both continuous and discrete values, where the discrete values correspond to symbolic concepts, such as « good », « bad », « yellow » or « green ». This means that it is difficult to make any far going assumptions about the properties of the preference function (linearity, monotonicity, ...).

It is obvious that several of these criteria are highly correlated and would normally be combined before treatment by NN methods. However, one particularity of decider preference functions is that even for two seemingly highly correlated criteria, one (or several) particular values may strongly change the decider's preferences. An exaggerated example of this kind of a situation is when the decider absolutely does not accept a car that consumes more than 12 liters per 100 kms, whilst he does not care about the price of the car, he will give such a car a very bad score. If we combine the price and the consumption, it is impossible to take this aspect into account.

Car	FF	Hp	Tr	T	m ³	l	kg	km	s	l	Es	Im	Eq	Sc.
Audi A6 2.8	229300	174	4	1	12.2	510	1445	219	30	12.3	4	3	55	59
BMW 520i	189900	150	4	2	11.6	460	1445	211	30.9	10.6	4	4	44	69
Citroen ZX Aura 1.8 i	109900	103	4	1	9.7	343	1008	188	35.3	10.1	2	2	30	49
Citroen XM V6-24	298100	200	4	1	11.7	540	1565	235	29	12.6	4	2	70	5
Ford Mondeo 2000 Gh.	141000	136	4	1	10.7	470	1287	208	31.2	10	4	2	57	63
Mazda Xedos 9	205900	143	4	1	11.9	420	1410	202	31.4	10.3	4	2	62	65
Opel Omega CD V6	194000	170	4	2	12.4	530	1510	223	31	11	4	3	68	69
Peugeot 405 2.0	133700	123	4	1	10.4	470	1135	197	32.8	9.3	3	3	43	63
Rover 220 GSi	119500	136	4	1	9.9	350	1160	202	31	9.4	3	2	42	60
SAAB 900 S 2.0-16	147800	130	4	1	11.4	451	1285	200	31.8	10.2	4	4	50	63
Volvo 850 T5	239000	225	4	1	11.6	445	1415	240	27.7	11.7	5	4	71	76
WW Passat VR6	197800	174	4	1	11.3	495	1365	224	29.2	11.6	4	2	55	65

Table 4. Data matrix for the training examples of the car application.

The training data used are shown in Table 4. The data was normalized into the interval [0,1] by making the minimal value of each criteria correspond to 0 and the maximal value to 1. The limited number of examples is quite typical for this kind of problems, where finding examples

easily requires a big effort from the decider. Since the examples are usually obtained through pairwise comparisons, the effort rapidly grows with the number of examples.

The scores (column Sc. in the table) have been calculated by first transforming the original data into scores using a criteria-dependent discrete utility function, then calculating the weighted sum for each car. The weights correspond to the importance given to each criteria by the decision maker. The scores obtained with the weighted sum are then further modified according to the following rules:

- IF the price of the car $> 200.000,-$ FF THEN the importance of all other criteria exponentially decreases and is zero for $300.000,-$ FF.
- IF the fuel consumption $> 11L/100km$ THEN the importance of all other criteria exponentially decreases and is zero for $14L/100km$.
- IF there is over 200hp but not a four wheel drive THEN exponentially reduce the utility of the maximum power so that it is 30% for 240hp.

These three rules modify the preference function in a moderate way, but much enough to render the use of linear models very inefficient.

References

- Benaim, M. 1994. On Functional Approximation with Normalized Gaussian Units. *Neural Comp.* **6**, 319-333.
- Boyce, D. E., Farhi, A., Weischedel, R. 1974. *Optimal Subset Selection - Multiple Regression, Interdependence and Optimal Network Algorithms*. Springer-Verlag, Berlin.
- Carlin, M. 1992. Radial Basis Function Networks and Nonlinear Data Modelling. In *Neuro-Nimes-92 proceedings*, pp. 623-633. EC2, Nanterre.
- Debenham, R. M., Garth S. C. J. 1991. The Detection of Eyes Using Radial Basis Functions. In *Artificial Neural Networks*, T. Kohonen, K. Mäkisara, and J. Kangas, eds., Vol. 1, pp. 81-86. Elsevier, Amsterdam.
- Deco, G., Ebmeyer, J. 1993. Coarse Coding Resource-Allocating Network. *Neural Comp.* **5**, 105-114.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

Demuth, H., Beale, M 1993. *Neural Network Toolbox for Use With Matlab User's Guide*. MathWorks Inc., Massachusetts.

Gorinevsky, D., Connolly, T.H. 1994. Comparison of Some Neural Network and Scattered Data Approximations: The Inverse Manipulator Kinematics Example. *Neural Comp.* **6**, 521-542.

Hardy, R. L. 1971. Multiquadric Equations of Topography and Other Irregular Surfaces. *Journal of Geophysical Research* **71**, 1905-1915.

Hardy, R. L. 1982. Surface Fitting with Biharmonic and Harmonic Models. In *Proceedings of the NASA workshop on Surface Fitting*, pp. 136-146. Center for approximation theory, Texas A&M University.

Hartman, E., Keeler, J.D., Kowalski, J. 1990. Layered Neural Networks with Gaussian hidden units are universal approximators. *Neural Comp.* **2**, 210-215.

Hartman, E., Keeler, J.D. 1991. Predicting the Future: Advantages of Semilogical Units. *Neural Comp.* **3**, 566-578.

Hecht-Nielsen, R. 1990. *Neurocomputing*. Addison-Wesley Publishing Company, Massachusetts.

Jokinen, P. 1991. *Continuously Learning Nonlinear Networks with Dynamic Capacity Allocation*. PhD thesis, Tampere University of Technology.

Klopfenstein, R. W., Sverdlow, R. 1983. Approximation by Uniformly Spaced Gaussian Functions. *Approximation Theory IV*, 575-580.

Kohonen, T. 1984. *Self-organization and associative memory*. Springer-Verlag, Berlin.

Le Cun, Y., Denker, J., Solla, S. 1990. Optimal Brain Damage. In *Proceedings of IEEE Conference on Neural Information Processing Systems*, 598-605.

Lee, Y. 1991. Handwritten Digit Recognition Using K Nearest-Neighbor, Radial-Basis Function, and Backpropagation Neural Networks. *Neural Comp.* **3**, 440-449.

Michelli, C. A. 1986. Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions. *Constructive Approximation* **2**, 11-22.

Moody, J., Darken, C. 1988. Learning with Localized Receptive Fields. In *Proceedings of the 1988 Connections Models Summer School*, Touretzky et al., eds., pp. 133-143. Morgan-Kaufman Publishers.

Moody, J., Darken, C. 1989. Fast learning in networks of locally-tuned processing units. *Neural Comp.* **1**, 281-294.

Nowlan, S. 1990. Maximum likelihood competitive learning. In *Proceedings of Neural Information Processing Systems*, 574-582.

Park, J., Sandberg, I. W. 1991. Universal approximation using radial basis function networks. *Neural comp.* **3**, 246-257.

Park, J., Sandberg, I. W. 1993. Approximation and Radial-Basis-Function Networks. *Neural Comp.* **5**, 305-316.

Platt, J. A. 1991. Resource-Allocating Network for Function Interpolation. *Neural Comp.* **3**, 213-225.

Powell, M. J. D. 1987. Radial Basis Functions for Multivariable Interpolation: A Review. In *Algorithms for Approximation*, Mason and Cox, eds., pp. 143-167. Clarendon Press, Oxford.

Roberts, S., Tarassenko, L. A. 1994. Probabilistic Resource Allocating Network for Novelty Detection. *Neural Comp.* **6**, 270-284.

Rumelhart, D. E., McClelland, J. L. et al. 1988. *Parallel Distributed Processing Vol. 1*. MIT Press, Massachusetts.

Tomasini, L. 1993. *Apprentissage d'une représentation statistique et topologique d'un environnement*. PhD thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Paris.

Weiss, S.M., Kulikowski, C.A. 1991. *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems*. Morgan Kaufmann Publishers.

Annexe C

Présentation de l'outil StocDech

Le développement de l'outil StocDech a débuté par une étude confiée à la Compagnie de Prospection géologique française HORIZON (CPGF Horizon) par L'ADEME (Agence de l'Environnement et de la Maîtrise de l'Énergie). L'objet de l'étude était la mise au point d'un outil informatique d'aide à la décision pour la recherche de sites de stockage de déchets. Pour conduire cette étude, CPGF Horizon a établi une convention avec le laboratoire « Ingénierie de l'Environnement » de l'École des Mines de Saint-Étienne par l'intermédiaire de la société Armines. StocDech a donc été développé à l'École des Mines de Saint-Étienne dans le cadre de ce travail de thèse.

1. Utilisation du logiciel

StocDech est développé sur des compatibles IBM/PC, sous l'interface utilisateur Microsoft Windows. Cette interface permet l'utilisation de plusieurs logiciels en parallèle et une communication homme-machine à l'aide de composants multiples: boutons, menus, listes, barres de défilement etc., dont un certain nombre sera illustré ici¹³.

La fenêtre montrée dans la Figure 1-1 est la fenêtre principale du logiciel. Cette fenêtre présente une série de boutons qui permettent d'accéder aux différentes fonctions du système. La chouette qui se trouve dans la fenêtre principale symbolise la sagesse attribuée à la

¹³ Toutes les copies d'écran montrées utilisent les critères et les secteurs d'une étude effectuée en Suisse par le C.E.A.T. [Rey et al., 1988], introduits à titre d'exemples.

chouette dans la vieille mythologie grecque. Ceci reflète notre espérance pour que cet outil permette de prendre des décisions meilleures qu'auparavant.



Figure 1-1. Fenêtre principale en mode « utilisateur ».

Le contenu de cette fenêtre dépend en partie du mode d'utilisation. Le mode « utilisateur » permet seulement d'accéder aux données des actions et aux résultats des différentes méthodes. La définition des critères de sélection et la mise à jour des calculs sont accessibles uniquement en mode « administrateur ».

1.1 Définition des critères

La première étape dans un projet de décision multicritère est la définition des critères de sélection. Chaque critère est défini dans la fenêtre montrée dans la Figure 1-2. Un critère est défini par les éléments suivants:

- Un titre qui est utilisé dans la fenêtre de saisie de données (Figure 1-3), dans les explications etc.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

- Un code base de données, qui est unique pour chaque critère, qui sert d'identification du critère dans les bases de données. Ce nom ne doit contenir que des caractères alphanumériques dont le nombre doit rester inférieur ou égal à 10.
- Une définition ou explication, qui est affichée en sélectionnant le nom du critère dans la fenêtre de saisie de données. Cette explication peut être soit un texte soit un fichier. Si l'explication est un texte simple, il est affiché dans une boîte de dialogue simple.

Figure 1-2. La fenêtre «Définition d'un critère».

- La nature (type) du critère: numérique, symbolique ou déduit.
- L'unité pour les critères numériques.
- Le poids exprimant l'importance du critère.
- Une liste de valeurs symboliques possibles qui est introduite de la valeur la plus défavorable vers la valeur la plus favorable.
- Une liste de valeurs limites pour les critères numériques, qui sert à traduire les valeurs quantitatives en valeurs qualitatives (la valeur symbolique). Cette discrétisation est nécessaire pour déterminer les valeurs symboliques de la base de

règles et pour obtenir les valeurs d'utilité nécessaires à l'utilisation des méthodes multicritère.

1.2 Saisie de données des actions potentielles

Quand tous les critères de sélection ont été définis, l'utilisateur doit entrer les données des actions potentielles. Ceci peut se faire directement à partir de bases de données existantes ou par la fenêtre de saisie montrée dans la Figure 1-3. Le type d'un critère est dans cette fenêtre indiqué par sa couleur et par son emplacement:

Critères déduits dont la valeur est déterminée en fonction de valeurs d'autres critères par une base de règles.

Critères symboliques, qui prennent une valeur symbolique d'une liste de valeurs possibles.

Critères numériques, qui prennent une valeur quantitative.

Figure 1-3. Fenêtre de modification des secteurs.

Les critères déduits sont des critères qualitatifs pour lesquels il existe des règles qui permettent de déduire la valeur à partir des valeurs d'autres critères (ce qui demande la

Chapitre 7. Perspectives: Des explications pour les décisions négociées

présence d'une base de règles). Un exemple simple d'une telle règle pourrait être «SI Exposition Visuelle est Réduite et Valeur Ecologique est Moyenne ALORS Esthétique est Favorable». En absence d'une règle applicable, la valeur ancienne ne change pas, ce qui permet à l'utilisateur d'entrer les valeurs de ces critères à la main. Cette valeur saisie à la main ne changera pas avant qu'une règle soit applicable. Ces critères sont identifiés par la couleur turquoise de leur titre.

Les critères qualitatifs (non-déduits) suivent directement les critères déduits. Pour modifier leur valeur il suffit de cliquer sur le bouton où la valeur actuelle est affichée, et de choisir une nouvelle valeur dans la liste des valeurs qui s'affiche.

Les critères numériques se situent après les critères qualitatifs. Pour modifier une valeur, il suffit de cliquer sur la valeur actuelle et d'en saisir une nouvelle (à l'aide du pavé numérique). Seuls les valeurs numériques ou un champ vide sont acceptés. Un champ vide correspond à la valeur «Indéterminé».

La place disponible dans la fenêtre de modification des secteurs ne permet pas toujours de bien expliquer la signification de chaque critère. Dans ce cas il est possible d'obtenir de l'information supplémentaire en cliquant avec la souris sur le champ qui contient le nom du critère. Pour les critères dont les valeurs sont déduites par la base de règles il est, en plus, possible d'obtenir une explication sur la valeur du critère. Si c'est une règle qui a permis de déduire la valeur du critère, cette règle est identifiée et utilisée pour fournir l'explication.

1.3 Affichage des résultats

Les classements des actions potentielles sont accessibles à l'aide du bouton indiquant la méthode souhaitée dans la fenêtre d'accueil. L'affichage d'un classement se compose d'une liste (deux listes identiques pour la méthode Electre I, comme dans la Figure 1-4) qui montre le classement du meilleur au pire. Dans le cas d'Electre I, les limites de concordance et de discordance sont affichées pour pouvoir les modifier facilement.

Pour obtenir une explication du classement d'une action potentielle il suffit de la choisir dans la liste. L'explication est affichée dans le champ de texte correspondant. Il est également possible d'obtenir un fichier texte avec le classement des secteurs suivi par une explication pour chaque secteur.

StocDech a été conçu pour être aussi automatique que possible au niveau des mises à jour des résultats et la gestion des données en général. Il n’y a aucune limitation ni sur le nombre de critères de sélection ni sur le nombre d’actions potentielles. Le décideur peut donc se concentrer sur les trois étapes de décision - définition des critères, saisie des actions potentielles et analyse des résultats. Les fonctionnalités d’explication permettent d’ajouter une quatrième étape. Par contre, StocDech n’offre pas de fonctionnalités automatiques d’analyse de sensibilité et de robustesse.

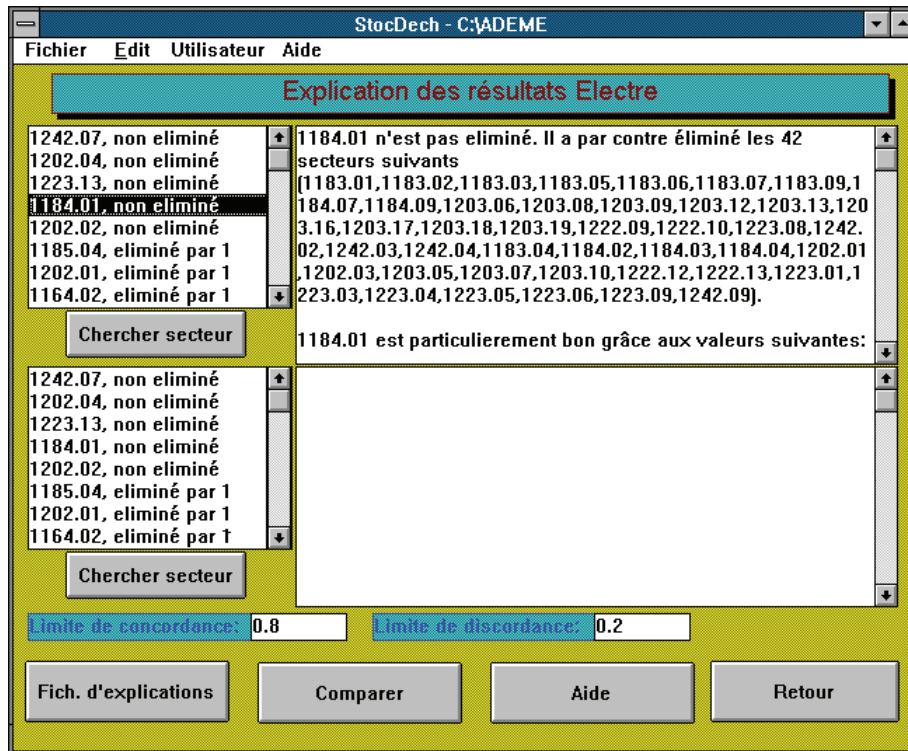


Figure 1-4. Fenêtre de résultat de la méthode Electre.

2. Implémentation des méthodes d'aide à la décision multicritère

Les méthodes d'aide à la décision employées dans le logiciel StocDech appartiennent d'une part aux méthodes de choix multicritère (la méthode Electre I et la somme pondérée) et d'autre part aux systèmes à base de règles. Ce sont des méthodes de nature assez différentes au niveau de la représentation des données et du raisonnement. Il faut donc d'abord définir un système d'équivalence qui permet d'appliquer ces méthodes de la même façon.

2.1 Représentation des données dans StocDech

Les critères de sélection peuvent prendre soit des valeurs qualitatives comme «grand», «petit», etc., comme pour les critères symboliques, soit des valeurs quantitatives comme pour les critères numériques. Pour les méthodes d'aide à la décision, il est avantageux de traiter toute l'information d'une façon qualitative. Pour pouvoir le faire, les valeurs quantitatives sont traduites en valeurs qualitatives par des intervalles de valeurs de la façon indiquée dans la Figure 2-1.

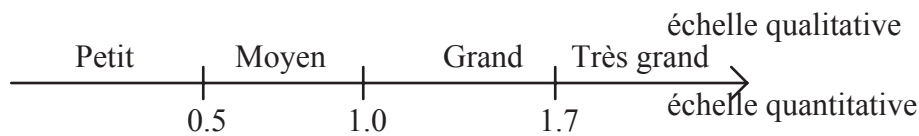


Figure 2-1. Exemple de transformation des valeurs numériques aux valeurs symboliques.

L'utilisation des méthodes de choix multicritère comme la somme pondérée et Electre I nécessite une transformation supplémentaire de cette information qualitative en *valeur d'utilité*. Cette transformation est faite d'une façon très simple: la première valeur qualitative définie pour un critère correspond à une valeur d'utilité de un, la deuxième à une valeur d'utilité de deux etc., comme indiqué dans la Figure 2-2.

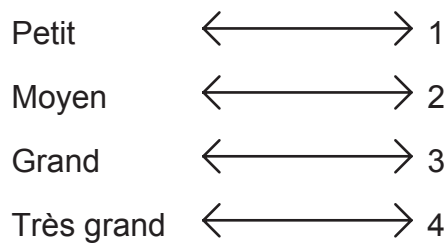


Figure 2-2. Exemple d'une transformation des valeurs symboliques en valeurs d'utilité.

La valeur «indéterminée» peut être utilisée, mais elle correspond par défaut à une valeur d'utilité de zéro. La valeur indéterminée est donc le pire des cas (la valeur la plus mauvaise), mais cela peut être changé par la définition de la valeur «indéterminé» comme une des valeurs possibles pour un critère.

2.2 La méthode de la somme pondérée

Cette méthode est facile à calculer et à comprendre, ce qui constitue un de ses avantages principaux. Des méthodes dérivées de la somme pondérée existent, comme la méthode AHP (Analytical Hierarchy Process) de Saaty [Saaty, 1984], qui permet une hiérarchisation de critères de sélection. Des logiciels existent pour utiliser cette méthode [Grand & Juliot, 1990]. La valeur de préférence de l'action i est calculée de la façon suivante:

$$n_i = \sum_{j=1}^N p_j \cdot g_j(a_i)$$

où:

- n_i est la valeur de préférence de l'action i ,
- N est le nombre de critères,
- p_j est le poids du critère j ,
- $g_j(a_i)$ est la valeur d'utilité de l'action i pour le critère j .

Les valeurs d'utilité sont divisées par l'échelle des valeurs d'utilité possibles du critère pour les normaliser entre zéro et un. La valeur de la somme pondérée maximale est alors la somme des poids de tous les critères. Il est important de connaître le type de normalisation utilisé car le résultat de la méthode peut parfois dépendre de la normalisation, comme cela est montré dans [Pomerol & Barba-Romero, 1993].

2.3 Electre I

Les méthodes Electre sont principalement issues des travaux de B. Roy, dont le premier (Electre I) est présenté dans [Roy, 1968]. Elles sont des méthodes multicritère d'un type appelé agrégation partielle (c'est à dire regrouper les actions potentielles dans des classes ordonnées). Il existe actuellement au moins cinq versions d'Electre [Rommel, 1989; Pomerol & Barba-Romero, 1993], [Maystre et al., 1994].

La méthode Electre I a été choisie pour sa simplicité et pour pouvoir comparer les résultats obtenus avec ceux trouvés pour le problème de choix de site en Suisse, où la même méthode a

Chapitre 7. Perspectives: Des explications pour les décisions négociées

été utilisée. Il existe néanmoins plusieurs interprétations possibles de la méthode Electre I [Bertier & Bourouche, 1975], [Pomerol & Barba-Romero, 1993].

Toutes les versions d'Electre utilisent à peu près le même principe. Elles n'ont pas comme but de calculer des valeurs de préférence globales pour chaque action potentielle, elles les comparent plutôt deux à deux pour déterminer si l'une surclasse l'autre suffisamment pour décider s'il y a une préférence entre les deux.

Deux calculs sont utilisés simultanément pour déterminer les surclassements - celui d'indices de concordance et celui d'indices de discordance. Le rôle de l'indice de concordance est de comparer deux actions au niveau de tous les critères. Si l'action a est meilleur que l'action b pour tous les critères, il est évident que a surclasse b . Si a ne surclasse b que pour un sous-ensemble de critères, le surclassement est moins sûr.

Le rôle de l'indice de discordance est d'empêcher le surclassement dans des cas où b a une valeur beaucoup plus forte que a pour un seul critère, même si a surclasse b par concordance. Le surclassement est dans ce cas empêché parce que les deux actions sont souvent incomparables et il n'est pas possible de déterminer laquelle est meilleure.

L'intérêt de l'indice de discordance est montré par un exemple simple. Si on compare une Rolls Royce et une 2 CV, il est évident que la Rolls Royce surclasse la 2 CV sur chaque critère au niveau de concordance, sauf pour le prix, ce qui donne un indice de discordance très fort. Cette situation implique une incomparabilité entre les deux voitures, c'est à dire qu'il n'est pas possible de dire laquelle surclasse l'autre.

L'indice de concordance est obtenu en faisant la somme des poids des critères pour lesquels le secteur a est supérieur ou égal au secteur b . Cette somme est divisée par la somme des poids de tous les critères. Ainsi a surclassera b par concordance si cet indice \geq au seuil de concordance fixé par l'utilisateur.

$$IC_{a,b} = \frac{\{j : g_j(a) \geq g_j(b)\} \sum p_j}{\sum_{j=1}^N p_j}$$

où:

- $IC_{a,b}$ est l'indice de concordance de l'action a par rapport à l'action b ,

- N est le nombre de critères,
- p_i est le poids du critère i ,
- $g_j(a)$ est la valeur d'utilité de l'action a pour le critère j ,
- $g_j(b)$ est la valeur d'utilité de l'action b pour le critère j .

L'indice de discordance concerne les critères où l'action a est moins bonne que l'action b . On prend le critère où la différence de valeur d'utilité est la plus grande et on divise cette différence par la longueur de l'échelle de valeur d'utilité la plus longue pour obtenir l'indice de discordance. Ainsi a surclasse b par discordance si cet indice \leq au seuil de discordance. Pour dire que a surclasse b , il faut que le surclassement soit fait par concordance ET par discordance en même temps.

$$ID_{a,b} = \frac{\{j: g_j(a) < g_j(b)\} \text{Sup}(g_j(b) - g_j(a))}{E_{Max}}$$

où:

- $ID_{a,b}$ est l'indice de discordance du secteur a par rapport au secteur b ,
- $g_j(a)$ est la valeur d'utilité de l'action a pour le critère j ,
- $g_j(b)$ est la valeur d'utilité de l'action b pour le critère j ,
- E_{Max} est l'échelle de valeur d'utilités la plus longue de tous les critères.

Le choix des seuils de concordance et de discordance détermine le nombre de secteurs qui sont éliminés. Plus le seuil de concordance est élevé, moins de secteurs sont éliminés. Plus le seuil de discordance est petit, moins de secteurs sont éliminés. Les bons seuils sont à trouver par l'utilisateur.

Plusieurs interprétations d'Electre I vont encore au-delà des surclassements simples. Les surclassements d'Electre I ne sont pas forcément transitifs, ce qui veut dire qu'il est possible que a surclasse b en même temps que b surclasse a . Il est même possible d'obtenir des circuits de type « a surclasse b surclasse c surclasse a ». Plusieurs méthodes ont été proposées pour résoudre ce problème, mais il est difficile à dire quelle interprétation est la meilleure [Pomerol & Barba-Romero, 1993].

StocDech ne fait pas de détection de circuits car l'interprétation n'en est pas toujours évidente et dépend plutôt du décideur qui doit traiter cette situation¹⁴. L'approche utilisée est de trier les actions par nombre d'actions surclassantes. Les actions qui sont surclassées par aucune autre action sont encore triées par ordre décroissant d'actions surclassées par elles. Ceci semble être une bonne façon de traiter les risques de non-transitivité.

3. Base de règles

La base de règles est implémentée avec le système Nexpert Object [Neuron Data, 1991]. Une base de règles a été développée uniquement pour le problème de choix de site en Suisse [Rey et al., 1988]. Cette base de règles montre les problèmes au niveau du nombre de règles nécessaires pour traiter le problème. Pour dix critères avec trois valeurs possibles nous avons déjà 3 à puissance 10 combinaisons (59049). Si une règle traite une seule de ces combinaisons, il devient impossible de définir toutes ces règles.

Heureusement, il est possible de définir des règles générales, qui couvrent plusieurs combinaisons à la fois. La difficulté dans la construction d'une base de règles est d'en assurer la cohérence et la complétude (la couverture de chaque cas possible). Dans la base de règles développée pour le problème Suisse, ces problèmes ont été résolus (ou diminués) par une faible précision et des règles très générales.

L'arborescence de concepts intermédiaires utilisés pour définir les règles est présentée dans la Figure 3-1. Cette représentation arborescente permet de définir les critères déduits qui déterminent la valeur des critères de niveau supérieur.

Seulement deux valeurs sont possibles pour tous les critères de haut niveau (c'est-à-dire les critères qui ne sont pas des feuilles de l'arbre). Cette limitation vient du fait que l'on se trouve très vite en face d'une explosion combinatoire si l'on a un plus grand nombre de valeurs

¹⁴ D'autres raisons pour cette interprétation « étrange » de la méthode Electre I sont la volonté des élus de Rhône-Alpes d'utiliser la méthode Electre I, le désir d'éviter de perdre les « brillants seconds » et les contraintes au niveau de temps de calcul et de programmation.

possibles. Deux valeurs possibles pour six critères donnent 64 combinaisons possibles. S'il y a un critère avec trois valeurs possibles il y aura 96 combinaisons. S'il y a deux critères avec trois valeurs possibles il y aura 144 combinaisons, et c'est le cas ici pour les critères «Géologie globale» et «Aptitude agricole». Le critère «Résultat final» a quatre valeurs possibles, qui sont «Très bon», «Bon», «Moyen» et «Mauvais».

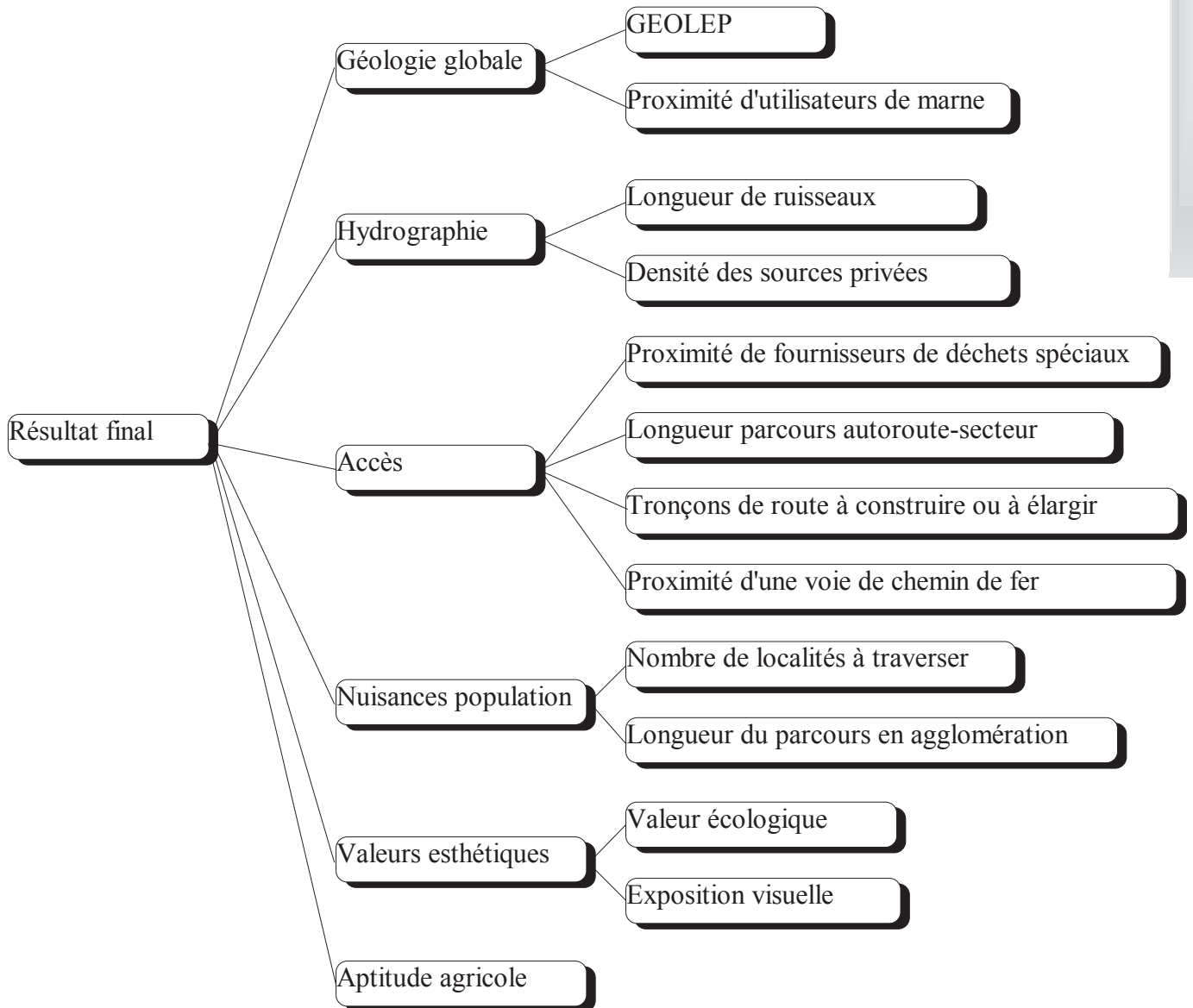


Figure 3-1. Arborescence de concepts intermédiaires pour le choix de site en Suisse.

Les importances des critères déduits sont obtenues par la somme des poids des critères de base utilisés pour trouver la valeur du critère déduit (les poids des critères de base sont ceux indiqués dans l'Annexe A). Elles sont donc les suivantes:

Chapitre 7. Perspectives: Des explications pour les décisions négociées

- (C1) Nuisances population 22.5
- (C2) Valeurs esthétiques 20.5
- (C3) Accès 18.5
- (C4) Géologie globale 15.5
- (C5) Aptitude agricole 15
- (C6) Hydrographie 8

Les trois critères les plus importants sont ainsi «Accès», «Nuisances population» et «Valeurs esthétiques». Les règles pour obtenir le résultat final peuvent être définies de la façon suivante:

Les 6 critères ont la valeur MAX (maximale, la meilleure) => «Très bon»

5 critères ont la valeur MAX dont 3 parmi les critères les plus importants => «Très bon»

4 critères ont la valeur MAX dont 3 parmi les critères les plus importants => «Bon»

3 critères parmi les critères les plus importants ont la valeur MAX => «Moyen»

2 critères parmi les critères les plus importants ont la valeur MAX => «Moyen»

1 critère parmi les critères les plus importants a la valeur MAX => «Mauvais»

0 critères parmi les critères les plus importants a la valeur MAX => «Mauvais»

Pour prendre en compte des valeurs autres que MAX (et MIN), il est possible d'ajouter d'autres règles. Il est alors possible de définir par exemple la règle suivante:

SI Nuisances population = MAX & Valeurs esthétiques = MAX & Accès = MAX & Géologie globale = MAX & Aptitude agricole = «Moyenne» & Hydrographie = MAX

ALORS Résultat final = «Très bon»

Les règles pour déduire les valeurs des critères de haut niveau sont les suivants:

Géologie globale («bon», «moyen» ou «mauvais»)

SI Géologie GEOLEP = GEOLEP 7 & Proximité d'util. de marne < 5 km ALORS
Géologie globale = «bon»

Chapitre 7. Perspectives: Des explications pour les décisions négociées

SI Géologie GEOLEP = GEOLEP 7 & Proximité d'util. de marne = «5 km <= d <= 10 km» ALORS Géologie globale = «bon»

SI Géologie GEOLEP = GEOLEP 8 & Proximité d'util. de marne < 5 km ALORS Géologie globale = «bon»

SI Géologie GEOLEP = GEOLEP 9 & Proximité d'util. de marne < 5 km ALORS Géologie globale = «moyen»

SI Géologie GEOLEP = GEOLEP 5 ALORS Géologie globale = «bon»

SI Géologie GEOLEP = GEOLEP 6 ALORS Géologie globale = «bon»

SI Géologie GEOLEP = GEOLEP 7 ALORS Géologie globale = «moyen»

SI Géologie GEOLEP = GEOLEP 8 ALORS Géologie globale = «moyen»

SI Géologie GEOLEP = GEOLEP 9 ALORS Géologie globale = «mauvais»

SI Géologie GEOLEP = GEOLEP 10 ALORS Géologie globale = «mauvais»

Hydrographie («bon» ou «mauvais»)

SI Longueur de ruisseaux = Aucune & Densité de sources privées = Aucune ALORS Hydrographie = Bon

SI Longueur de ruisseaux = Aucune & Densité de sources privées = Faible ALORS Hydrographie = Bon

SI Longueur de ruisseaux = «<= 10 m/ha» & Densité de sources privées = Aucune ALORS Hydrographie = Bon

Pour les autres cas: Hydrographie = Mauvais

Accès («bon» ou «mauvais»)

SI Proximité de fournisseurs de déchets spéciaux = Favorable ALORS Accès = Bon

SI Proximité de fournisseurs de déchets spéciaux = Moyenne & Longueur parcours autoroute-secteur = Faible ALORS Accès = Bon

Chapitre 7. Perspectives: Des explications pour les décisions négociées

SI Proximité de fournisseurs de déchets spéciaux = Moyenne & Longueur parcours autoroute-secteur = Moyenne & Tronçons de route à construire ou à élargir = Aucun ALORS Accès = Bon

SI Proximité de fournisseurs de déchets spéciaux = Moyenne & Longueur parcours autoroute-secteur = Moyenne & Tronçons de route à construire ou à élargir = A élargir ALORS Accès = Bon

SI Proximité d'une voie de chemin de fer = Présente ALORS Accès = Bon

Pour les autres cas: Accès = Mauvais

Nuisances population («faible» ou «forte»)

SI Nombre de localités à traverser = Aucune & Longueur du parcours en agglomération = Nul ALORS Nuisances population = Faible

SI Nombre de localités à traverser = Aucune & Longueur du parcours en agglomération = «< 1 km» ALORS Nuisances population = Faible

SI Nombre de localités à traverser = Une & Longueur du parcours en agglomération = Nul ALORS Nuisances population = Faible

SI Nombre de localités à traverser = Une & Longueur du parcours en agglomération = «< 1 km» ALORS Nuisances population = Faible

SI Nombre de localités à traverser = Deux & Longueur du parcours en agglomération = «< 1 km» ALORS Nuisances population = Faible

Pour les autres cas: Nuisances population = Forte

Valeurs esthétiques («favorable» ou «défavorable»)

SI Valeur écologique = Faible & Exposition visuelle = Réduite ALORS Valeurs esthétiques = Favorable

SI Valeur écologique = Faible & Exposition visuelle = Moyenne ALORS Valeurs esthétiques = Favorable

SI Valeur écologique = Moyenne & Exposition visuelle = Réduite ALORS Valeurs esthétiques = Favorable

Pour les autres cas: Valeurs esthétiques = Défavorable

Annexe D

Tableaux des données utilisées et des résultats des différentes méthodes

Cette annexe montre les données brutes et les résultats obtenus pour les deux applications de choix de voiture personnelle et le choix de sites de stockage de déchets industriels ultimes en Suisse. Les informations sur le choix de sites de stockage de déchets industriels ultimes dans la région Rhône-Alpes n'ont pas été incluses pour des raisons de confidentialité. Les données montrées sont des données « brutes », qui n'ont subi aucune transformation de normalisation. Les transformations spécifiques à chaque méthode, comme les transformations quantitative-qualitative-note et normalisations sont expliquées ailleurs dans ce document de thèse.

Les classements obtenus pour toutes les actions potentielles sont montrés pour la méthode de somme pondérée, le SAD neuronal et le SIAD neuronal. Pour le choix de voiture, le classement indiqué pour la somme pondérée est celui de la somme pondérée modifiée par les trois règles « non-linéarisantes ». Pour l'application choix de sites de stockage de déchets en Suisse, les classements obtenus par la méthode Electre I et par une base de règles sont indiqués pour permettre certaines comparaisons. Par contre, l'approche SIAD n'a pas été utilisée dans cette application.

1. Données et résultats pour le choix de voiture

Les données techniques viennent des deux revues automobile « L'auto-journal N° 14/15 1995 » et « Action Auto Moto, N° 2 1995 ». Ceci a permis d'identifier des critères de sélection généralement utilisés pour le choix de voiture, car les journaux du domaine savent quelles informations intéressent les consommateurs le plus. Les données sont donc réelles, ce

Chapitre 7. Perspectives: Des explications pour les décisions négociées

qui augmente la valeur des résultats obtenus par rapport à un problème complètement fictif.

Les critères de sélection retenus sont:

Prix, exprimé en francs français.

Puissance maximale, exprimée en chevaux « réels ».

Type de transmission, automatique ou manuelle.

Roues motrices, les possibilités sont traction avant, arrière ou à quatre roues.

Dimensions extérieures, en m³ calculés par la multiplication
longueur*largeur*hauteur.

Capacité du coffre en dm³ (litres).

Poids en kg.

Vitesse maximale en km/h.

Parcours 1000 mètres avec départ à l'arrêt, exprimé en secondes.

Consommation moyenne en litres par 100 km.

Esthétique, information subjective du décideur (l'auteur du document, dans ce cas).

Exprimé par une note sur une échelle de 1 à 5.

Image de marque, information subjective du décideur (l'auteur du document, dans ce cas). Exprimé par une note sur une échelle de 1 à 4.

Équipement en série, note de 0 à 83, calculée à partir d'un sous-ensemble de critères.

Les données de toutes les voitures potentielles sont indiquées dans le Tableau 1-1. Les voitures ont été acceptées en appliquant certaines conditions. Premièrement, sont exclus les voitures à plus de 300.000,- Fr et sans intérêt personnelle (Cabriolets, Wagons, Tout Terrain, très petites, ...). Le nombre de voitures utilisées a ainsi pu être réduit à 113. L'ensemble de voitures ainsi obtenu correspond à une certaine clientèle, qui est la petite famille ayant des ressources suffisantes pour sans problème pouvoir acheter une voiture coûtant jusqu'à 200.000,- francs.

Marque, modèle	1	2	3	4	5	6	7	8	9	10	11	12	13
Alfa-Romeo 33 Q4	137500	132	man.	2x2	9	290	1070	198		8	2	3	44
Alfa-Romeo 155 2.0	136900	143	man.	Avant	10.9	525	1340	205	30.5	9.08	3	3	27
Alfa-Romeo 155 V6	169900	166	man.	Avant	10.9	525	1420	215	30.2	11.5	3	3	35
Alfa-Romeo 155 Q4	194900	190	man.	2x2	10.9	410	1535	225	29.2	12.1	3	3	35
Alfa-Romeo 164 V6	210900	184	man.	Avant	11.4	504	1400	230			4	3	57
Alfa-Romeo 164 V6 Turbo	231000	205	man.	Avant	11.4	504	1500	237	28.1	11.4	4	3	57
Alfa-Romeo 164 V6 24V	248700	210	man.	Avant	11.4	504	1500	240	28	11.4	4	3	57
Alfa-Romeo 164 Q4	283000	230	man.	2x2	10.9	410	1680	240	29.1	13.6	4	3	71
Audi 80 2.0 E	142700	115	man.	Avant	10.7	430	1230	190	33	9.7	4	4	28
Audi 80 16S	152300	140	man.	Avant	10.7	430	1270	201	31	10.5	4	4	36
Audi 80 2.6E	185800	150	man.	Avant	10.7	430	1330	212	30.6	11.6	4	4	49
Audi A6 2.0	155900	115	man.	Avant	12.2	510	1355	190	33.4	9.8	4	4	42
Audi A6 2.6	186600	150	man.	Avant	12.2	510	1440	209	31.3	11.3	4	4	42
Audi A6 2.8	229300	174	man.	Avant	12.2	510	1445	219	30	12.3	4	4	55
BMW 316i Pack	144900	102	man.	Arriere	10.5	435	1190	191	32.7	9	4	5	57
BMW 318i Pack	155900	116	man.	Arriere	10.5	435	1205	198	31.7	9.3	4	5	51
BMW 320i Pack	193000	150	man.	Arriere	10.5	435	1315	214	30.5	10	4	5	62
BMW 325i	225300	192	man.	Arriere	10.5	435	1330	233	28.1	10.6	4	5	62
BMW 520i	189900	150	man.	Arriere	11.6	460	1445	211	30.9	10.6	4	5	44
BMW 525i Worldline	236000	192	man.	Arriere	11.6	460	1480	230	28.8	11	4	5	56
Citroën ZX Aura 1.8 i	109900	103	man.	Avant	9.7	343	1008	188	35.3	10.1	2	3	30
Citroën ZX Volcane 2.0 i	119900	123	man.	Avant	9.7	343	1104	201	32	9.4	3	3	35
Citroën ZX 16V	135000	150	man.	Avant	9.6	324	1150	220	30.9	9.9	3	3	49
Citroën Xantia 1.8i SX	120000	103	man.	Avant	10.7	480	1776	188	33.4	10.2	3	3	31
Citroën Xantia 2.0i VSX	154400	123	man.	Avant	10.7	480	1238	198	32	10.9	3	3	44
Citroën Xantia 16V	177800	150	man.	Avant	10.7	480	1325	213	31.6	10.9	3	3	49
Citroën XM 2.0i VSX	167500	135	man.	Avant	11.7	540	1395	205	32.8	10.8	4	3	42
Citroën XM Turbo C.T. VSX	199000	150	man.	Avant	11.7	540	1416	215	31.3	12.1	4	3	49
Citroën XM V6 VSX	230000	170	man.	Avant	11.7	540	1493	222	29.8	12.6	4	3	49
Citroën XM V6-24	298100	200	man.	Avant	11.7	540	1565	235	29	12.6	4	3	70
Fiat Tipo 2.0 16V	120900	139	man.	Avant	9.7	350	1180	202	31.1	10.4	3	2	51
Fiat Tempra 1.8 SX	99200	103	man.	Avant	10.6	500	1200	188	32.6	9.2	3	2	36
Fiat Croma 2.0 16V	143800	137	man.	Avant	11.4	500	1285	200	30.4	10.4	3	2	44
Fiat Croma 2.0 Turbo	179900	150	man.	Avant	11.4	500	1300	210	30	10.4	3	2	47
Ford Escort 1800i Ghia	90900	105	man.	Avant	9.6	380	1035	187	33	9.2	3	3	37
Ford Mondeo 1800 Ghia	127400	115	man.	Avant	10.7	470	1277	197	32.5	9.1	4	3	49
Ford Mondeo 2000 Ghia	141000	136	man.	Avant	10.7	470	1287	208	31.2	10	4	3	57

Chapitre 7. Perspectives: Des explications pour les décisions négociées

Ford Mondeo V6 Ghia	160000	170	man.	Avant	10.7	470	1300	220	30	10.5	4	3	57
Ford Scorpio 2000 Ghia	174000	120	man.	Arrière	11.9	440	1315	192	33	11	4	3	69
Ford Scorpio 2900 V6 24V	258300	195	auto	Arrière	11.9	440	1345	225	29.6	12.6	4	3	83
Honda Civic 1.6 VTi	133000	160	man.	Avant	10.2	376	1105	215	29	9.1	3	3	61
Honda Concerto 1.6i S	110700	111	man.	Avant	10	363	1060	190	32.6	9.6	3	3	56
Honda Accord 2.0 i	105500	115	man.	Avant	11	405	1240	197	32.2	9.5	4	3	23
Honda Accord E ES	170500	131	man.	Avant	11	405	1300	200	32.5	9.9	4	3	77
Lancia Delta 2.0 HF	147500	186	man.	Avant	9.5	240	1330	220	28.5	11.3	4	4	41
Lancia Delta HF Integrale	252500	215	man.	4x4	9.4	200	1340	220	28.5	11.3	4	4	45
Lancia Dedra 2.0 LS	139000	113	man.	Avant	10.6	480	1210	195	32.7	10.3	4	4	43
Lancia Dedra HF Turbo	186700	162	man.	Avant	10.6	480	1275	215	29.2	11.4	4	4	50
Lancia Dedra HF Integrale	209700	169	man.	4x4	10.6	450	1375	215	29	11.4	4	4	50
Lancia Thema 16v LS	190700	152	man.	Avant	11.5	550	1300	205	30.7	10.6	4	4	56
Lancia Thema Turbo 16v LS	214400	201	man.	Avant	11.5	550	1330	230	28	11.6	4	4	63
Lancia Thema 3.0 V6 LX	272400	171	man.	Avant	11.5	440	1330	220	29.2	11.6	4	4	71
Mazda 323 GTi	99900	128	man.	Avant	9.2	336	1040	196	30.4	9.6	2	3	36
Mazda 626 GLXi 2000	134700	116	man.	Avant	11.4	455	1175	200	31.7	8.9	4	3	44
Mazda 626 Gti 2500	178900	165	man.	Avant	11.4	455	1280	220	31.5	9	4	3	56
Mazda Xedos 6	157300	144	man.	Avant	10.4	407	1195	214	30.9	9.3	4	3	51
Mazda Xedos 9	205900	143	man.	Avant	11.9	420	1410	202	31.4	10.3	4	3	62
Mercedes C 200	174000	136	man.	Arrière	11	430	1365	198	31.9	9.9	4	5	44
Mercedes C 220	197000	150	man.	Arrière	11	430	1410	210	31.7	10.8	4	5	44
Mercedes C 280	223000	193	man.	Arrière	11	430	1490	230	29.3	11.7	4	5	44
Mercedes E 200 Prestige	214000	136	man.	Arrière	11.8	520	1360	200	31.2	10.6	4	5	64
Mercedes E 220 Prestige	242500	150	man.	Arrière	11.8	520	1370	210	30.4	10.7	4	5	64
Mercedes E 280 Prestige	271000	193	man.	Arrière	11.8	520	1490	230	29.5	13.3	4	5	64
Opel Astra CD 1.6 Si	104500	100	man.	Avant	9.7	360	1010	190	32.9	8.7	3	4	42
Opel Vectra CDX 2.0	135000	136	man.	Avant	10.6	530	1200	210	30.2	8.7	4	4	68
Opel Vectra V6	168000	170	man.	Avant	10.7	530	1265	233	28.5	10.3	4	4	77
Opel Omega CD 2.0	174000	136	man.	Arrière	12.4	530	1425	210	31.5	9.7	4	4	68
Opel Omega CD V6	194000	170	man.	Arrière	12.4	530	1510	223	31	11	4	4	68
Opel Omega MV6	248000	211	man.	Arrière	12.4	530	1575	240	29.2	11.4	4	4	83
Peugeot 306 XT 1.8	110300	103	man.	Avant	9.4	338	1080	185	33.8	9.7	3	4	31
Peugeot 306 S16	145300	150	man.	Avant	9.4	338	1160	215	30.3	10.2	3	4	49
Peugeot 405 2.0	133700	123	man.	Avant	10.4	470	1135	197	32.8	9.3	3	4	43
Peugeot 405 Mi16	165000	150	man.	Avant	10.5	470	1180	215	31.1	10.1	3	4	43
Peugeot 405 T16	224000	200	man.	4x4	10.5	392	1340	235	27.7	12.3	3	4	58
Peugeot 605 SRI	167500	135	man.	Avant	12.1	497	1440	205	32.3	10.3	4	4	44

Chapitre 7. Perspectives: Des explications pour les décisions négociées

Peugeot 605 SV3.0	262000	170	man.	Avant	12.1	497	1570	222	29.8	12.6	4	4	69
Renault Clio 16V	119000	137	man.	Avant	8.3	265	1000	209	30.2	8.9	3	3	41
Renault 19 16S	134000	137	man.	Avant	9.8	386	1135	212	30.7	9.3	3	3	41
Renault Laguna RXE 2.0	139000	115	man.	Avant	11.3	452	1245	200	33.2	10	5	3	43
Renault Laguna V6	189000	170	man.	Avant	11.3	452	1370	220	30	12.3	5	3	75
Renault Laguna Baccara V6	202000	170	man.	Avant	11.3	452	1370	220	30	12.3	5	3	77
Renault Safrane RT 2.0	163000	107	man.	Avant	12.4	480	1370	189	33.9	10.2	4	3	41
Renault Safrane RXE 2.2	235000	140	man.	Avant	12.4	480	1410	206	32.9	11.8	4	3	75
Renault Safrane RXE V6	262000	170	man.	Avant	12.4	480	1465	220	30.4	12.2	4	3	75
Rover 220 GSi	119500	136	man.	Avant	9.9	350	1160	202	31	9.4	3	3	42
Rover 420 GSi	119500	136	man.	Avant	10.3	410	1165	201	30.4	9.3	4	3	49
Rover 620 Si ABS	139150	131	man.	Avant	10.9	430	1270	200	32.3	9.9	4	3	50
Rover 623 Si	169000	158	man.	Avant	10.9	430	1320	216	31	10.5	4	3	62
Rover 820 Ti Lux	196850	180	man.	Avant	11.7	530	1410	220	29	11.1	4	3	71
SAAB 900 S 2.0-16	147800	130	man.	Avant	11.4	451	1285	200	31.8	10.2	4	5	50
SAAB 900 S 2.3-16	158400	150	man.	Avant	11.4	451	1295	210	30.3	10.1	4	5	55
SAAB 900 SE 2.5 V6	207000	170	man.	Avant	11.4	451	1430	225	28.9	10.8	4	5	68
SAAB 900 SE 2.0 Turbo	217395	185	man.	Avant	11.4	451	1415	230	28.3	10.4	4	5	68
SAAB 9000 CSE 2.3-16	212700	147	man.	Avant	12	610	1335	210	31.6	10.9	4	5	69
SAAB 9000 CS Ecopower	247000	170	man.	Avant	12	610	1345	220	29.1	11.9	4	5	71
SAAB 9000 CSE 2.3 Turbo	278000	200	man.	Avant	12	610	1355	235	27.5	10.9	4	5	71
Toyota Carina E 2.0 GLi	132450	133	man.	Avant	10.8	470	1220	205	31.3	8.6	4	4	42
Volvo 440 GLE 2.0	123900	109	man.	Avant	10.2	330	1040	186	32	9.8	3	5	45
Volvo 440 1.8 Turbo	134000	120	man.	Avant	10.2	330	1069	197	30.7	9.9	3	5	55
Volvo 850 GLT 2.0	170000	143	man.	Avant	11.6	445	1340	202	33.6	10.4	4	5	54
Volvo 850 GLT 2.5	189000	170	man.	Avant	11.6	445	1330	215	30.1	10.4	4	5	54
Volvo 850 T5	239000	225	man.	Avant	11.6	445	1415	240	27.7	11.7	5	5	71
Volvo 940 GLE 2.3	140000	130	man.	Arrière	12	471	1383	185	31	12	4	5	44
Volvo 940 2.3 Coté Sud	157000	130	man.	Arrière	12	471	1383	185	31	12	4	5	75
VW Golf GL	107900	90	man.	Avant	9.6	330	1020	180	33.4	9.2	3	3	42
VW Golf GTI 16 S	142100	150	man.	Avant	9.6	330	1090	215	29.5	9.2	3	3	50
VW Golf VR 6	170100	174	man.	Avant	9.6	330	1155	225	28.1	10.4	3	3	55
VW Vento GL	107900	90	man.	Avant	10.5	550	1075	180	33.4	9.2	3	3	34
VW Vento GT	124300	150	man.	Avant	10.5	550	1090	215	30	9.2	3	3	39
VW Vento VR6	174500	174	man.	Avant	10.5	550	1210	225	28.5	10.4	3	3	55
VW Passat GL	146800	115	man.	Avant	11.3	495	1240	195	32.8	9.1	4	3	50
VW Passat GT 16S	169500	150	man.	Avant	11.3	495	1315	213	30	10	4	3	55
VW Passat VR6	197800	174	man.	Avant	11.3	495	1365	224	29.2	11.6	4	3	55

Tableau 1-1. Données de base pour le choix de voiture.

Les critères utilisés pour évaluer le niveau d'équipement en série sont de type binaire, ce qui signifie qu'un équipement est soit fourni en série, soit il ne l'est pas. Chaque critère a un poids qui est choisi par le décideur en fonction de ses préférences. La note maximale devient ainsi 83, la somme des poids. Les critères utilisés sont les mêmes que ceux de « L'auto-journal »:

- Direction assistée, poids 9.
- ABS, poids 8.
- Climatisation, poids 7.
- Airbag conducteur, poids 6.
- Banquette arrière fractionnée, poids 6.
- Radio K7 , poids 6.
- Rétroviseurs électriquement réglables, poids 5.
- Roues alliage, poids 5.
- Sellerie cuir, poids 2.
- Sièges chauffantes, poids 7.
- Toit ouvrant, poids 6.
- Verrouillage centralisé, poids 8.
- Quatre vitres électriques (avant et arrière), poids 8.

Les données de toutes les voitures potentielles concernant l'équipement en série sont indiquées dans le Tableau 1-2.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

Marque, modèle	1	2	3	4	5	6	7	8	9	10	11	12	13
Alfa-Romeo 33 Q4	1	1			1			1				1	1
Alfa-Romeo 155 2.0	1						1	1				1	
Alfa-Romeo 155 V6	1	1					1	1				1	
Alfa-Romeo 155 Q4	1	1					1	1				1	
Alfa-Romeo 164 V6	1	1	1				1	1		1		1	1
Alfa-Romeo 164 V6 Turbo	1	1	1				1	1		1		1	1
Alfa-Romeo 164 V6 24V	1	1	1				1	1		1		1	1
Alfa-Romeo 164 Q4	1	1	1	1		1	1	1	1	1		1	1
Audi 80 2.0 E	1				1		1					1	
Audi 80 16S	1	1			1		1					1	
Audi 80 2.6E	1	1			1		1	1				1	1
Audi A6 2.0	1	1		1	1		1					1	
Audi A6 2.6	1	1		1	1		1					1	
Audi A6 2.8	1	1		1	1		1	1				1	1
BMW 316i Pack	1	1	1	1	1		1					1	1
BMW 318i Pack	1	1	1	1			1					1	1
BMW 320i Pack	1	1	1	1	1		1	1				1	1
BMW 325i	1	1	1	1	1		1	1				1	1
BMW 520i	1	1		1			1					1	1
BMW 525i Worldline	1	1	1	1			1	1				1	1
Citroën ZX Aura 1.8 i	1						1					1	1
Citroën ZX Volcane 2.0 i	1						1	1				1	1
Citroën ZX 16V	1	1			1		1	1				1	1
Citroën Xantia 1.8i SX	1				1							1	1
Citroën Xantia 2.0i VSX	1	1			1		1					1	1
Citroën Xantia 16V	1	1			1		1	1				1	1
Citroën XM 2.0i VSX	1	1		1	1		1					1	
Citroën XM Turbo C.T. VSX	1	1	1	1	1		1					1	
Citroën XM V6 VSX	1	1	1	1	1		1					1	
Citroën XM V6-24	1	1	1	1	1	1	1	1	1			1	1
Fiat Tipo 2.0 16V	1	1			1		1	1	1			1	1
Fiat Tempra 1.8 SX	1				1		1					1	1
Fiat Croma 2.0 16V	1	1			1			1				1	1
Fiat Croma 2.0 Turbo	1	1	1					1	1			1	1
Ford Escort 1800i Ghia	1			1	1							1	1
Ford Mondeo 1800 Ghia	1		1	1	1		1					1	1
Ford Mondeo 2000 Ghia	1	1	1	1	1		1					1	1

Chapitre 7. Perspectives: Des explications pour les décisions négociées

Ford Mondeo V6 Ghia	1	1	1	1	1		1				1	1
Ford Scorpio 2000 Ghia	1	1	1	1	1	1	1				1	1
Ford Scorpio 2900 V6 24V	1	1	1	1	1	1	1	1	1	1	1	1
Honda Civic 1.6 VTi	1	1			1	1	1	1			1	1
Honda Concerto 1.6i S	1	1			1	1	1				1	1
Honda Accord 2.0 i	1					1					1	
Honda Accord E ES	1	1	1	1	1	1	1	1	1	1		1
Lancia Delta 2.0 HF	1	1		1			1	1				1
Lancia Delta HF Integrale	1	1	1		1			1	1			1
Lancia Dedra 2.0 LS	1	1					1	1				1
Lancia Dedra HF Turbo	1	1	1				1	1				1
Lancia Dedra HF Integrale	1	1	1				1	1				1
Lancia Thema 16v LS	1	1	1		1		1	1				1
Lancia Thema Turbo 16v LS	1	1	1		1		1	1		1		1
Lancia Thema 3.0 V6 LX	1	1	1		1	1	1	1	1	1		1
Mazda 323 GTi	1				1		1					1
Mazda 626 GLXi 2000	1	1			1		1					1
Mazda 626 Gti 2500	1	1			1		1	1		1		1
Mazda Xedos 6	1	1			1	1					1	1
Mazda Xedos 9	1	1		1	1	1	1				1	1
Mercedes C 200	1	1		1			1					1
Mercedes C 220	1	1		1			1					1
Mercedes C 280	1	1		1			1					1
Mercedes E 200 Prestige	1	1	1	1	1		1	1	1			1
Mercedes E 220 Prestige	1	1	1	1	1		1	1	1			1
Mercedes E 280 Prestige	1	1	1	1	1		1	1	1			1
Opel Astra CD 1.6 Si	1	1			1	1	1					1
Opel Vectra CDX 2.0	1	1	1	1	1	1	1	1				1
Opel Vectra V6	1	1	1	1	1	1	1	1	1	1		1
Opel Omega CD 2.0	1	1	1	1	1	1	1	1				1
Opel Omega CD V6	1	1	1	1	1	1	1	1				1
Opel Omega MV6	1	1	1	1	1	1	1	1	1	1	1	1
Peugeot 306 XT 1.8	1				1							1
Peugeot 306 S16	1	1			1		1	1				1
Peugeot 405 2.0	1		1			1	1					1
Peugeot 405 Mi16	1	1					1	1				1
Peugeot 405 T16	1	1	1			1	1	1	1			1
Peugeot 605 SRI	1	1		1			1					1

Chapitre 7. Perspectives: Des explications pour les décisions négociées

Peugeot 605 SV3.0	1	1	1	1		1	1	1		1		1	1
Renault Clio 16V	1				1		1	1				1	1
Renault 19 16S	1				1		1	1				1	1
Renault Laguna RXE 2.0	1		1		1		1					1	1
Renault Laguna V6	1	1	1	1	1	1	1	1		1		1	1
Renault Laguna Baccara V6	1	1	1	1	1	1	1	1	1	1		1	1
Renault Safrane RT 2.0	1	1			1		1	1				1	
Renault Safrane RXE 2.2	1	1	1	1	1	1	1	1		1		1	1
Renault Safrane RXE V6	1	1	1	1	1	1	1	1		1		1	1
Rover 220 GSi	1				1		1				1	1	1
Rover 420 GSi	1	1					1	1			1	1	1
Rover 620 Si ABS	1	1		1	1		1					1	1
Rover 623 Si	1	1	1		1		1	1			1	1	1
Rover 820 Ti Lux	1	1	1	1			1	1	1	1	1	1	1
SAAB 900 S 2.0-16	1	1		1	1		1					1	1
SAAB 900 S 2.3-16	1	1		1	1		1	1				1	1
SAAB 900 SE 2.5 V6	1	1	1	1	1	1	1	1				1	1
SAAB 900 SE 2.0 Turbo	1	1	1	1	1	1	1	1				1	1
SAAB 9000 CSE 2.3-16	1	1	1	1	1		1	1		1		1	1
SAAB 9000 CS Ecopower	1	1	1	1	1		1	1	1	1		1	1
SAAB 9000 CSE 2.3 Turbo	1	1	1	1	1		1	1	1	1		1	1
Toyota Carina E 2.0 GLi	1				1	1	1					1	1
Volvo 440 GLE 2.0	1	1		1	1							1	1
Volvo 440 1.8 Turbo	1	1		1	1		1	1				1	1
Volvo 850 GLT 2.0	1	1		1	1		1	1		1		1	
Volvo 850 GLT 2.5	1	1		1	1		1	1		1		1	
Volvo 850 T5	1	1	1	1	1		1	1	1	1		1	1
Volvo 940 GLE 2.3	1	1		1	1					1		1	
Volvo 940 2.3 Coté Sud	1	1	1	1	1	1	1	1		1		1	1
VW Golf GL	1			1	1		1					1	1
VW Golf GTI 16 S	1	1		1	1			1				1	1
VW Golf VR 6	1	1		1	1		1	1				1	1
VW Vento GL	1			1	1		1					1	
VW Vento GT	1			1	1		1	1				1	
VW Vento VR6	1	1		1	1		1	1				1	1
VW Passat GL	1	1		1	1		1					1	1
VW Passat GT 16S	1	1		1	1		1	1				1	1
VW Passat VR6	1	1		1	1		1	1				1	1

Tableau 1-2. Données sur les équipements en série

1.1 Classement par la somme pondérée modifiée

Le classement « correct » obtenu par une somme pondérée modifiée par trois règles, avec indication de la valeur de préférence, est le suivant:

Opel Vectra V6	77.70	BMW 316i Pack	65.61	Audi A6 2.0	59.53
SAAB 900 SE 2.0 Turbo	76.69	WW Golf VR 6	65.01	Volvo 440 GLE 2.0	59.40
Volvo 850 T5	76.40	Lancia Delta 2.0 HF	64.98	Fiat Tipo 2.0 16V	59.05
BMW 325i	74.26	WW Passat GT 16S	64.88	Renault Clio 16V	58.97
Lancia Thema Turbo 16v LS	73.81	Mazda Xedos 9	64.65	Audi A6 2.8	58.92
SAAB 900 SE 2.5 V6	72.21	WW Passat VR6	64.64	Fiat Croma 2.0 16V	58.84
Opel Omega MV6	70.52	SAAB 9000 CS Ecopower	64.35	Mazda 323 GTi	58.75
Volvo 850 GLT 2.5	70.45	Peugeot 405 Mi16	64.19	Fiat Croma 2.0 Turbo	58.69
BMW 525i Worldline	70.32	Mercedes C 200	64.14	Citroen XM Turbo C.T. VSX	58.64
Honda Civic 1.6 VTi	69.82	Volvo 850 GLT 2.0	63.88	Honda Concerto 1.6i S	58.56
Opel Vectra CDX 2.0	69.60	WW Golf GTI 16 S	63.49	WW Passat GL	58.45
Mazda Xedos 6	69.59	Ford Mondeo 2000 Ghia	63.35	Citroen Xantia 2.0i VSX	58.42
Rover 820 Ti Lux	69.57	Alfa-Romeo 164 V6 24V	63.31	Renault Laguna RXE 2.0	58.38
Opel Omega CD V6	69.15	Mazda 626 GLXi 2000	63.11	Opel Astra CD 1.6 Si	58.24
Alfa-Romeo 164 V6 Turbo	69.12	Lancia Thema 16v LS	62.75	Ford Scorpio 2000 Ghia	58.23
BMW 320i Pack	69.00	Audi 80 2.6E	62.70	Lancia Dedra 2.0 LS	58.03
BMW 520i	68.88	Volvo 440 1.8 Turbo	62.68	Fiat Tempra 1.8 SX	57.36
WW Vento VR6	68.18	Peugeot 405 2.0	62.67	Renault Safrane RXE 2.2	57.03
Mercedes C 280	67.78	Audi A6 2.6	62.62	Citroen ZX Volcane 2.0 i	56.97
Opel Omega CD 2.0	67.72	SAAB 900 S 2.0-16	62.52	Audi 80 2.0 E	56.36
Lancia Dedra HF Integrale	67.57	Mercedes C 220	62.33	Ford Escort 1800i Ghia	55.28
SAAB 900 S 2.3-16	67.42	Mercedes E 220 Prestige	62.23	Renault Safrane RT 2.0	54.90
Mercedes E 200 Prestige	67.09	Alfa-Romeo 155 Q4	61.96	Lancia Delta HF Integrale	54.15
SAAB 9000 CSE 2.3-16	66.97	Ford Mondeo 1800 Ghia	61.83	Citroen Xantia 1.8i SX	53.56
Honda Accord E ES	66.64	Citroen ZX 16V	61.82	Citroen XM V6 VSX	53.45
Volvo 940 2.3 Cote Sud	66.56	Renault 19 16S	61.75	WW Vento GL	53.32
Rover 623 Si	66.54	Rover 620 Si ABS	61.69	Peugeot 306 XT 1.8	53.31
Mazda 626 Gti 2500	66.45	Alfa-Romeo 155 2.0	61.66	WW Golf GL	50.49
Ford Mondeo V6 Ghia	66.45	Volvo 940 GLE 2.3	61.62	Citroen ZX Aura 1.8 i	48.80
Rover 420 GSi	66.40	Peugeot 306 S16	61.41	Ford Scorpio 2900 V6 24V	48.14
WW Vento GT	66.37	Audi 80 16S	61.27	Renault Safrane RXE V6	47.10
Toyota Carina E 2.0 GLi	66.27	Peugeot 605 SRI	61.05	Peugeot 605 SV3.0	44.18
Alfa-Romeo 164 V6	66.21	Alfa-Romeo 33 Q4	60.65	SAAB 9000 CSE 2.3 Turbo	42.05
Peugeot 405 T16	66.20	Rover 220 GSi	60.30	Lancia Thema 3.0 V6 LX	41.48
Lancia Dedra HF Turbo	66.01	Honda Accord 2.0 i	60.16	Mercedes E 280 Prestige	29.78
BMW 318i Pack	65.95	Citroen Xantia 16V	59.94	Alfa-Romeo 164 Q4	14.64
Renault Laguna V6	65.77	Alfa-Romeo 155 V6	59.81	Citroen XM V6-24	4.93
Renault Laguna Baccara V6	65.72	Citroen XM 2.0i VSX	59.80		

1.2 Classement par SAD neuronal

Le classement obtenu par le SAD neuronal utilisant un ensemble d'entraînement avec douze voitures réelles et deux pseudo-exemples (voiture idéale et anti-idéale), avec indication de la valeur de préférence, est le suivant:

SAAB 900 SE 2.0 Turbo	76.91	Volvo 940 GLE 2.3	64.76	Rover 220 GSi	60.59
Volvo 850 T5	76.41	SAAB 9000 CSE 2.3 Turbo	64.59	Citroen Xantia 2.0i VSX	60.57
BMW 325i	74.73	Peugeot 605 SRI	64.54	Renault Safrane RT 2.0	60.52
BMW 525i Worldline	71.09	Mazda Xedos 6	64.42	Lancia Dedra 2.0 LS	60.25
Opel Omega CD 2.0	70.64	Toyota Carina E 2.0 GLi	64.27	Rover 820 Ti Lux	60.09
Volvo 850 GLT 2.5	70.63	Audi 80 16S	64.11	Honda Accord 2.0 i	60.05
SAAB 900 SE 2.5 V6	70.51	Peugeot 306 S16	63.98	Volvo 440 GLE 2.0	60.00
BMW 320i Pack	70.34	Alfa-Romeo 155 2.0	63.90	Audi 80 2.0 E	59.86
Opel Vectra V6	70.16	Alfa-Romeo 155 V6	63.87	Ford Mondeo 1800 Ghia	59.65
Mercedes C 280	69.83	SAAB 9000 CS Ecopower	63.84	Opel Astra CD 1.6 Si	58.97
Lancia Dedra HF Turbo	69.52	Ford Scorpio 2000 Ghia	63.73	Alfa-Romeo 164 V6	58.81
BMW 520i	69.51	Peugeot 405 T16	63.67	Audi A6 2.8	58.49
Lancia Dedra HF Integrale	68.94	Lancia Delta HF Integrale	63.45	Renault Laguna Baccara V6	58.44
Opel Omega CD V6	68.92	WW Golf GTI 16 S	63.41	Renault Clio 16V	58.38
Ford Mondeo V6 Ghia	68.80	Ford Mondeo 2000 Ghia	63.13	Citroen ZX Volcane 2.0 i	58.25
Lancia Delta 2.0 HF	68.15	BMW 316i Pack	63.06	WW Vento GL	57.57
Mercedes E 220 Prestige	68.10	Lancia Thema Turbo 16v LS	63.05	Honda Concerto 1.6i S	57.39
WW Passat GT 16S	68.02	Citroen XM 2.0i VSX	62.76	Fiat Tipo 2.0 16V	57.05
SAAB 900 S 2.3-16	67.89	Citroen Xantia 16V	62.68	Mazda 323 GTi	56.89
Mercedes E 200 Prestige	67.46	Mazda 626 GLXi 2000	62.55	Fiat Tempra 1.8 SX	56.04
Mazda 626 Gti 2500	67.27	Peugeot 405 2.0	62.54	Ford Escort 1800i Ghia	55.59
Mercedes C 220	67.22	Fiat Croma 2.0 Turbo	62.54	Renault Safrane RXE 2.2	55.44
Lancia Thema 16v LS	66.54	Rover 420 GSi	62.39	Mercedes E 280 Prestige	55.44
Audi 80 2.6E	66.33	Volvo 440 1.8 Turbo	62.38	Peugeot 306 XT 1.8	55.41
WW Vento VR6	66.29	SAAB 900 S 2.0-16	62.30	Lancia Thema 3.0 V6 LX	54.24
Rover 623 Si	66.19	Renault 19 16S	62.08	Citroen Xantia 1.8i SX	54.05
Peugeot 405 Mi16	66.14	Citroen ZX 16V	62.07	WW Golf GL	53.55
WW Vento GT	66.03	Fiat Croma 2.0 16V	61.95	Opel Omega MV6	50.66
Audi A6 2.6	65.97	Volvo 850 GLT 2.0	61.88	Citroen XM V6 VSX	50.15
Honda Civic 1.6 VTi	65.92	Citroen XM Turbo C.T. VSX	61.53	Citroen ZX Aura 1.8 i	48.74
Volvo 940 2.3 Cote Sud	65.81	Rover 620 Si ABS	61.49	Ford Scorpio 2900 V6 24V	47.72
Opel Vectra CDX 2.0	65.74	Audi A6 2.0	61.42	Alfa-Romeo 164 V6 Turbo	45.64
Mercedes C 200	65.41	Honda Accord E ES	61.18	Peugeot 605 SV3.0	41.44
WW Golf VR 6	65.35	Renault Laguna V6	61.03	Alfa-Romeo 164 Q4	39.73
SAAB 9000 CSE 2.3-16	64.99	WW Passat GL	61.02	Renault Safrane RXE V6	38.93
BMW 318i Pack	64.92	Alfa-Romeo 33 Q4	60.67	Alfa-Romeo 164 V6 24V	38.72
WW Passat VR6	64.81	Renault Laguna RXE 2.0	60.67	Citroen XM V6-24	5.03
Mazda Xedos 9	64.76	Alfa-Romeo 155 Q4	60.63		

1.3 Classement par SIAD neuronal

Le classement obtenu par le SIAD neuronal utilisant un ensemble d'entraînement avec douze voitures réelles et 48 pseudo-exemples (y compris voiture idéale et anti-idéale), avec indication de la valeur de préférence, est le suivant:

SAAB 900 SE 2.0 Turbo	87.47	Alfa-Romeo 164 V6	64.14	Alfa-Romeo 33 Q4	56.59
Opel Vectra V6	87.24	Opel Astra CD 1.6 Si	63.89	Fiat Tempra 1.8 SX	56.17
SAAB 900 SE 2.5 V6	82.63	Mazda Xedos 9	63.76	WW Golf GL	56.15
Volvo 850 GLT 2.5	79.37	Audi A6 2.6	63.75	Volvo 440 GLE 2.0	55.71
Opel Vectra CDX 2.0	78.36	BMW 316i Pack	63.75	Peugeot 306 XT 1.8	55.29
SAAB 900 S 2.3-16	77.30	Alfa-Romeo 155 2.0	63.61	Renault Safrane RT 2.0	55.17
Lancia Thema Turbo 16v LS	77.30	WW Golf GTI 16 S	63.43	Alfa-Romeo 155 V6	54.73
Volvo 850 T5	76.69	BMW 525i Worldline	63.35	Lancia Dedra 2.0 LS	54.08
Mazda 626 Gti 2500	74.66	WW Golf VR 6	63.14	Mercedes E 200 Prestige	53.84
Toyota Carina E 2.0 GLi	73.69	Peugeot 306 S16	63.13	Fiat Croma 2.0 16V	53.79
Lancia Delta 2.0 HF	72.56	Mercedes C 280	62.98	Volvo 940 2.3 Cote Sud	53.58
Ford Mondeo V6 Ghia	72.30	Peugeot 405 Mi16	62.85	Citroen Xantia 1.8i SX	53.45
BMW 325i	71.95	SAAB 9000 CS Ecopower	62.77	Ford Scorpio 2000 Ghia	52.55
Rover 820 Ti Lux	71.95	SAAB 900 S 2.0-16	62.68	Fiat Croma 2.0 Turbo	52.51
Opel Omega CD 2.0	71.47	Ford Mondeo 1800 Ghia	62.50	Citroen XM Turbo C.T. VSX	51.92
WW Passat GT 16S	71.22	Peugeot 405 2.0	61.90	Opel Omega MV6	51.80
BMW 320i Pack	71.00	Audi 80 16S	61.66	Citroen Xantia 16V	51.48
Honda Civic 1.6 VTi	70.67	Renault Clio 16V	61.63	Volvo 940 GLE 2.3	51.01
Rover 623 Si	69.11	Renault Laguna RXE 2.0	61.42	Lancia Dedra HF Integrale	50.89
Opel Omega CD V6	68.95	WW Passat GL	61.38	Fiat Tipo 2.0 16V	50.74
Renault Laguna V6	68.23	Renault 19 16S	60.76	Citroen ZX Aura 1.8 i	49.48
Lancia Dedra HF Turbo	67.85	Volvo 440 1.8 Turbo	60.72	Alfa-Romeo 164 V6 24V	48.86
BMW 520i	67.74	Rover 620 Si ABS	60.63	Mercedes E 220 Prestige	48.16
Mazda Xedos 6	66.96	Citroen ZX 16V	60.54	SAAB 9000 CSE 2.3 Turbo	47.62
WW Vento GT	66.86	Rover 220 GSi	60.47	Peugeot 405 T16	46.51
Honda Accord E ES	66.60	Mercedes C 220	59.88	Citroen Xantia 2.0i VSX	45.89
WW Passat VR6	66.32	Ford Escort 1800i Ghia	59.72	Renault Safrane RXE 2.2	45.48
SAAB 9000 CSE 2.3-16	66.28	Volvo 850 GLT 2.0	59.47	Citroen XM V6 VSX	44.70
BMW 318i Pack	66.14	Audi 80 2.6E	59.08	Alfa-Romeo 155 Q4	43.64
Peugeot 605 SRI	66.09	Audi A6 2.0	58.95	Lancia Thema 3.0 V6 LX	42.40
Lancia Thema 16v LS	66.01	Alfa-Romeo 164 V6 Turbo	58.43	Lancia Delta HF Integrale	39.80
Rover 420 GSi	65.79	Mazda 323 GTi	58.42	Peugeot 605 SV3.0	39.74
WW Vento VR6	65.61	Audi A6 2.8	58.33	Renault Safrane RXE V6	38.36
Mazda 626 GLXi 2000	65.58	Audi 80 2.0 E	58.26	Ford Scorpio 2900 V6 24V	31.69
Renault Laguna Baccara V6	65.54	Citroen ZX Volcane 2.0 i	57.54	Mercedes E 280 Prestige	29.16
Honda Accord 2.0 i	64.63	Honda Concerto 1.6i S	57.28	Alfa-Romeo 164 Q4	14.20
Mercedes C 200	64.21	Citroen XM 2.0i VSX	57.03	Citroen XM V6-24	5.89
Ford Mondeo 2000 Ghia	64.15	WW Vento GL	56.72		

2. Les données et les résultats de l'application choix de sites de stockage de déchets industriels ultimes en Suisse

Les deux problèmes de choix de sites de stockage de déchets industriels ultimes traités dans la région Rhône-Alpes en France et en Suisse sont pratiquement équivalents. Le cas Suisse

présente néanmoins l'avantage de pouvoir publier les données utilisées, car elles se trouvent dans [Rey et al., 1988]. Les droits de publication des données sur Rhône-Alpes sont moins clairs. La grande quantité de données rend également les données de Rhône-Alpes difficiles à publier.

Les données Suisses présentent une différence importante par rapport aux données de Rhône-Alpes. Les données de Rhône-Alpes sont toutes quantitatives, tandis que les données Suisses sont qualitatives aussi bien que quantitatives. Les critères de sélection utilisés dans l'étude Suisse sont:

Aptitude agricole du terrain. Exprimé par une note sur une échelle de 1 à 3. Poids 15.0.

Proximité d'une voie de chemin de fer. Exprimé par une note sur une échelle de 1 à 3.

Poids 5.0.

Proximité d'utilisateurs de marne. Exprimé par une note sur une échelle de 1 à 3.

Poids 3.5.

Exposition visuelle. Exprimé par une note sur une échelle de 1 à 3. Poids 11.0.

Appréciation de la géologie globale. Exprimé par une note sur une échelle de 1 à 6.

Poids 12.0.

Nombre de localités à traverser. Exprimé sur une échelle de 1 à 6, où 6 correspond à aucune localité et 1 à cinq localités ou plus. Poids 12.5.

Longueur de ruisseaux. . Exprimé par une note sur une échelle de 1 à 3. Poids 4.5.

Tronçons à construire ou à élargir. Exprimé par une note sur une échelle de 1 à 3.

Poids 4.0.

Proximité des fournisseurs de déchets spéciaux. Exprimé par une note sur une échelle de 1 à 3. Poids 4.5.

Longueur du parcours en agglomération en mètres. Poids 10.0.

Longueur du parcours sortie d'autoroute-secteur en km. Poids 5.0.

Densité des sources privées en nombre/ha. Poids 3.5.

Valeur écologique. Exprimé par une note sur une échelle de 1 à 3. Poids 9.5.

Les valeurs de chaque secteur sont indiquées dans le Tableau 2-1.

Chapitre 7. Perspectives: Des explications pour les décisions négociées

Nom du secteur	1	2	3	4	5	6	7	8	9	10	11	12	13
1164.03	2	1	2	2	5	3	3	3	1	900	10.4	0	1
1183.01	1	2	1	3	1	3	1	3	2	500	14	0	2
1183.02	1	1	1	3	2	4	3	2	2	200	6.3	0	3
1183.03	1	3	1	3	3	4	1	1	2	500	5	0	1
1183.05	1	1	1	3	3	4	3	3	2	500	4	0	3
1183.06	1	1	1	3	2	3	3	1	2	800	8	0.01	2
1183.07	1	1	1	1	3	4	3	3	2	600	3	0	2
1183.09	2	2	1	2	4	3	3	1	2	1200	9	0.01	1
1184.07	2	1	2	3	4	4	3	1	1	1300	6.2	0	2
1184.09	2	2	3	3	4	4	3	3	1	1400	7	0	1
1185.02	2	1	2	3	5	4	3	1	1	1000	5	0	2
1202.05	2	1	1	1	5	4	2	3	2	500	6	0.01	2
1203.04	2	1	2	1	5	4	3	1	3	500	7	0.01	2
1203.06	2	1	2	3	3	3	2	1	3	1200	7.7	0.04	2
1203.08	2	1	2	2	2	5	3	3	2	0	3.2	0	2
1203.09	2	1	2	2	1	4	3	1	2	500	4.9	0	3
1203.12	2	1	2	2	2	5	3	3	2	0	6.1	0	2
1203.13	2	1	3	2	4	3	3	1	2	1000	10	0	3
1203.16	2	1	3	2	2	2	3	1	2	1500	11.7	0	2
1203.17	2	1	3	2	2	4	3	3	2	300	10.8	0.02	2
1203.18	2	1	3	3	2	5	3	2	2	0	10.6	0.11	2
1203.19	1	1	1	3	3	4	2	2	2	2400	7	0.02	1
1222.04	1	1	2	2	5	1	3	3	3	2200	10.2	0	3
1222.07	1	1	2	1	5	2	3	3	3	1900	7	0.03	3
1222.08	1	1	3	1	5	3	2	3	3	1500	9	0	2
1222.09	1	1	2	2	3	3	3	3	3	1200	5.6	0.02	2
1222.10	1	1	2	2	4	2	3	2	3	1500	7.3	0	3
1223.08	2	1	2	2	3	2	3	1	3	1400	5.9	0.01	3
1223.11	2	3	3	1	4	1	3	1	3	2500	9	0	3
1242.02	1	2	1	1	3	3	3	2	3	500	4.4	0.02	3
1242.03	1	1	2	2	4	4	2	2	3	1800	7	0.02	1
1242.04	1	1	2	1	4	4	2	1	3	1000	8	0	1
1242.05	1	1	2	1	5	4	3	1	3	700	7	0.01	2
1164.02	2	1	2	3	5	4	3	3	1	700	8.2	0	2
1183.04	1	1	1	2	3	5	3	3	2	0	4	0	3
1184.01	2	2	2	3	3	5	3	3	2	0	0.5	0	3
1184.02	2	2	3	2	4	4	3	1	2	500	3.2	0	3

1184.03	2	1	3	3	4	5	1	2	1	0	8.2	0	1
1184.04	2	1	3	3	4	3	2	3	1	900	4.3	0	3
1184.05	2	1	3	1	6	4	3	3	1	1400	7.4	0	2
1184.06	2	1	3	2	6	4	2	3	1	300	9.6	0.01	2
1184.08E	2	1	3	2	6	2	3	3	1	1800	8.2	0	3
1184.08N	2	1	3	2	6	4	1	3	1	1000	7.8	0	2
1184.08W	2	1	3	2	6	3	3	3	1	400	14.2	0	3
1185.04	2	1	2	3	6	3	3	2	1	600	10	0	3
1202.01	2	3	3	2	3	5	2	2	3	0	9	0.01	2
1202.03	1	1	2	2	3	5	3	3	3	0	0.4	0	3
1202.04	2	1	1	2	5	5	3	3	3	0	2	0.01	2
1202.02	1	3	2	3	5	5	3	1	3	0	6.4	0	2
1203.02	2	1	2	1	5	4	3	2	3	900	3.1	0	3
1203.05	2	1	2	3	4	4	2	1	3	600	4.5	0.01	2
1203.07	2	1	2	3	2	4	3	3	3	600	4.6	0	2
1203.10	2	1	2	3	4	4	3	3	3	500	7	0	1
1222.12	1	1	3	2	4	4	3	3	3	400	3.1	0.01	3
1222.13	1	1	3	1	4	5	3	3	3	0	1	0	3
1223.01	2	1	3	2	3	4	3	3	3	300	2	0.01	3
1223.03	2	2	2	3	4	2	3	2	3	1500	8	0	2
1223.04	2	1	3	2	4	2	3	2	3	1000	5.2	0	3
1223.05	2	1	3	2	3	3	3	2	3	900	5.1	0	3
1223.06	2	1	3	3	2	3	3	3	3	1200	5.7	0	3
1223.07	2	3	2	2	4	2	3	1	3	1700	7	0.01	3
1223.09	2	1	3	2	3	3	3	3	3	1300	8	0	3
1223.13	3	3	3	3	4	2	3	3	3	1800	12.6	0.01	2
1242.07	1	1	3	3	5	5	3	3	3	0	5.1	0.12	3
1242.09	1	1	3	2	4	5	3	2	3	0	5	0	2

Tableau 2-1. Données de base pour le choix de sites de stockage de déchets industriels ultimes en Suisse.

2.1 Classement par somme pondérée

Le classement obtenu par la méthode de somme pondérée est indiqué ci-dessous. Ceci est le classement « correct » utilisé pour définir les exemples d'apprentissage de l'approche

neuronale. Le premier chiffre indique la place dans le classement, le deuxième est le nom du secteur et le troisième la valeur de préférence.

1184.01	78.76	1203.18	62.00	1202.05	53.90
1242.07	72.40	1203.12	61.58	1203.06	52.69
1202.04	70.48	1185.02	61.31	1223.08	51.79
1202.02	69.93	1242.09	61.28	1164.03	50.38
1223.01	69.60	1223.05	61.25	1242.02	48.52
1223.13	69.16	1183.05	60.62	1222.10	48.08
1202.01	67.39	1223.09	60.60	1242.05	47.49
1184.02	67.31	1183.04	60.25	1222.09	47.28
1185.04	67.12	1223.04	60.09	1183.03	47.26
1202.03	65.55	1184.09	60.08	1223.11	47.08
1164.02	65.35	1184.03	59.37	1183.09	45.99
1223.06	64.93	1223.03	58.57	1222.04	45.50
1184.04	64.69	1184.08N	58.44	1183.07	44.83
1184.08W	64.65	1184.08E	58.10	1222.07	44.61
1203.07	64.35	1223.07	57.59	1222.08	44.23
1222.13	63.98	1184.07	57.27	1183.06	43.40
1203.10	63.93	1203.13	57.22	1183.01	42.32
1222.12	63.71	1203.17	56.72	1203.16	41.93
1203.05	62.65	1183.02	56.59	1242.03	40.90
1203.02	62.65	1184.05	55.99	1203.19	37.60
1203.08	62.64	1203.04	55.79	1242.04	36.82
1184.06	62.55	1203.09	55.24		

2.2 Classement par Electre I

Le classement obtenu par la méthode Electre I est indiqué ci-dessous. Le premier chiffre indique la place dans le classement, suivie par le nom du secteur, la mention si le secteur a été éliminé ou s'il a plutôt éliminé d'autres secteurs et le nombre de secteurs éliminés ou éliminants. La limite de concordance utilisée était de 0.8 et la limite de discordance de 0.2.

1242.07, a éliminé 14	1223.11, éliminé par 4	1223.05, éliminé par 8
1202.02, a éliminé 18	1242.09, éliminé par 4	1183.02, éliminé par 10
1223.13, a éliminé 10	1184.06, éliminé par 4	1242.05, éliminé par 11
1184.01, a éliminé 42	1202.03, éliminé par 5	1184.07, éliminé par 11
1202.04, a éliminé 30	1222.08, éliminé par 5	1223.04, éliminé par 12
1202.01, éliminé par 1	1203.05, éliminé par 5	1203.09, éliminé par 15
1184.03, éliminé par 1	1203.07, éliminé par 5	1203.06, éliminé par 15
1185.04, éliminé par 1	1203.18, éliminé par 5	1203.13, éliminé par 16
1164.02, éliminé par 1	1223.06, éliminé par 5	1203.17, éliminé par 17
1203.02, éliminé par 1	1183.04, éliminé par 5	1242.02, éliminé par 17
1184.04, éliminé par 2	1185.02, éliminé par 5	1183.07, éliminé par 19
1184.08E, éliminé par 2	1222.04, éliminé par 6	1223.08, éliminé par 20
1184.08W, éliminé par 2	1183.05, éliminé par 6	1242.03, éliminé par 20
1203.10, éliminé par 2	1202.05, éliminé par 7	1183.01, éliminé par 20
1184.02, éliminé par 2	1203.12, éliminé par 7	1222.10, éliminé par 21
1222.13, éliminé par 3	1203.08, éliminé par 7	1242.04, éliminé par 21
1223.07, éliminé par 3	1222.12, éliminé par 7	1203.19, éliminé par 26
1184.09, éliminé par 3	1203.04, éliminé par 8	1222.09, éliminé par 27
1183.03, éliminé par 4	1222.07, éliminé par 8	1183.06, éliminé par 28
1223.01, éliminé par 4	1164.03, éliminé par 8	1183.09, éliminé par 28
1184.08N, éliminé par 4	1223.09, éliminé par 8	1203.16, éliminé par 39
1184.05, éliminé par 4	1223.03, éliminé par 8	

2.3 Classement par SAD neuronal

Le classement donné par le SAD neuronal, obtenu avec un ensemble d'apprentissage de 12 exemples et deux pseudo-exemples, est indiqué ci-dessous. Le premier chiffre indique la place dans le classement, le deuxième est le nom du secteur et le troisième la valeur de préférence calculée.

1184.01 71.42	1203.12 60.80		1223.11 54.76
1202.02 69.92	1203.10 60.61		1223.08 54.21
1223.13 67.29	1184.08W	60.41	1183.03 53.93
1202.01 66.35	1223.09 60.40		1202.05 53.92
1223.01 66.15	1203.17 60.14		1222.09 53.64
1184.02 64.87	1184.06 59.91		1242.02 53.20
1242.07 64.56	1223.04 59.90		1203.06 52.69
1222.13 63.98	1184.07 59.86		1183.07 52.68
1222.12 63.72	1223.03 59.74		1183.06 52.44
1223.06 63.63	1223.07 59.61		1203.16 52.33
1202.03 63.07	1183.05 58.42		1222.10 51.97
1184.09 62.81	1183.04 58.41		1183.01 51.89
1184.04 62.62	1203.13 58.02		1203.04 51.55
1203.07 62.07	1184.08E	58.01	1183.09 50.93
1203.18 62.00	1184.03 57.44		1222.08 50.18
1164.02 61.77	1203.09 57.33		1222.04 48.27
1203.08 61.57	1203.02 57.24		1242.05 47.98
1185.04 61.37	1184.08N	57.19	1203.19 47.79
1185.02 61.33	1184.05 57.09		1242.03 45.69
1223.05 61.27	1203.05 56.91		1222.07 44.63
1202.04 61.11	1183.02 56.59		1242.04 36.82
1242.09 60.88	1164.03 55.08		

2.4 Classement par base de règles

Le classement obtenu par une base de règles est indiqué ci-dessous. La description complète de la base de règles utilisée se trouve dans l'Annexe C. Le premier chiffre indique la place dans le classement, le deuxième est le nom du secteur et le troisième la valeur de préférence qualitative, qui peut prendre les quatre valeurs « très bon », « bon », « moyen » et « mauvais ».

Il est à noter qu'il y a plus de différences entre le classement obtenu par la base de règles qu'entre les classements obtenus par d'autres méthodes. Ceci montre bien l'effet de la « non-linéarité » de la fonction de préférence. La base de règles permet, par exemple, de supprimer la redondance qu'il y a entre l'accès routière et l'accès par chemin de fer. Si un existe, l'utilité de l'autre diminue toute de suite. C'est notamment pour cette raison que le secteur 1222.12 est très bon ici, même s'il n'est pas très bien placé dans les autres classements.

1184.02 Tres bon		1203.16 Mauvais	
1202.02 Tres bon		1203.17 Mauvais	
1222.12 Tres bon		1203.19 Mauvais	
1223.01 Tres bon		1222.07 Mauvais	
1223.05 Tres bon		1222.08 Mauvais	
1183.02 Bon		1222.09 Mauvais	
1183.05 Bon		1223.11 Mauvais	
1203.09 Bon		1242.03 Mauvais	
1183.04 Bon		1242.04 Mauvais	
1184.01 Bon		1184.03 Mauvais	
1202.03 Bon		1184.05 Mauvais	
1203.05 Bon		1184.06 Mauvais	
1203.07 Bon		1184.08E	Mauvais
1242.07 Bon		1184.08N	Mauvais
1183.01 Moyen			
1183.03 Moyen			
1183.06 Moyen			
1183.07 Moyen			
1202.05 Moyen			
1203.04 Moyen			
1203.06 Moyen			
1203.08 Moyen			
1203.12 Moyen			
1203.18 Moyen			
1222.04 Moyen			
1222.10 Moyen			
1223.08 Moyen			
1242.02 Moyen			
1242.05 Moyen			
1164.02 Moyen			
1184.04 Moyen			
1184.08W	Moyen		
1185.04 Moyen			
1202.01 Moyen			
1202.04 Moyen			
1203.02 Moyen			
1203.10 Moyen			
1222.13 Moyen			
1223.03 Moyen			
1223.04 Moyen			
1223.06 Moyen			
1223.07 Moyen			
1223.09 Moyen			
1223.13 Moyen			
1242.09 Moyen			
1164.03 Mauvais			
1183.09 Mauvais			
1184.07 Mauvais			
1184.09 Mauvais			
1185.02 Mauvais			
1203.13 Mauvais			

Liste des figures

Figure 2-1. Combinaison de l'analyse multicritère et de la base de règles dans StocDech.

Figure 2-2. Exemple d'une fonction de préférence (a) par somme pondérée pour deux critères de sélection, (b) par somme pondérée modifiée par deux règles.

Figure 3-1. Schéma d'un neurone artificiel.

Figure 3-2. Exemples de fonctions de sortie.

Figure 4-1. Structure du réseau de neurones INKA.

Figure 4-2. Algorithme d'apprentissage de INKA.

Figure 5-1. Exemple d'une fonction de préférence non-linéaire et continue pour deux critères de sélection, obtenue par une somme pondérée modifiée pour certaines parties.

Figure 5-2. Position des 21 exemples dans le plan x_1, x_2 avec leur ordre d'apparition.

Figure 5-3. Apprentissage de la fonction de préférence linéaire de la Figure 2-2(a) par INKA. (a) Fonction apprise en utilisant uniquement les deux premiers exemples, (b) fonction apprise après l'utilisation des huit premiers exemples.

Figure 5-4. Apprentissage de la fonction de préférence non-linéaire de la Figure 5-1 par INKA. Les figures illustrent la fonction apprise (a) après l'introduction de 8 exemples, dont aucun venant du « trou », (b) après l'introduction du premier exemple se situant dans le trou (exemple numéro 10), et (c) le résultat final après 21 exemples.

Figure 6-1. Modèle de connectivité de la carte auto-organisatrice.

Figure 6-2. Voisinage pour une carte auto-organisatrice de Kohonen avec une structure logique en deux dimensions.

Figure 6-3. Illustrations de l'apprentissage par carte auto-organisatrice dans le cas d'une espace d'entrée en deux dimensions et une structure logique en deux dimensions.

Figure 6-4. Saisie graphique des notes ordinales quantifiées par le principe de MHM.

Figure 7-1. Représentation symbolique de la carte de Kohonen utilisée.

Figure 9-1. Algorithme général d'un SIAD [Dong, 1992].

Figure 10-1. Méthodologie interactive d'aide à la décision intégrant l'approche neuronale.

Figure 10-2. Evolution de la valeur ajoutée par rapport au coût d'obtention des données [Schlaifer, 1959].

Figure 11-1. Évolution de la valeur de préférence pour l'action 10 en fonction du critère x_1 dans le cas de la fonction de préférence de la Figure 2-2.

Figure 11-2. Évolution de la valeur de préférence en fonction des critères x_1 et x_2 pour la fonction de préférence de la Figure 2-2.

Figure 11-3. Comparaison de la fonction de préférence pour deux actions, permettant d'analyser comment l'une pourrait dépasser l'autre en modifiant la valeur du critère x_1 .

Figure 11-4. Visualisation des coefficients de certitude de INKA en fonction des valeurs de x_1 et de x_2 après l'introduction des deux premiers exemples de la fonction de préférence non-linéaire de la Figure 2-2.

Figure 11-5. Évolution du coefficient de certitude en fonction de la valeur du critère x_1 pour l'action 10.

Figure 11-6. Évolution de la moyenne des coefficients de certitude sur toute l'espace d'entrée en fonction du nombre d'exemples pour la fonction en trois dimensions de la Figure 2-2.

Figure 12-1. Fonction de préférence pour le choix de voiture, critère « volume du coffre », qui devrait être monotone et croissante mais qui ne l'est pas.

Figure 12-2. Analyse graphique de la fonction de préférence pour le critère prix (a) avant l'introduction de pseudo-exemples, (b) après l'introduction de 9 pseudo-exemples pour corriger la fonction pour le critère prix.

Figure 12-3. Évolution du coefficient de certitude moyen de INKA sur l'ensemble de voitures potentielles en fonction du nombre d'exemples.

Figure 13-1. Algorithme de « k-fold cross validation ».

Figure 13-2. (a) RMSE calculé pour l'ensemble d'actions potentielles, (b) valeur moyenne et (c) écart type de RMSE estimés par « leaving-one-out » en fonction du nombre d'exemples pour le choix de voiture.

Figure 13-3. Comparaison entre la différence moyenne réelle et estimée par « leaving-one-out » en fonction du nombre d'exemples pour le choix d'une voiture.

Figure 13-4. Comparaison entre la différence maximale réelle et estimée par « leaving-one-out » en fonction du nombre d'exemples pour le choix d'une voiture.

Figure 14-1. Evolution de l'écart type de la différence moyenne estimée par « leaving-one-out » pour l'application choix de voiture.

Figure 14-2. Pourcentage de stabilité en fonction du nombre d'exemples pour l'application de choix de voiture.

Figure 14-3. Le nombre de voitures constamment parmi les cinq meilleures en fonction du nombre d'exemples pour le problème de choix de voiture.

Figure 15-1. Évolution de la confiance moyenne sur l'ensemble d'actions potentielles en fonction du nombre d'exemples pour le choix d'un site de stockage.

Figure 15-2. (a) RMSE entre le classement donné par la somme pondérée et le classement donné par INKA pour l'ensemble d'actions potentielles, (b) valeur moyenne de RMSE et (c) écart type de RMSE estimés par « leaving-one-out » pour le choix de site.

Figure 15-3. Comparaison entre (a) la différence moyenne entre le classement donné par la somme pondérée et le classement donné par INKA et (b) la différence moyenne estimée par « leaving-one-out » pour le choix de site.

Figure 15-4. Comparaison entre (a) la différence maximale entre le classement calculé par somme pondérée et le classement calculé par INKA et (b) la différence maximale estimée par « leaving-one-out » pour le choix de site.

Figure 15-5. (a) Evolution de l'écart type de la différence moyenne donnée par « leaving-one-out » et (b) le nombre de sites constamment parmi les cinq meilleures.

Figure 17-1. Exemple d'un arbre d'inférence pour classifier un site de stockage de déchets en Suisse.

Figure 18-1. Exemple d'une explication fondée sur la base de règles, produite par StocDech.

Figure 18-2. Exemple d'une explication sur la somme pondérée d'un site suisse, produite par StocDech.

Figure 18-3. Exemple d'une explication sur l'élimination ou non-élimination d'un site de stockage de déchets Suisse par Electre I, produite par StocDech.

Figure 18-4. Explication comparative entre deux sites de stockage de déchets potentiels en Suisse, fondée sur la relation de surclassement de la méthode Electre et produite par StocDech.

Figure 19-1. Illustration du calcul de l'importance et de la valeur d'utilité contextuelle.

Figure 20-1. Exemple d'une explication de type base de règles pour un site suisse, produite automatiquement en utilisant les concepts intermédiaires et les utilités contextuelles.

Figure 20-2. Exemple d'une explication de type somme pondérée pour un site suisse, produite automatiquement en utilisant les importances et les utilités contextuelles.

Figure 20-3. Exemple d'une explication mixte pour un site suisse, produite automatiquement en utilisant les concepts intermédiaires, les importances et les utilités contextuelles.

Figure 21-1. Explication produite automatiquement pour la SAAB 900 S 2.0-16.

Figure 21-2. Évolution du niveau de confiance pour les explications concernant la SAAB 900 S 2.0-16 en fonction du nombre d'exemples.

Figure 21-3. Évolution du niveau de confiance des explications concernant (a) la BMW 316i Pack, (b) la Volvo 850 T5 et (c) la SAAB 900 SE 2.0 Turbo en fonction du nombre d'exemples.

Figure 21-4. Explication par similarité pour la BMW 316i Pack construite à partir d'une explication sur la SAAB 900 S 2.0-16.

Figure 21-5. Explication fournie par le décideur sur la Volvo 850 T5 (gauche) et utilisée comme prototype pour construire l'explication par similarité de la SAAB 900 SE 2.0 Turbo (droite).

Figure 22-1. Système de diagnostic de panne de voiture.

Figure 22-2. Exemple d'un réseau sémantique pour le diagnostic « eau dans le carburateur ».

Figure 22-3. Exemple d'un réseau sémantique pour expliquer le diagnostic « eau dans le carburateur » utilisant les concepts intermédiaires « neuronaux ».

Figure 23-1. L'approche SPREAD.

Figure 23-2. Procédure de décision pour le choix de site.

Glossaire

Le glossaire a été inclus pour servir de référence aux concepts utilisés dans ce mémoire de thèse, sur lesquels il existe des utilisations variées en fonction du domaine et de l'auteur. La définition d'un grand nombre de concepts présentés sont repris de [Maystre et al., 1994], parfois avec des modifications. Les définitions reprises sans modification sont soulignées. Les concepts en *italique* sont définis dans cette liste.

<u>Acteur</u> (d'un processus de sélection)	Individu ou groupe d'individus influençant directement ou indirectement la <i>décision</i> .
---	--

<u>Action</u>	Politique, programme, projet ou candidat faisant l'objet de l'analyse multicritère.
Action consensuelle	Une <i>action potentielle</i> sur lequel les <i>acteurs</i> du problème peuvent se mettre d'accord.
<u>Action potentielle</u>	<i>Action</i> provisoirement jugée possible par un des <i>intervenants</i> au moins ou présumée telle par l' <i>homme d'étude</i> , en vue de l' <i>aide à la décision</i> .
Action de compromis	<i>Action</i> , potentielle ou non, qui permet au <i>décideur</i> d'effectuer des comparaisons avec d'autres <i>actions</i> dans l'objectif d'avancer le processus de recherche vers la <i>recommandation</i> finale.
<u>Action de référence</u>	<i>Action</i> servant de référence pour les <i>actions potentielles</i> dans la problématique β . Elle sert de limite à des catégories auxquelles les <i>actions potentielles</i> seront attribuées.
<u>Agrégation</u>	Opération permettant d'obtenir des informations sur la préférence globale entre les <i>actions potentielles</i> , à partir d'informations par critère.
<u>Aide à la décision</u>	Activité de l' <i>homme d'étude</i> ayant pour objectif d'obtenir des éléments de réponse aux questions que se pose un <i>intervenant</i> d'un <i>processus de décision</i> . Ces éléments concourent à éclairer la <i>décision</i> , à améliorer la cohérence entre l'évolution de ce processus, les objectifs et le système de valeurs de cet <i>intervenant</i> . Normalement, le processus conduit à une <i>recommandation</i> .
Analyse de données	Analyse destinée à mieux comprendre des ensembles de données au niveau des corrélations, des groupes de données, etc.
<u>Analyse de robustesse</u>	Analyse cherchant à déterminer le domaine de variation de certains paramètres dans lequel une <i>recommandation</i> reste robuste (stable).
Analyse de sensibilité	Répétition de l' <i>analyse multicritère</i> originale en faisant varier, isolément puis collectivement, les valeurs attribuées à l'origine à tous

	les paramètres de l' <i>analyse multicritère</i> ou aux facteurs ayant une influence sur les valeurs de ces paramètres.
Analyse factorielle	Méthodes d' <i>analyse de données</i> permettant de remplacer les variables d'origine par des variables plus significatives, les axes factoriels.
Apprentissage automatique	Modification automatique des paramètres du système apprenant pour correspondre au système ayant produit les <i>exemples d'apprentissage</i> .
Apprentissage supervisé	<i>Apprentissage automatique</i> où les bonnes réponses des <i>exemples d'apprentissage</i> sont connus.
Apprentissage non supervisé	<i>Apprentissage automatique</i> où les bonnes réponses des <i>exemples d'apprentissage</i> sont à définir par le système apprenant lui-même, qui doit donc faire une sorte d' <i>analyse de données</i> (classification).
Base de connaissances	Système de représentations de connaissances introduites par un expert ou définies par un <i>apprentissage automatique</i> .
Base de faits	Module d'une <i>base de connaissances</i> qui contient les valeurs des variables et des hypothèses.
Base de règles	Module d'une <i>base de connaissances</i> qui contient les règles permettant d'effectuer un raisonnement sur les faits de la <i>base de faits</i> .
Biais	Variable d'entrée supplémentaire d'un neurone permettant de définir une <i>valeur de sortie</i> par défaut qui correspond au <i>poids</i> associé à cette variable, ayant la valeur constante de un.
Classement	Place d'une <i>action potentielle</i> dans un <i>ordre d'actions</i> potentielles.
Concept intermédiaire	Concept facultatif qui permet de mieux structurer le raisonnement ou les explications d'un système d' <i>aide à la décision</i> , dont la valeur est déterminée par le système de raisonnement même.
Contexte	Situation de <i>décision</i> , dans ce document avant tout le contexte de <i>décision</i> défini par l' <i>action potentielle</i> évaluée.

	<i>décision</i> défini par l' <i>action potentielle</i> évaluée.
Couche	Structure logique pour organiser les neurones et les connexions entre eux dans un réseau de neurones.
Couche cachée	<i>Couche</i> dont les neurones ne sont connectés ni aux entrées ni aux sorties du <i>réseau de neurones</i> .
Couche d'entrée	<i>Couche</i> dont les neurones sont connectés aux entrées du <i>réseau de neurones</i> .
Couche de sortie	<i>Couche</i> dont les neurones sont connectés aux sorties du <i>réseau de neurones</i> .
Critère de sélection	Critère utilisé pour évaluer les <i>actions potentielles</i> dans un <i>processus de décision</i> .
<u>Décideur</u>	Celui des <i>intervenants</i> d'un <i>processus de décision</i> auquel s'adresse l' <i>aide à la décision</i> . Il s'agit de l'entité (par exemple personne physique ou morale, communauté) qui apprécie les « possibles », fixe les objectifs, exprime les préférences et est censée les faire prévaloir dans l'évolution du processus.
Décideur abstrait	Un <i>acteur</i> d'un <i>processus de décision</i> dont les <i>préférences</i> sont accessibles uniquement par l'intermédiaire de données et non par une interaction directe. Les préférences des <i>décideurs</i> abstraits ont une influence directe sur la <i>décision</i> globale et sur son succès d'exécution en particulier, même s'il n'est pas le <i>décideur</i> même.
<u>Décision</u> (globale)	Options arrêtées lors du <i>processus de décision</i> . Choix final du <i>décideur</i> .
Descente du gradient	Méthode de recherche d'une solution optimale à un problème de minimisation.
Ensemble d'actions potentielles	L'ensemble dans lequel la solution ou les solutions d'un problème de choix doit être identifiée.

Ensemble d'entraînement	Ensemble d' <i>exemples d'apprentissage</i> qui sert à modifier les paramètres du système apprenant par l' <i>apprentissage automatique</i> .
Ensemble de validation ou de test	Ensemble d' <i>exemples d'apprentissage</i> qui sert à vérifier les capacités de <i>généralisation</i> acquises par l' <i>apprentissage automatique</i> .
Exemple d'apprentissage	Un vecteur de valeurs d'entrée représentant les caractéristiques d'une <i>action</i> (potentielle ou non) et un vecteur de valeurs de sortie s'il s'agit de l' <i>apprentissage supervisé</i> .
Exemple d'entraînement	Un <i>exemple d'apprentissage</i> utilisé dans le processus d'entraînement.
Exemple de préférence	Un <i>exemple d'entraînement</i> où il y a une <i>valeur de préférence</i> associée aux valeurs des <i>critères de sélection</i> de l' <i>action</i> concernée.
Exemple de validation	Un <i>exemple d'apprentissage</i> utilisé dans le processus de validation des résultats de l'entraînement.
Explication reconstructive	Explication concernant un résultat construite a posteriori, indépendamment du processus de raisonnement.
Fonction d'activation	Fonction qui détermine la <i>valeur d'activation</i> d'un neurone en fonction de ces valeurs d'entrée, ses <i>poids</i> et ses paramètres internes.
Fonction d'utilité	Fonction qui définit la relation entre la valeur d'un <i>critère de sélection</i> et son utilité.
Fonction d'utilité additive	<i>Fonction de préférence</i> qui utilise un calcul additif pour combiner les <i>valeurs d'utilité</i> en une <i>valeur de préférence</i> .
Fonction de préférence	Fonction continue qui représente les préférences d'un <i>acteur</i> dans une <i>processus de décision</i> .
Fonction de préférence contextuelle	<i>Fonction de préférences</i> où l' <i>importance</i> et l' <i>utilité</i> des valeurs des critères peut changer en fonction du <i>contexte</i> , défini par l' <i>action</i> traitée.

Fonction de sortie	La fonction qui détermine la <i>valeur de sortie</i> d'un neurone artificiel en fonction de sa <i>valeur d'activation</i> .
Généralisation	Fonctionnalité qui permet à un système de <i>décision</i> de s'exprimer sur de nouveaux cas grâce à l'information extraite à partir de cas connus.
Graphe de surclassement	Graphe qui représente les relations de surclassement entre <i>actions</i> , telle qu'elles sont définies pour les méthodes Electre, par exemple.
Groupe d'inférence	Un ensemble de règles de production ou autre méthode de résolution destiné à résoudre un sous-problème défini du problème complet.
Homme d'étude	Personne qui fait l'intermédiaire entre les experts / les <i>décideurs</i> et le système informatique d' <i>aide à la décision</i> .
Importance d'un critère	Information subjective fournie par le <i>décideur</i> qui correspond à ses préférences.
Intelligence artificielle	Domaine de recherche provenant avant tout de l'informatique. Consiste à trouver des représentations et des algorithmes permettant de résoudre des problèmes de la même façon que les êtres humains.
Intervalle de valeurs	Un <i>intervalle de valeurs</i> quantitatives qui sert à traduire les valeurs de l'intervalle en une valeur qualitative.
<u>Intervenant</u>	<i>Acteur</i> cherchant à influencer le <i>décideur</i> dans une des étapes du <i>processus de décision</i> , en raison de la nature de ses valeurs et donc en fonction de son système de préférences.
<u>Matrice des évaluations</u>	Matrice contenant les évaluations des performances de chaque <i>action potentielle</i> sur chacun des critères.
Méthode d'acquisition de connaissances	Méthode destinée à faciliter l'interaction avec l'expert ou le <i>décideur</i> , dont les connaissances sont à acquérir.
Minimum global	Une solution optimale à un problème de minimisation.
Minimum local	Une solution sous-optimale à un problème de minimisation.

Modèle de préférence	Un modèle symbolique, mathématique ou autre représentant les préférences d'un <i>décideur</i> ou d'un groupe de <i>décideurs</i> permettant de fournir des <i>recommandations</i> .
Modèle de préférence contextuel	<i>Modèle de préférences</i> permettant un changement de l' <i>importance d'un critère de sélection</i> ou de l' <i>utilité d'une valeur d'un critère de sélection</i> en fonction du <i>contexte</i> défini par l' <i>action</i> évaluée.
Module d'explication	La partie d'un système de <i>décision</i> qui justifie et explique une <i>recommandation</i> du système, déterminé par le <i>module de raisonnement</i> . Le <i>module d'explication</i> peut, en plus, posséder son propre <i>module de raisonnement</i> pour produire les explications.
Module de raisonnement	La partie d'un système de <i>décision</i> qui détermine la <i>recommandation</i> du système.
Moteur d'inférences	Système de résolution des <i>systèmes expert</i> , qui utilise les connaissances dans la <i>base de connaissances</i> pour déduire les valeurs des hypothèses.
Note	(voir <i>Valeur de préférence</i>)
Note ordinale quantifiée	<i>Valeur de préférence</i> qui exprime la préférence du <i>décideur</i> entre deux <i>actions</i> sur une échelle qui définit le degré de préférence.
<u>Ordre</u>	<i>Préordre</i> asymétrique (par exemple si <i>a</i> domine <i>b</i> , alors <i>b</i> ne domine pas <i>a</i>), dans lequel il n'y a pas d'ex aequo.
Poids	<i>Valeur ordinale</i> ou cardinale qui exprime l' <i>importance d'un critère</i> .
<u>Préordre</u>	Relation réflexive (par exemple <i>a</i> surclasse <i>a</i>) et transitive (par exemple si <i>a</i> surclasse <i>b</i> et <i>b</i> surclasse <i>c</i> , alors <i>a</i> surclasse <i>c</i>), dans laquelle les ex aequo sont possibles.
<u>Prescription</u>	Comportement, proposé par l' <i>homme d'étude</i> , ayant un caractère contraignant pour le <i>décideur</i> (comme au sens médical). Suite à un changement de terminologie (1993), ce terme a été remplacé par celui

	de <i>recommandation</i> .
Préservation de topologie	Propriété des cartes auto-organisatrices qui permet de garder une notion de voisinage entre les <i>actions</i> définies par les critères de sélection et la représentation de classes d' <i>actions</i> dans une espace à dimension inférieure.
<u>Processus de décision</u>	Succession d'étapes au cours duquel les multiples options, qui vont conditionner la <i>décision globale</i> , sont définies. La <i>décision globale</i> ne constitue donc qu'une petite partie de ce processus.
<u>Pseudo-critère</u>	Fonction dont le pouvoir discriminant est caractérisé par deux <i>seuils</i> , l'un d'indifférence, l'autre de préférence.
Pseudo-exemple	Un <i>exemple</i> qui n'est pas une <i>action potentielle</i> ou réalisable, mais qui peut être utilisé comme <i>exemple d'apprentissage</i> pour augmenter la précision et la stabilité de l'apprentissage.
<u>Recommandation</u>	Comportement proposé par l' <i>homme d'étude</i> par rapport auquel le <i>décideur</i> conserve son entière liberté. Suite à un changement de terminologie (1993), ce terme remplace celui de <i>prescription</i> .
<u>Seuil (de discrimination)</u>	Valeur fixée par l' <i>homme d'étude</i> indiquant la limite: <ul style="list-style-type: none"> • entre deux situations de préférence, au niveau de chaque <i>critère</i> (<i>seuil</i> de concordance, de discordance, d'indifférence, de préférence, de veto), • entre deux situations de préférence, au niveau global (<i>seuil</i> de concordance globale, de discordance globale), • entre deux valeurs de l'indice de crédibilité du surclassement.
Utilité d'une valeur	Valeur qui exprime l' <i>utilité d'une valeur</i> d'un <i>critère de sélection</i> pour la <i>valeur de préférence</i> d'une <i>action</i> .
Valeur cardinale	Valeur absolue définie sur une échelle prédéfinie, comme les nombre réels dans l'intervalle $[0, 100]$, par exemple.

Valeur d'activation	Niveau « d'excitation » d'un neurone.	
NOM : FRÄMLING	DATE de SOUTENANCE	
Valeur d'utilité	Chiffre qui exprime l'utilité de la valeur d'un critère pour le <i>décideur</i> .	
(avec précision du nom de jeune fille, le cas échéant)		
Valeur ordinale	Valeur relationnelle qui exprime une relation de préférence entre deux valeurs.	
TITRE : Modélisation et apprentissage des préférences par réseaux de neurones pour l'aide à la décision multicritère		
Valeur de préférence	Une « note » qui exprime le niveau de satisfaction du <i>décideur</i> pour une <i>action</i> .	
NATURE : Doctorat	Numéro d'ordre : 96 ISAL 025	
Valeur de sortie	Résultat donné par un réseau neuronal ou un neurone.	
Valeur de sortie	Le « bon » résultat fourni à un réseau neuronal pendant	
Cote B.I.U. - Lyon : T 50/210/19 /	et	bis
attendue	l'apprentissage supervisé pour permettre au réseau d'apprendre.	
RESUME :		
<p>La modélisation des préférences des décideurs pour des problèmes de choix est l'objet principal de ce travail de thèse. Afin d'obtenir des modèles plus réalistes, des fonctions de préférence contextuelle sont présentées dans ce mémoire de thèse. La complexité d'expression de telles fonctions est réduite par le recours à des techniques d'apprentissage automatique à partir de réseaux de neurones.</p> <p>L'identification des fonctions de préférence est réalisée par des réseaux de neurones, fondés sur les principes de la régression non-linéaire, à partir d'exemples de décision. Une telle technique n'est pas toujours applicable en pratique à cause du nombre élevé d'exemples nécessaire. Le réseau neuronal INKA, développé dans le cadre de ce travail, effectue la régression avec un nombre d'exemples relativement petit. INKA offre également des temps d'apprentissage courts par rapport à d'autres techniques, ce qui est nécessaire pour une acquisition interactive de la fonction de préférence.</p> <p>INKA est utilisé dans le système interactif d'aide à la décision (SIAD) présenté, qui est un des premiers à mettre en œuvre un apprentissage automatique d'une fonction de préférence globale. La visualisation de la fonction apprise et les indicateurs de précision et de sensibilité permettent au décideur d'estimer le moment d'arrêter la recherche de solutions. Ceci est particulièrement utile pour apprendre les préférences des décideurs « abstraits » (acteurs sociaux, systèmes naturels, ...), qui ne peuvent pas utiliser directement le système.</p> <p>Les fonctionnalités d'explication développées dans ce travail permettent de justifier les recommandations des SIAD, ce qui a constitué un des grands défis du domaine de l'aide à la décision multicritère et des réseaux neuronaux. L'intérêt de telles explications est de faciliter la prise de décisions négociées dans le cadre de projets d'aménagement complexes ou pour améliorer des produits dont les ventes dépendent</p>		
MOTS-CLES : Aide décision, réseau neuronal, justification, système expert, décision collective, préférence, apprentissage.		
Laboratoire (s) de recherches : Laboratoire Ingénierie de l'Environnement à l'École Nationale Supérieure des Mines de Saint-Étienne, centre SIMADE (Sciences Informatiques, Modélisation, Aide à la Décision et		
Directeur de thèse : Monsieur Albert MATHON, Professeur		
Président de jury :		
Composition du jury :	Messieurs	M INSTA.L.I.É R