Université Lumière Lyon 2
Faculté des Sciences économiques et de gestion
ECOLE DOCTORALE DE SCIENCES COGNITIVES
Thèse de doctorat
Spécialité : Informatique
Présentée et soutenue publiquement par
Edwige P. Fangseu Badjio
Le 10 Décembre 2005

Evaluation qualitative et guidage des utilisateurs en fouille visuelle de données

Directeur de thèse : Djamel A. Zighed
Laboratoire : Laboratoire ERIC de l'Université Lumière Lyon 2 Co-encadrant : François POULET
Laboratoire : ESIEA Pôle ECD

JURY Rapporteurs : Annie Morin Enseignant/Chercheur HDR IRISA, Rennes Gilles Venturini Professeur Université de Tours Examinateurs : Nadir Belkhiter Professeur Université de Laval, Québec Henri Briand Professeur Université de Nantes Anne-Marie Kempf Directeur ESIEA Pôle ECD François Poulet Enseignant/Chercheur HDR ESIEA Pôle ECD Djamel A. Zighed Professeur Université de Lyon 2

Table des matières

Dédicace

A Jasmine, A Duclaux, A mes parents, A toute ma famille.

Remerciements

Gloire soit tout d'abord rendue à l'Eternel mon Dieu par qui toute chose est possible.

Je remercie Mme Anne-Marie Kempf, directeur de ESIEA Pôle ECD pour sa gentillesse, son écoute et pour m'avoir soutenue dans ce travail.

Je remercie également M. Djamel A. Zighed et M. François Poulet pour avoir accepté de m'encadrer et pour m'avoir permis de mener à bien mes travaux de thèse.

Je remercie Mme Annie Morin et M. Gilles Venturini pour m'avoir fait l'honneur d'accepter d'être rapporteurs de ma thèse.

J'exprime ma gratitude à M. Henri Briand, M. Nadir Belkhiter, M. Djamel Zighed, Mme Anne-Marie Kempf et M. François Poulet pour avoir accepté de faire partie de mon jury de thèse.

Je remercie les membres du laboratoire ESIEA Pôle ECD pour les moments chaleureux et moins chaleureux passés ensemble. Je pense aussi au personnel de l'ESIEA-Ouest en général et plus particulièrement à Pamela SPINOSI, Gaëlle HUGUENIN et Suzanne.

Ces trois dernières années de ma vie ont été parsemées de doutes, d'hésitations, d'embûches. Des paroles réconfortantes, des encouragements, la présence de frères, amis et connaissances m'ont été d'une très grande utilité. Raison pour laquelle je tiens à dire merci à : Amédée, Amélie, Anne, Anne Marie, Anicet, Arouna, Bénédicte, Benoît, Berline, Brice T., Brice D., Chrispin, Christophe, Doris, Emile, Francine D., Francine N., Gilles, Gilles Thierry, Guy Hervé, Hilbert, Hondjack, Hortense, Jean, Jean-Claude, Laïka, Laurent, Marcellin, Marie-Pierre, Mathurin, Maurice, Michel, Narcisse, Patrice, Pascaline, Patience, Père Michel, Père Victor, Raoul, Raoul Serge, Roger, Samuel, Sarah, Sydonie, Thierry, Victor et Yves.

J'exprime ma profonde reconnaissance à : Mensah Noël Gounongbé, Léonard Jamfa, Eudoxie et Emmanuel Douya, Jocelyn et Sylvain Ngassa.

Je ne saurais oublier mes parents : Marie et Lucas Badjio, mes frères et sœurs : Duclaux, Jasmine, Carrel, Elvire, Sorelle et Annie, mes tantes : Sylvie, Sandrine, Jacqueline, Julienne et Helène, mes cousins : Celeste, Roméo, Mercémine, Monica et Patrick.

Résumé

Nos travaux s'inscrivent dans le domaine de la fouille visuelle de données (plus précisément en classification) et se fondent sur l'extraction de connaissances dans les données, l'apprentissage automatique, la qualité des interfaces et des logiciels, l'ergonomie des logiciels, le génie logiciel et l'interaction homme machine. L'évaluation de la qualité des modèles obtenus est basée la plupart du temps sur une estimation du taux de mauvaise classification. Cette estimation du taux de mauvaise classification est nécessaire mais pas suffisante pour l'évaluation de la qualité des outils de fouille visuelle de données. En effet, les outils et techniques de ce type utilisent des interfaces, des représentations graphiques, des ensembles de données et nécessitent la participation des utilisateurs finaux. Partant d'un état de l'art sur la visualisation, la fouille visuelle et la qualité des logiciels, nous proposons une méthode d'inspection experte et une méthode de diagnostic pour une analyse et une évaluation qualitative fine qui tient compte des spécificités du domaine abordé. Nous avons développé des guides de style et des critères de qualité pour l'analyse et le diagnostic des outils de fouille visuelle. Du point de vue des utilisateurs, afin d'utiliser les informations relatives à leurs profils et à leurs préférences tout au long du processus de fouille, nous avons aussi proposé un modèle de l'utilisateur final des outils de fouille visuelle.

Des études de cas menées avec la méthode de diagnostic proposée nous permettent de relever des problèmes autres que ceux résultant de l'estimation du taux de mauvaise classification. Ce travail présente aussi des solutions apportées à deux problèmes recensés durant l'analyse et le diagnostic des outils de fouille visuelle existants : le choix du meilleur algorithme pour une tâche de classification supervisée et le prétraitement de grands ensembles de données.

Nous avons considéré le problème du choix du meilleur algorithme de classification comme un problème de décision multicritères. L'intelligence artificielle permet d'apporter des solutions à l'analyse multicritères. Nous utilisons les résultats issus de ce domaine à travers le paradigme multi-agents et le raisonnement à partir de cas pour proposer une liste d'algorithmes d'efficacité décroissante pour la résolution d'un problème donné et faire évoluer les connaissances de la base de cas.

En ce qui concerne le traitement des ensembles de données de très grande taille, les limites de l'approche visuelle concernant le nombre d'individus et le nombre de dimensions sont connues de tous. Pour pouvoir traiter ces ensembles de données, une solution possible est d'effectuer un prétraitement de l'ensemble de données avant d'appliquer l'algorithme interactif de fouille. La réduction du nombre d'individus est effectuée par l'application d'un algorithme de clustering, la réduction du nombre de dimensions se fait par la combinaison des résultats d'algorithmes de sélection d'attributs en appliquant de la théorie du consensus (avec une affectation visuelle des poids).

Nous évaluons les performances de nos nouvelles approches sur des ensembles de données de l'UCI et du Kent Ridge Bio Medical Dataset Repository.

Mots clés : ECD, FVD, qualité, interaction, évaluation qualitative, ergonomie des logiciels, analyse des tâches et des utilisateurs, guidage des utilisateurs, système multi-agents, analyse multicritères, traitement de grands ensembles de données

Abstract

The research context of these works is the visual data mining domain and more precisely supervised data classification. Other related fields are: knowledge extraction in the data, machine learning, quality of interface, software ergonomic, software engineering and human machine interaction.

The result provided by a visual data mining tool is a data model. Generally, in order to access the quality of visual data mining tools, there is an estimation of the rate of bad classification. We believe that, this estimation is necessary but not sufficient for the evaluation of visual data mining tools. In fact, this type of tools use interfaces, graphical representations, data sets and require the participation of the end-users. On the basis of a state of the art on visualization, visual data mining and software quality, we propose two analysis and evaluation methods: an inspection method for experts and a diagnosis method which can be used by end-users for analysis and quality evaluation that takes account of the specificities of the treated domain.

We developed guidelines and quality criteria (measures and metrics) for the analysis and the diagnosis of the visual data mining tools. From the users' point of view, in order to use information relating to their profiles and their preferences throughout the mining process, we also proposed a user model of visual data mining tools.

Case studies performed with the proposed diagnosis method enable us to raise other problems than those resulting from the estimation of the rate of bad classification.

This work presents also solutions brought to two problems listed during the analysis and the diagnosis of some existing visual data mining tools: the choice of the best algorithm to perform for a supervised classification task and the pre-treatment of very large data sets.

We considered the problem of the choice of the best classification algorithm as a multi criteria decision problem. Artificial intelligence allows bringing solutions to the multi criteria analysis. We use the results coming from this domain through the multi-agents paradigm and the case based reasoning to propose a list of algorithms of decreasing effectiveness for the resolution of a given problem and to evolve knowledge of the case base.

For the treatment of very large data sets, the limits of visual approaches concerning the number of records and the number of attributes are known. To be able to treat these data sets, a solution is to perform a pre-treatment of the data set before applying the interactive algorithm. The reduction of the number of records is performed by the application of a clustering algorithm, the reduction of the number of attributes is done by the combination of the results of feature selection algorithms by applying the consensus theory (with a visual weight assignment tool).

We evaluate the performances of our new approaches on data sets of the UCI and the Kent Ridge Bio Medical Dataset Repository.

Key words: KDD, VDM, quality, interaction, qualitative evaluation, software ergonomic, task and user analysis, user guidance, multi-agents system, multicriteria analysis, very large data sets treatment

Introduction

Cadre de nos travaux : l'extraction de connaissances dans les données

Un programme techniquement efficace ne signifie pas qu'il est convivial pour les utilisateurs. Dans le cadre de cette recherche, l'idée est de s'assurer de la meilleure qualité des logiciels de FVD. Le travail abordé est à la croisée de plusieurs disciplines : l'extraction de connaissances dans les données (ECD) en général et plus particulièrement la fouille visuelle de données, l'intelligence artificielle, les interfaces homme machines, l'ergonomie des logiciels et les sciences sociales. Le domaine de l'ECD est né du besoin de découverte de structures (modèles, comportement) dans de masses de plus en plus importantes de données. En effet, des estimations montrent que la quantité de données disponibles à travers le monde s'accroît continuellement [Fayyad et Uthurusamy, 2002]. Ces données peuvent représenter des transactions de cartes de crédit, des appels téléphoniques ou des factures de supermarchés. Concrètement, en terme de chiffres, la page de statistiques du moteur de recherche GOOGLE [Google, 2005] par exemple estime à 250-300 millions le nombre de requêtes par jour pour plus de 8 milliards de pages indexées en mars 2005. En un seul jour, l'agence australienne pour le bien être reçoit sur son site plus de 11 millions de requêtes, le supermarché américain Walmart

[Domingos et Hulten, 2001] effectue plus de 20 millions de transactions de vente. Il serait très difficile voire même impossible de traiter cette masse d'information sans appui de méthodes automatiques, c'est le but de l'ECD. L'ECD peut être défini comme le processus non trivial d'extraction à partir de données de connaissances valides, inconnues, potentiellement utiles et compréhensibles [Fayyad et al., 1996]. Plusieurs phases de préparation, d'exploration et de traitement de ces données sont alors nécessaires. Plus précisément, l'ECD procède par plusieurs étapes parmi lesquelles nous pouvons citer la fouille de données (FD), [Kodratoff, 1996] et [Zighed et Rakotomalala, 2003].Les étapes en amont de la FD (figure 1) ont pour objectif de préparer les données, de les pré traiter. Suite à la FD, on obtient un modèle des données qui est évalué afin d'être considéré comme connaissance.

Figure 1 Processus d'extraction de connaissances dans les données

En amont de la fouille de données, on assiste à :

- la compréhension du domaine d'application : il s'agit d'expliciter la connaissance *a priori* et les buts à atteindre,
- la création d'un sous-ensemble cible des données (à partir de l'entrepôt) dans lequel appliquer la recherche. Les données se présentent alors sous la forme usuelle en statistique d'un fichier, observations ou unités statistiques en lignes dont chaque champ ou colonne contient les valeurs prises par les variables considérées,
- le nettoyage des données : il s'agit d'éliminer les erreurs, les données manquantes ou de traiter les valeurs atypiques,
- la transformation des données : cette opération consiste soit en une « normalisation », une linéarisation ou une compression.
- Après ces différentes étapes, la fouille de données proprement dite est opérée et comporte :
- l'explicitation de l'objectif et de la stratégie d'analyse : exploration, classification, discrimination, segmentation, recherche de singularités, modélisation, prévision,...
- le choix des méthodes, des algorithmes en privilégiant interprétabilité ou prédictibilité. Mise en oeuvre des outils informatiques appropriés pour aboutir à une modélisation.

En aval de la fouille de données, les opérations suivantes sont réalisées :

- les tests : sur la base de critères à préciser (qualité d'ajustement, de prévision, simplicité, visualisations graphiques...),
- la prévision,
- la diffusion de l'information pour une prise de décision.

La FD ainsi décrite peut se faire de façon automatique ou alors de façon interactive et

itérative. Le traitement automatique consiste en une « boîte noire » recevant en entrée des données prétraitées et fournissant en sortie des modèles ou le comportement des données pour les phases de post traitement.

Le traitement interactif et itératif ou fouille visuelle de données (FVD) implique beaucoup plus l'utilisateur qui participe activement à la construction du modèle de données. A cet effet, ce dernier utilise ses capacités humaines en reconnaissance de formes et le cas échéant, les connaissances du domaine des données.

Mais, la majorité des travaux de recherche en FD en général est consacrée au développement des modèles prédictifs des données [Kohavi, 2000] et à l'évaluation de la pertinence de ces modèles, donc au point de vue technique des outils. Nos travaux se situent dans le cadre plus spécifique de la FVD pour la classification supervisée des données. D'une part, prédire un modèle ou le comportement des données et comprendre ces prédictions peuvent ne pas aller de pair [Saporta, 2005]. Les outils de FVD en dépit de leur nécessité n'ont d'utilité que si les utilisateurs finaux acceptent de s'en servir. Il est donc primordial d'assurer leur performance et surtout leur convivialité. C'est la contribution principale apportée par ce travail qui s'appuie autant que possible sur une combinaison de stratégies et d'expertises.

Motivations

Dans le domaine de l'ECD, de nombreux efforts sont concentrés sur le développement des techniques optimales de découverte de corrélations, de motifs, de tendance ou de distribution des données. Parallélisme, incrémentation et distribution en sont quelques unes. On reconnaît une bonne méthode de fouille de données à sa capacité de [Domingos et Hulten, 2001] et [Han et Kamber, 2001] :

1. traiter de grands ensembles de données et des données de tout type en un temps constant,
2. requérir très peu de connaissances du domaine d'application,
3. occuper une place constante en mémoire quelque soit la quantité des données traitées,
4. créer un modèle des données en une seule lecture de ces données,
5. être paramétrable pour satisfaire à certaines contraintes,
6. s'appliquer à des données bruitées,
7. fournir un modèle de données quelque soit l'étape de traitement dans lequel il se trouve,
8. produire un modèle équivalent à celui susceptible d'être obtenu par n'importe quel algorithme de fouille ne respectant pas les contraintes spécifiées ci-dessus,
9. s'adapter aux variations des ressources interactionnelles.

En complément à cet ensemble de facteurs de qualité, [Inselberg, 1985], [Ankerst et al., 1999], [Ankerst, 2000] et [Poulet, 2002a] vont s'intéresser à l'exploitation des capacités humaines en reconnaissance de formes par les méthodes de fouille, on va alors parler de FVD. Les avantages de cette approche sont : 10.

- l'augmentation de la confiance et de la compréhensibilité des modèles conçus car les utilisateurs finaux participent à leur construction,
- l'utilisation des possibilités humaines en reconnaissance des formes,
- l'utilisation des connaissances du domaine des données durant le processus de fouille.

Cette valeur ajoutée en ce qui concerne la FVD se réfère beaucoup plus à l'implication des utilisateurs finaux qui permet de combiner l'énorme capacité de stockage des ordinateurs et leur capacité de calcul aux connaissances créatives des utilisateurs qui peuvent être flexibles et adaptables. 1.

En plus de l'optimisation en ECD, un autre aspect très étudié concerne l'évaluation du point de vue technique de ces outils. A cet effet, des techniques telles que la validation croisée, le holdout, le boostrap, etc. ont vu le jour. Ces techniques d'évaluation reçoivent en entrée des ensembles de données et servent à démontrer la qualité, la pertinence des résultats obtenus ainsi que leur nécessité. Les utilisateurs finaux ne sont pas pris en compte dans cette évaluation, il n'existe pas de critères permettant de mesurer la qualité d'utilisation et d'obtenir le point de vue (objectif ou subjectif) des utilisateurs. Pourtant, ils sont impliqués dans la boucle de fouille. Les facteurs d'optimisation (1-9) et d'évaluation des outils de FVD reposent donc essentiellement sur les aspects techniques du domaine de la fouille et sont beaucoup plus accessibles et manipulables par les développeurs, des spécialistes du domaine concerné. Il existe pourtant des facteurs de qualité visibles par les utilisateurs finaux. A l'état actuel des recherches, il s'avère difficile de jauger de l'acceptabilité de ces outils de FVD.

Qu'est ce qui se passera lors du transfert grandeur nature des techniques d'ECD techniquement fiables et efficaces des laboratoires à un contexte d'utilisation ? Dans leurs études, [Whiteside et al., 1988] et [Wolf, 1989] montrent que plusieurs produits dont les tests en laboratoire ont été satisfaisants ne fonctionnent pas une fois transférés dans un contexte réel d'utilisation.

Nous pensons que les différents facteurs de qualité technique existants constituent des conditions nécessaires mais pas suffisantes pour l'évaluation qualitative en ECD en général et plus particulièrement en FVD. Nous proposons des méthodes d'inspection experte et de diagnostic nécessaires pour ce faire dans le domaine de la FVD.

L'étude qualitative des outils de FD que nous proposons n'a pas jusqu'à présent constitué une forte préoccupation et très peu de travaux y sont dédiés. Pourtant, si les utilisateurs refusent d'utiliser les produits finaux, le temps consacré au développement des méthodes performantes de FD du point de vue technique serait vain. Il s'avère nécessaire de déterminer les caractéristiques d'acceptabilité des outils de FVD afin de

pouvoir les évaluer.

Toute la difficulté inhérente à une telle préoccupation consiste à définir des moyens d'analyse, d'inspection, de diagnostic et de sondage des utilisateurs des produits existants. A cet effet, il peut être intéressant de sortir du cadre de la fouille de données et de rechercher des fondements dans d'autres disciplines.

Après l'analyse et l'évaluation des outils de FVD, nos travaux s'étendent à la définition des solutions aux problèmes recensés durant les phases d'analyse, d'évaluation ou de diagnostic.

Contributions

Partant des réflexions épistémologiques, du rapprochement de la FVD aux autres domaines de recherche pourvus de connaissances expérimentalement valides en études qualitatives de logiciels, notre objectif a été dans un premier temps de limiter par des recommandations ergonomiques applicables par des experts les insatisfactions des utilisateurs afin d'éviter les rejets des produits finaux de FVD. Nous avons aussi procédé au développement des métriques de qualité pour l'analyse, l'évaluation ou le diagnostic des logiciels de FVD. L'utilisation de ces métriques peut requérir la participation des utilisateurs finaux et permet une analyse très fine de ces systèmes. Les buts visés par les différentes études qualitatives menées sont : l'amélioration de la productivité des utilisateurs, la diminution des erreurs commises par les utilisateurs, la diminution des coûts de prise en main des systèmes, la diminution du support technique aux utilisateurs. Suite au diagnostic par des utilisateurs des outils de FVD existants, nous avons développé des supports aux activités de fouille de données.

Ce travail s'articule autour de trois problématiques :

1. La première problématique abordée concerne une étude qualitative basée sur l'expertise du domaine de la FVD et sur les utilisateurs finaux qui aboutit à la définition d'une méthode d'inspection experte des environnements de FVD, d'une méthode d'analyse et d'évaluation des systèmes de FVD.
2. Le deuxième problème abordé concerne donc le diagnostic des systèmes de FVD par des utilisateurs finaux.
3. Ces deux premières problématiques se fondent sur l'étude des travaux en génie logiciel, en interaction homme machine, en ergonomie du logiciel et en sciences sociales. Le diagnostic de systèmes de FVD permet de relever de nombreux problèmes.
4. Le troisième point de cette problématique permet la conceptualisation et l'implémentation des moyens informatiques nécessaires à la résolution de ces problèmes de qualité, notamment en ce qui concerne le guidage des utilisateurs à travers différents choix à réaliser sur l'environnement de fouille et le traitement des ensembles de données de très grande dimension.

Ce mémoire est divisé en trois parties de deux chapitres chacune. La première partie porte au chapitre premier sur l'état de l'art de la visualisation et de la FVD. En effet, avant d'aboutir à la FVD proprement dite, les outils de FD ont tout d'abord intégré des modules de visualisation de données pour les phases de prétraitement et pour les phases de post-traitement des données. Les méthodes de visualisation permettent de représenter d'énormes quantités de données en même temps à l'écran. Les couleurs permettent aux utilisateurs de percevoir les similarités et les dissimilarités dans ces milliers de données qui doivent être arrangées de façon à pouvoir exprimer certaines relations [Keim et Kriegel, 1994]. Le chapitre 2 présente un état de l'art du domaine de la qualité des logiciels.

La deuxième partie se réfère aux contributions que nous apportons en ce qui concerne les études qualitatives en FVD. Plus précisément, au chapitre 3 nous proposons une méthode d'inspection experte des outils de FVD et au chapitre 4 un ensemble de métriques et de mesures de qualité pour une analyse, une évaluation ou un diagnostic des outils de FVD par leurs utilisateurs finaux. Une étude de cas est aussi présentée dans ce chapitre. Les conclusions tirées de l'étude de cas nous permettent d'introduire nos contributions en amélioration de la qualité des logiciels de FVD dans la troisième partie. En effet, le diagnostic nous a permis de constater que durant l'étape de construction du modèle de données, il existait de nombreux choix à opérer, notamment, le choix de la méthode d'analyse de données à exécuter. Le chapitre 5 présente une méthode pouvant servir de support pour ce choix. Le diagnostic a aussi permis de constater qu'il est impossible ou pénible de traiter des ensembles de données de très grande taille. Le chapitre 6 présente une méthode de prétraitement des données pour ce faire avant la conclusion et les perspectives.

Partie 1 : Etat de l'art

Chapitre 1 : Visualisation et fouille visuelle de données

1.1 Introduction

Ce chapitre présente un état de l'art des méthodes de visualisation et de fouille visuelle de données (FVD). En effet, les méthodes de visualisation d'informations et de données multidimensionnelles trouvent leur origine dans la statistique et les disciplines scientifiques. Initialement, les méthodes de visualisation de données en statistique étaient essentiellement en 2D voire 3D et statiques. Plus récemment, on a assisté à l'utilisation des graphiques dynamiques avec possibilité de manipulation directe. Dans cet état de l'art, nous partons des méthodes de visualisation statistiques pour aboutir à la visualisation en extraction de connaissances. Pour chaque méthode traitée, nous présentons ses avantages et inconvénients. L'objectif ici est de mieux situer notre contribution dans le domaine de la FVD.

Les travaux de [Keim, 1996], [Schneiderman, 1996], [Card et al, 1999] et [Chi, 2000] proposent un ensemble de méthodes de visualisation de données et d'informations.

L'idée principale en visualisation de données et d'informations est de tirer le meilleur parti de la vitesse de traitement et des capacités graphiques des ordinateurs, permettant ainsi aux utilisateurs d'interpréter des masses importantes d'informations ou de données. Par rapport à la représentation textuelle, la visualisation permet de se forger une idée sur un nombre beaucoup plus important de données à travers une seule représentation graphique visualisée en une seule fois à l'écran. Les utilisateurs se servent pour cela de leurs aptitudes en perception visuelle. Dans la section 1.1.1, nous présentons quelques fondements théoriques de la visualisation de données et d'informations. En section 1.1.2, nous faisons un état des tâches principales en visualisation. Puis, la deuxième section de ce chapitre est consacrée à une classification des techniques de visualisation. La troisième section quant à elle introduit la FVD et présente l'état de l'art de ce domaine.

1.1.1 Objectifs, types et propriétés de représentations graphiques

Bertin et Tufte sont les pionniers des méthodes de visualisation de données et d'informations [Bertin, 1967], [Bertin, 1977], [Tufte, 1993] et [Tufte, 1990]. Dans leurs travaux ils prônent l'utilisation de règles générales de disposition de données. Plus précisément, [Bertin, 1977] présente une classification des propriétés susceptibles d'être perçues dans une représentation graphique de données :

- l'association qui est représentée par la taille, la brillance, la texture, la couleur, l'orientation, la forme,
- la sélection, représentée par la taille, la brillance, la texture, la couleur et l'orientation,
- le classement, représenté par la taille, la brillance et la texture,
- la mesure quantitative représentée par la taille.

Selon l'auteur, la couleur dans un objet graphique est utilisée pour le codage de diverses catégories et pour la segmentation. Il recommande à cet effet l'utilisation de couleurs précises : le vert, l'orange, le bleu clair, le cyan, le magenta et un maximum de 10 à 15 couleurs par graphique.

Selon [Keim, 1996], les méthodes de visualisation de données peuvent être classées en 3 catégories selon leurs buts : l'exploration des données, la confirmation des hypothèses et la présentation des données/résultats.

Pour l'exploration des données : on suppose que l'utilisateur n'a pas de connaissance à priori sur les données. De manière interactive, il aboutit à des hypothèses sur les données comme résultat de l'exploration. Durant cette étape d'exploration des données, la mémoire de l'utilisateur est stimulée, il progresse dans sa compréhension des données (relations entre données, outliers, proximité, etc…). A cet effet, les relations entre les données doivent être compréhensibles et une information contextuelle doit être fournie à l'utilisateur.

En ce qui concerne la confirmation d'hypothèses sur les données, au début des traitements, l'utilisateur a une hypothèse sur les données. La représentation graphique de ces données lui permet de confirmer ou d'infirmer cette hypothèse.

Pour la présentation des données/résultats, il s'agit de pouvoir communiquer des

données ou des résultats de traitements de manière claire et succincte aux utilisateurs. Comme souligné précédemment, la visualisation de données permet de présenter une grande quantité d'information aux utilisateurs, elle permet aussi de communiquer des modèles mentaux présentant des idées abstraites [Oudshoorn et al., 1996].

La visualisation d'informations peut être métaphorique ou idiomatique. Dans une représentation métaphorique, les données sont présentées par l'intermédiaire d'objets du monde réel. En ce qui concerne la représentation idiomatique, les objets proposés par ce type de représentation graphique ne sont pas forcément issus du monde réel.

Il existe des méthodes de visualisation génériques, qui peuvent servir à la représentation de tout type de données après une phase initiale de prétraitement. Dans cette catégorie de méthodes, nous pouvons citer les matrices 2D et 3D [Chambers et al, 1983], les coordonnées parallèles [Inselberg, 1985] et [Inselberg, 1998], les bar-charts [Keim, 1999] et les survey plot [Rao et Card, 1994], [Lohninger, 1994].

En effet, les données représentées graphiquement peuvent être de natures différentes : numériques, symboliques, discrètes, continues, relations entre données, variations des données.

Après avoir représenté graphiquement les données ou les informations, il est possible de procéder à divers traitements sur ces données à travers des interactions homme machine. La section suivante fera le point sur les différentes tâches de la visualisation d'informations.

1.1.2 Tâches principales de visualisation

Une définition des tâches principales de visualisation d'informations peut être trouvée dans [Schneiderman, 1996]. Ces tâches comprennent :

- la vue d'ensemble qui permet d'obtenir une idée d'ensemble des informations contenues dans les données représentées graphiquement,
- la seconde tâche de visualisation est le zoom (focalisation). Le zoom permet de naviguer vers des points d'intérêt sachant qu'il existe des méthodes pour ce faire sans toutefois perdre le contexte,
- le filtrage quant à lui permet de cacher les éléments non intéressants selon des critères définis par l'utilisateur. Cette tâche permet de mettre en exergue des points tels que des anomalies, de masquer des détails sans rapport avec les objectifs des utilisateurs,
- il est aussi possible d'obtenir des détails à la demande sur une représentation graphique. Le détail à la demande permet de choisir un élément ou un groupe d'éléments afin d'obtenir les détails y ayant trait. Il s'agit d'une tâche importante dans le cadre de la visualisation à l'aide de matrices de scatter plot par exemple,
- la comparaison permet de voir les relations entre éléments,
- la tâche historique permet les raffinements successifs tout au long du parcours interactif des utilisateurs,

- la dernière tâche, l'extraction permet de sauvegarder et de conserver les résultats d'une séquence de présentation de données.
- Le principal avantage de ces différentes tâches de visualisation d'informations est qu'elles permettent une coopération effective entre l'utilisateur et le système tout au long de l'exploration des données et de la découverte de corrélations dans ces données.

Une classification des techniques d'exploration de données qui a été présentée par [Keim, 1996] fait l'objet de la section 1.2.

1.2 Classification des techniques d'exploration de données

Les méthodes d'exploration de données existantes peuvent être classifiées comme suit :

- les techniques de base,
- les techniques géométriques,
- les techniques symboliques,
- les graphes,
- les techniques hiérarchiques,
- la visualisation 3D,
- les techniques dynamiques.

Chacune de ces catégories de techniques de visualisation sera illustrée dans les sections suivantes.

1.2.1 Les techniques de base

Les techniques de basesont constituées de méthodes telles que :les camemberts (figure 1.1), les histogrammes 2D ou 3D avec leurs variantes (boxplot (figure 1.2), spinogram (figure 1.3)), graphes de y=f(x), les diagrammes en mosaïque, en barres et en bande.

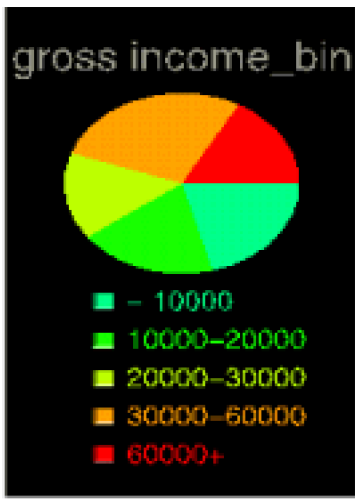


Figure 1.1 Camembert

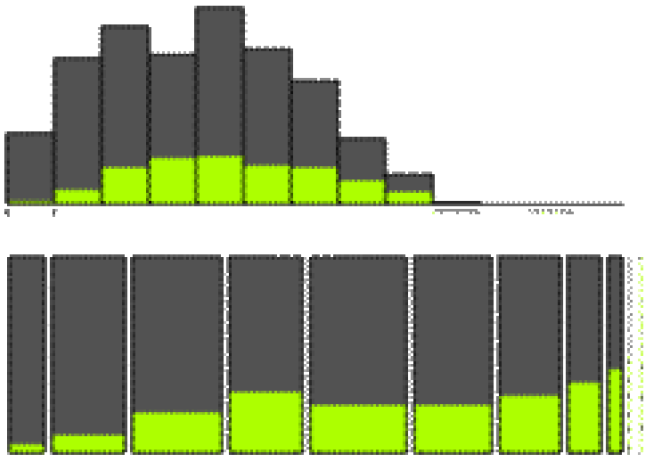


Figure 1.2 Spinogram

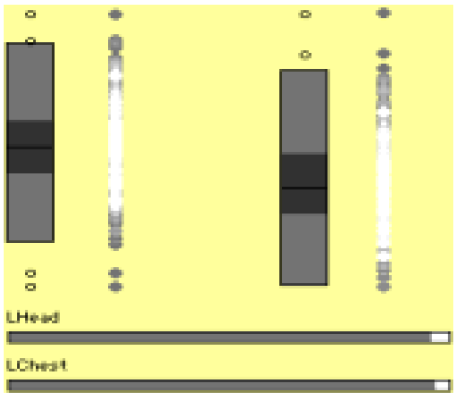


Figure 1.3 Boxplot

L'avantage majeur de ces techniques de base réside en leur compréhensibilité. L'inconvénient est qu'elles ne permettent de représenter que des relations simples ou simplifiées (pouvant être issues par exemple d'une agrégation de données de grande taille). Au final, la taille des données visualisées est relativement petite. Dans certains cas, les utilisateurs doivent pourtant comprendre et contrôler dans le détail des bases de données de plus en plus grandes et complexes. Il est nécessaire que les utilisateurs soient pourvus d'outils susceptibles de leur prêter main forte dans cette tâche.

1.2.2 Les techniques géométriques

Les techniques géométriques permettent une représentation graphique des données après transformations géométriques et projection. Comme exemples de techniques de ce type, nous pouvons citer les matrices de scatter plot 2D et 3D et les coordonnées parallèles qui seront décrites dans les paragraphes suivants. A ces techniques s'en ajoutent d'autres telles que le projection pursuit [Huber, 1985] ou le Grand Tour [Asimov, 1985].

1.2.2.1Matrices

Il s'agit d'une représentation sous forme matricielle de 2 variables. Une troisième variable est codée par la couleur (figure 1.4).

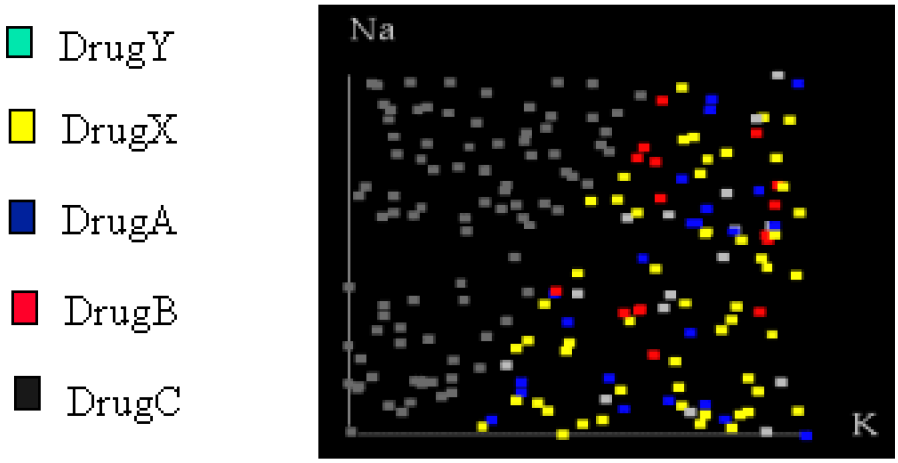


Figure 1.4 Matrice 2D

Des données pourvues de plusieurs variables peuvent utiliser la représentation matricielle de base en 2D : on parle de matrice de matrices de scatterplot (Figure 1.5). Dans cette représentation, chaque paire d'attributs est représentée par une matrice de scatter plot mettant ainsi en exergue leurs différentes corrélations.

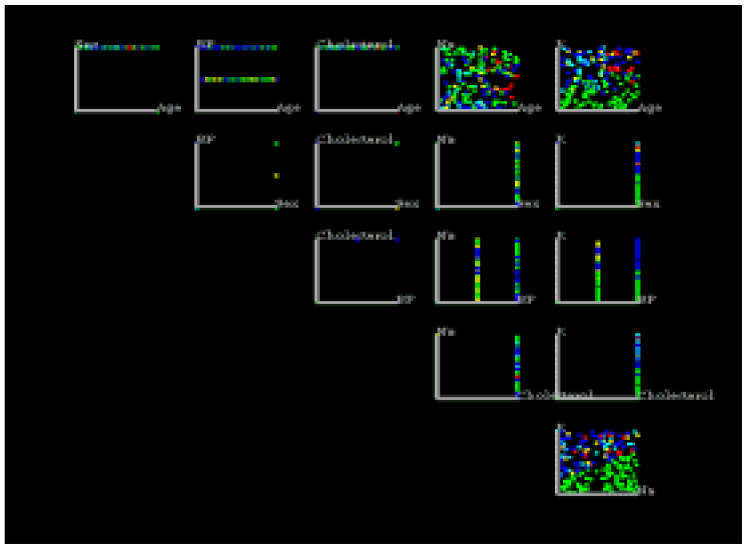


Figure 1.5 Matrices de matrices de scatter plot 2D

Ce type de représentation (figure 1.5) permet de découvrir de façon très explicite des corrélations entre les données. L'inconvénient majeur de la représentation en matrice de scatter plot 2D (figure 1.4) est que pour un ensemble de données pourvu de plusieurs attributs, il n'est pas possible d'avoir une vue d'ensemble de ces données. La matrice des matrices de scatter plot corrige cet inconvénient. En effet, la matrice 2D de représentation

de toutes les paires possibles d'attributs (figure 1.5) permet d'obtenir une vue d'ensemble des données à visualiser. Cette représentation est beaucoup plus intéressante. Le second inconvénient (pour les matrices de scatter plot) consiste en l'impossibilité pour cette approche de faire face au traitement des données décrivant un nombre important d'individus. A cet inconvénient s'ajoute l'impossibilité de traiter des données de grande dimension (nombre d'attributs et/ou d'observations) pour les matrices de matrices de scatter plot. En effet, l'espace disponible sur un écran ne permet pas de représenter correctement plus d'une vingtaine d'attributs pourvus ou non de nombreuses observations.

Il est possible de représenter une matrice de scatter plot en 3D. Par rapport aux structures en 2D, les structures 3D (figure 1.6) sont difficiles à mettre en œuvre car elles requièrent des processus significativement plus performants que ceux des structures 2D. Cependant, utilisé à bon escient, un graphique en 3D peut être extrêmement expressif.

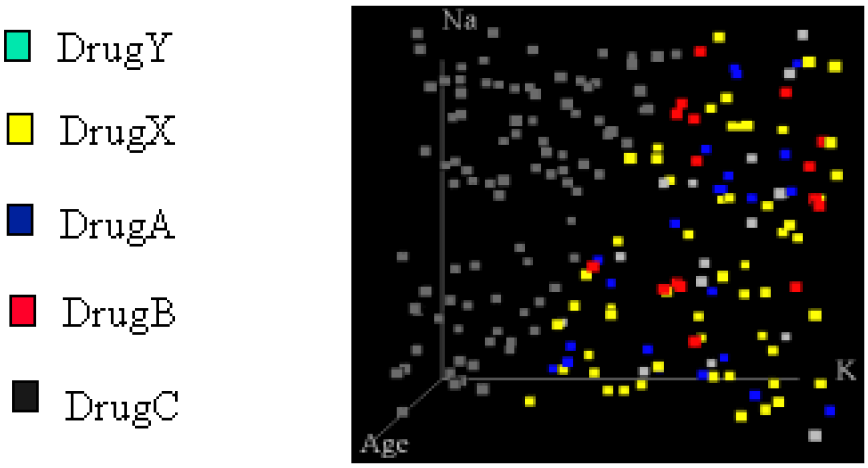


Figure 1.6 Matrice 3D

Dans ce type de représentation, tout comme pour les matrices de scatter plot 2D, se pose le problème de passage à l'échelle. La seconde limite est liée à l'impossibilité d'obtenir une vue d'ensemble des données. De plus, les matrices de scatter plot 2D et 3D ne traitent que des attributs numériques tout comme certaines versions des coordonnées parallèles qui font l'objet de la section suivante.

1.2.2.2Coordonnées parallèles

Les coordonnées parallèles [Inselberg, 1985] permettent de représenter les motifs et les corrélations contenues dans des données multi variées et multidimensionnelles par une représentation 2D.

Le but de cette méthode de visualisation est de représenter des données multidimensionnelles sans perte d'informations. Cette approche est très utilisée en fouille de données. La présentation de données multi variées par des coordonnées parallèles transforme la recherche de relations entre les variables en un problème 2D de reconnaissance de modèles.

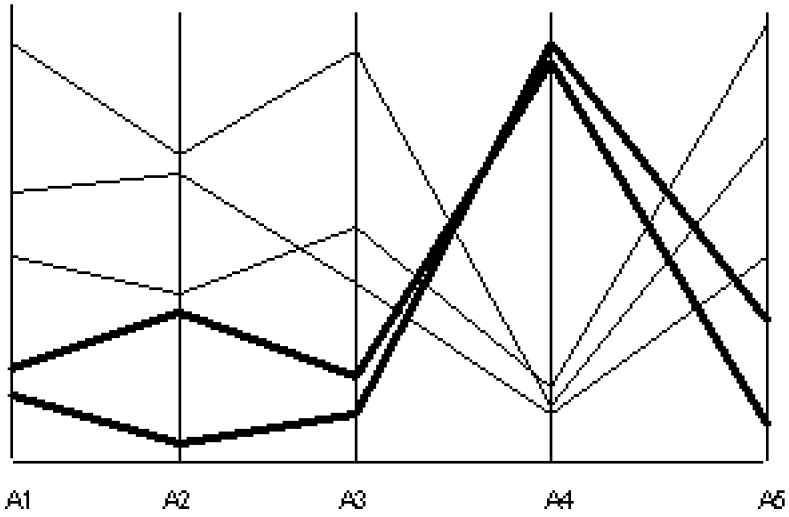


Figure 1.7 Coordonnées parallèles représentant 5 dimensions

La figure 1.7 présente une barre verticale par dimension. Un point dans un espace à N dimensions est représenté par une ligne polygonale. Ainsi, les corrélations positives et négatives entre les lignes sont facilement détectées. Il s'agit de la seule technique de représentation graphique de données vraiment multidimensionnelle. Il existe cependant une limite quant au nombre d'attributs et au nombre d'individus pouvant être représentés sur une même figure. Cette méthode nécessite aussi une bonne interface utilisateur pour le filtrage, le classement ou le marquage (figure 1.8). Bien qu'il s'agisse d'une technique puissante, l'apprentissage avec cette technique s'avère difficile. Dans les coordonnées parallèles, une importance primordiale est accordée à l'ordre des axes.

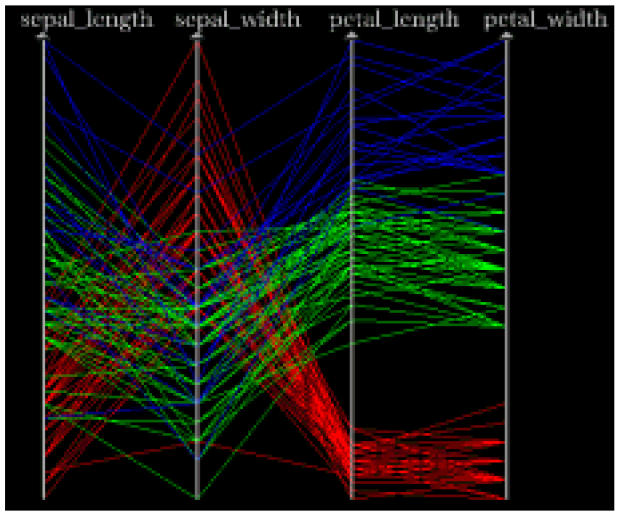


Figure 1.8 Coordonnées parallèles représentant 5 dimensions

1.2.2.3Conclusion

Le tableau 1.1 présente une synthèse des avantages et des inconvénients des méthodes de représentation graphique de type géométrique.

Tableau 1.1 Avantages et inconvénients des méthodes géométriques

Nom	Avantages	Inconvénients
Matrice de scatter plot 2D	Représentation explicite des corrélations entre deux attributs	Pas de vue d'ensemble des données, Passage à l'échelle non traité
Matrice de matrices de scatter plot 2D	Permet une vue d'ensemble des données	Existence d'une limite dans le nombre d'attributs et d'individus susceptibles d'être représentés Passage à l'échelle non traité
Matrice de scatter plot 3D	Plus expressif que les techniques 2D	Pas de vue d'ensemble des données Passage à l'échelle non traité
Coordonnées parallèles	Représentation multidimensionnelle en 2D sans perte d'information	Existence d'une limite dans le nombre d'attributs et d'individus susceptibles d'être représentés Importance primordiale dans l'ordre des axes Apprentissage difficile

L'objectif ici était de présenter quelques techniques de représentation graphique de type géométrique. Comme nous l'avons relevé à l'introduction de la section 1.2.2, il existe d'autres techniques géométriques. Il est à noter qu'il n'en existe pas qui soit meilleure que toutes les autres. Dans un système d'exploration de données, il s'avère nécessaire de fournir un catalogue de méthodes à l'utilisateur et lui prodiguer des conseils quant à la méthode la mieux indiquée pour ses besoins.

1.2.3 Les techniques symboliques

Les techniques symboliques représentent graphiquement les données sous forme d'éléments d'une icône ou d'une figure. Comme exemple, nous pouvons citer les faces de Chernoff [Chernoff, 1973] et les Stick figures [Pickett, 1970], [Pickett et Grinstein, 1988]. Les données sont représentées par exemple comme des vecteurs. Chaque champ de ce vecteur est une valeur d'un attribut. La figure 1.9 représente des exemples de vecteurs de ce type.

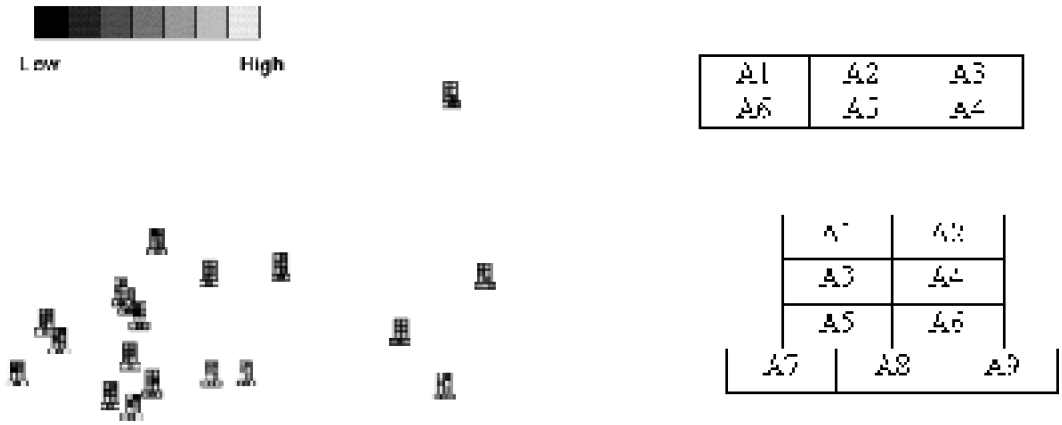


Figure 1.9 Représentation vectorielle de données

Malgré le fait que cette technique permet de représenter efficacement des données avec peu d'attributs, il s'avère impossible de l'utiliser pour des données pourvues d'un grand nombre d'observations. Dans ce cas, il est nécessaire de les agréger.

1.2.3.1 Faces de Chernoff

Les faces de Chernoff permettent de visualiser des données multi variées. Dans cette représentation, chaque structure faciale représente une variable particulière. Par exemple un attribut peut être codé par la forme de la bouche, un autre par la forme du nez, un autre par la couleur etc…

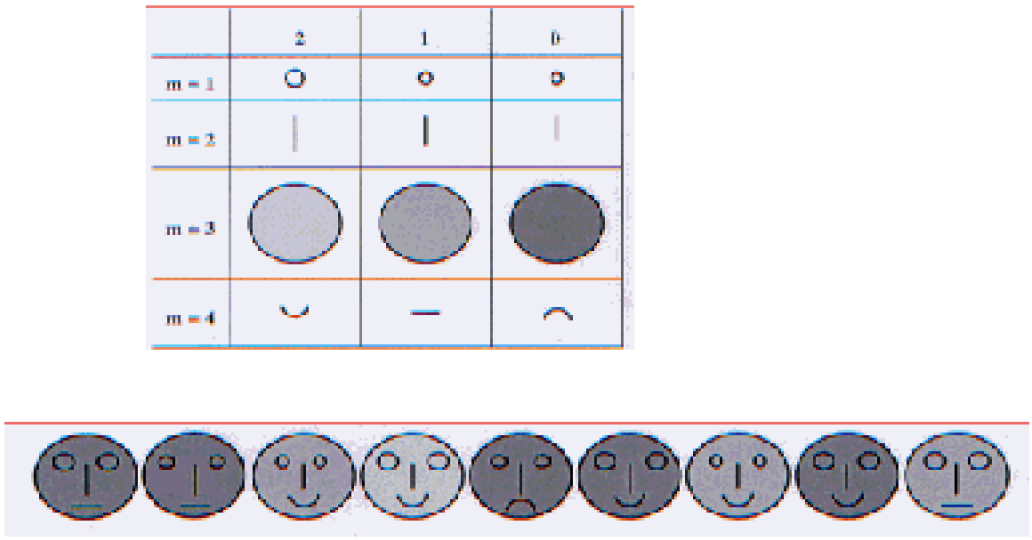


Figure 1.10 Codage des données par les faces de chernoff

L'avantage des faces de Chernoff est la possibilité de condenser les données, il s'en suit une facilité de compréhension pour l'usager. L'inconvénient majeur réside dans la subjectivité de l'affectation des expressions faciales aux différentes variables constituant les données et l'impossibilité de représenter plus d'une dizaine d'attributs.

1.2.3.2 Stick figure

La dernière technique de représentation graphique de type iconique que nous présentons est le diagramme en bâton ou stick figure.

La représentation graphique des données sous forme de stick figure est riche et surtout pratique du point de vue développement. Deux attributs de l'ensemble de données à traiter sont représentés par les coordonnées x et y de l'icône. Les autres attributs sont représentés par l'angle et/ou la longueur des segments (figure 1.11). Ce mode de visualisation était à l'origine constitué de 5 figures avec un angle de membre contrôlable.

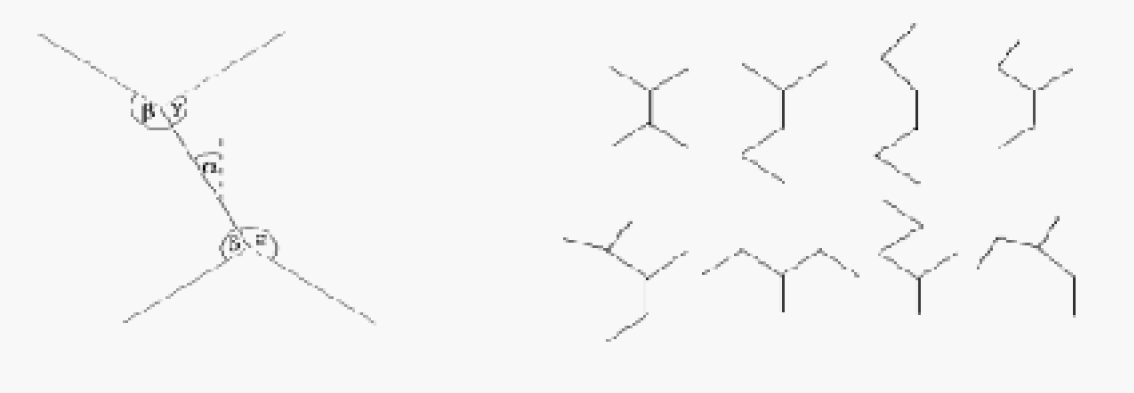


Figure 1.11 Codage de stick figures

La famille d'icônes de la figure 1.11 permet de représenter des données possédant 5 variables. Quatre variables peuvent être représentées par l'orientation de chaque membre de l'icône, 5 variables peuvent être représentées par l'inclinaison du corps de l'icône et les autres variables seront représentées par la longueur et la couleur des membres. L'application de la représentation de figures iconiques à un ensemble de données multidimensionnel aboutit à une représentation du type de celle de la figure 1.12. La notion de construction iconique permet ainsi de différencier des zones par leur texture dans une image complexe.

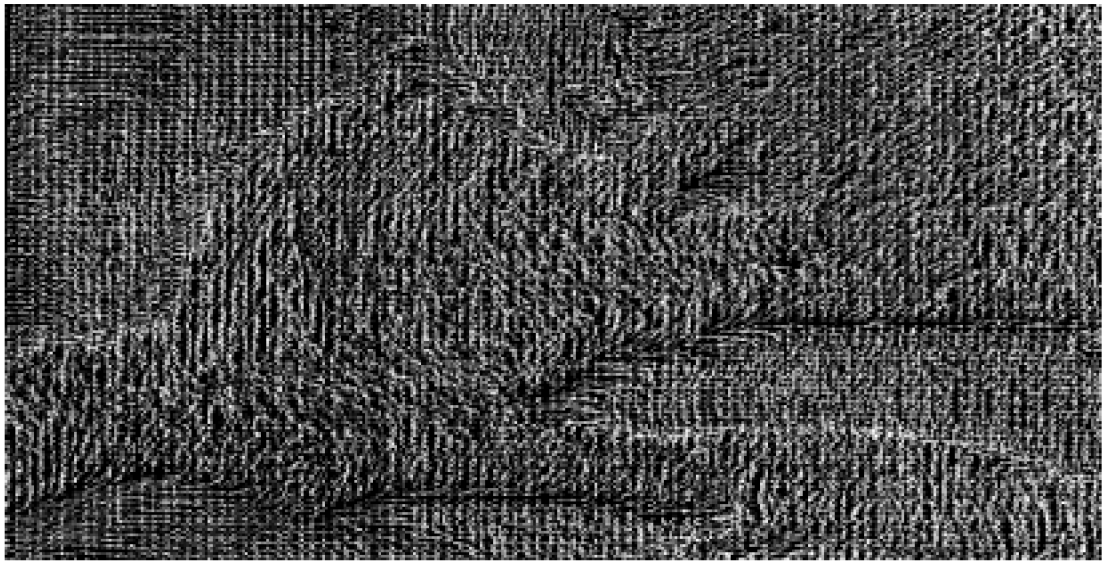


Figure 1.12 Représentation concrète des données avec un codage de stick figures

Cette technique présente une flexibilité au niveau des représentations graphiques qui peut s'avérer être un avantage et aussi un inconvénient. En effet, la distinction des corrélations dans la représentation finale dépend de l'affectation appropriée des paramètres des données et des paramètres visuels.

Tout comme les faces de Chernoff, les stick figures ne permettent de représenter qu'un nombre peu élevé de dimensions mais beaucoup d'individus.

1.2.3.3 Conclusion

Les différentes techniques symboliques présentées ne permettent de représenter que les données avec peu d'attributs. Dans ce type de visualisation de données, ainsi que dans les techniques géométriques et de base, les représentations graphiques sont statiques, il s'avère impossible d'y opérer les différentes tâches de visualisation qui dépendent d'informations contextuelles, décrivant en plus de la visualisation des données des propriétés de l'ensemble de données. Ces techniques ne servent donc qu'à présenter les données. Pourtant, pour être utilisable, un système de visualisation doit fournir assez de contexte pour permettre aux utilisateurs de se repérer et d'interagir. Cette notion de contexte est utilisée dans quelques unes des autres techniques qui font l'objet de la section suivante ainsi que la possibilité de traiter de grands ensembles de données

1.2.4 Autres techniques

1.2.4.1Techniques liées aux graphes : Fish Eye

Le Fish Eye [Furnas, 1986] permet une intégration de techniques de manipulation visuelle de données telles que le focus et le contexte dans une vue unique. Une fonction du niveau d'intérêt donne à chaque point dans la structure une valeur qui indique le niveau d'intérêt de l'utilisateur pour ce point pour une tâche donnée. L'intérêt décroît avec la

distance : si la distance au focus est grande, l'information ne sera affichée que si elle est intéressante.

En effet, cette représentation est organisée en niveaux et commence par un nœud racine qui donne naissance à différents nœuds fils. L'identité des nœuds à un niveau donné dépend de l'identité des nœuds aux niveaux précédents.

L'idée ici (figure 1.13) est que le nœud sur lequel on se focalise est agrandi et constitue le centre d'intérêt alors que les autres nœuds sont réduits, permettant ainsi de représenter un grand nombre d'informations tout en gardant le contexte.

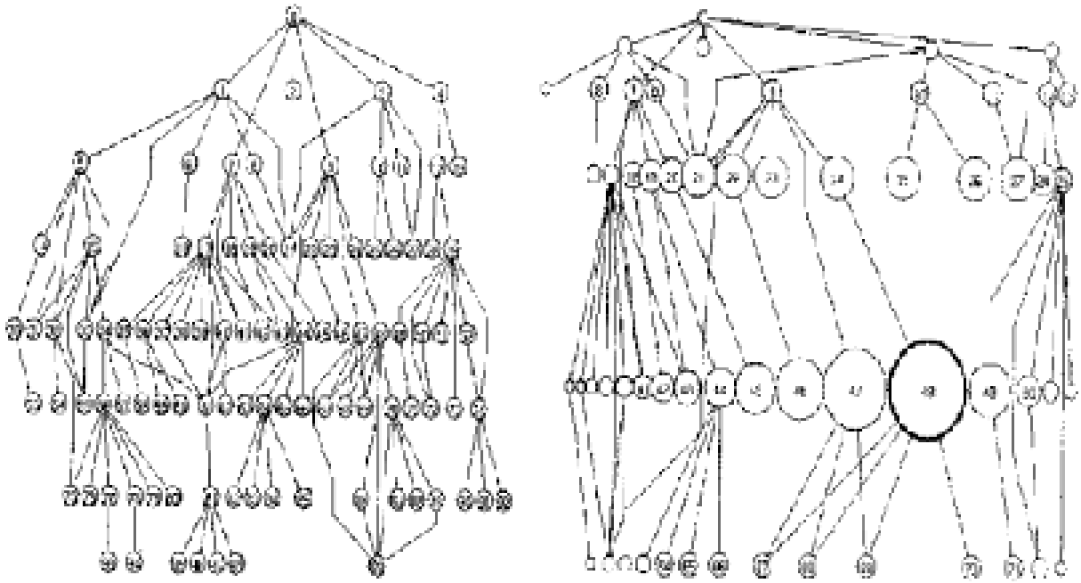


Figure 1.13 Représentation de fish eye

Plus explicitement, dans cette technique de visualisation, une fonction de distorsion exprimée par rapport à l'origine (point focal) est utilisée. Il est possible durant la navigation de gérer le niveau de détail. Les données représentées sont hiérarchisées, on a donc des réseaux contenant des sous réseaux.

Le principal avantage de cette technique de représentation de données est qu'elle permet de limiter l'encombrement visuel dans la représentation graphique d'une quantité importante de données. L'inconvénient est qu'elle ne permet pas de comparaison de deux éléments détaillés à moins qu'ils soient proches. Il faudrait pour ce faire deux déformations. Cette technique a par ailleurs une complexité élevée pour les graphes de grande taille.

1.2.4.2Techniques hiérarchique -3D : Cone Tree

Un cone tree [Robertson et al, 1991] est une représentation en 3 dimensions d'une structure arborescente (hiérarchie). La racine de l'arbre peut être représenté par un cube, une sphère ou un autre objet approprié. Les fils du nœud racine sont disposés près de la racine et constituent les racines des sous arbres. Contrairement aux autres méthodes utilisant la 3D, les arbres sous forme de cônes ont été directement développés en 3D, il

ne s'agit pas d'une généralisation de la 2D. Le cone tree offre des facilités de manipulation d'éléments visuels, permet de visualiser et de comprendre de grosses hiérarchies. Seules les informations potentiellement utiles sont présentées aux utilisateurs. Ainsi, la représentation simultanée des autres informations susceptibles de les distraire n'est pas faite. La charge cognitive des utilisateurs est ainsi réduite.

Tout comme la technique de fish eye, il n'est pas possible avec le cone tree d'obtenir une vue d'ensemble plus détaillée de plusieurs points à la fois et il existe aussi un problème de complexité pour les arbres de grande taille.

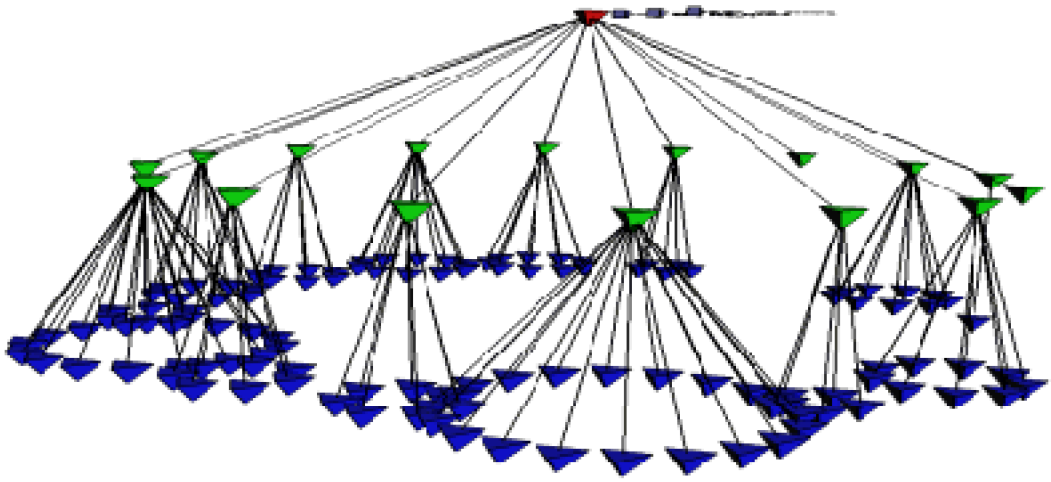


Figure 1.14 Représentation de cone tree

1.2.4.3Pixellisation [Keim, 1996]

L'idée de base des techniques orientées pixel est de représenter l'information par un point. On fait des correspondances entre valeurs des données et pixel d'écran et on arrange la disposition des pixels de façon adéquate. Nous détaillerons les techniques orientées pixel dans la section 1.3.2. La figure 1.15 présente un exemple de visualisation par une technique orientée pixel.

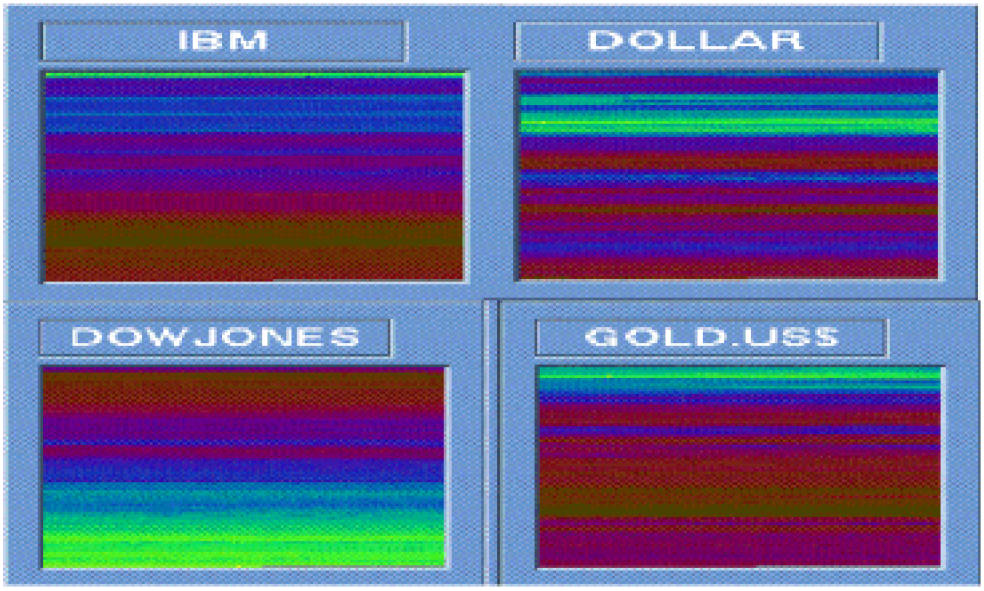


Figure 1.15 Représentation sous forme de pixel

1.2.4.4Data cube

Le data cube [Harinarayan et al., 1996] (figure 1.16) est un type de matrice multidimensionnelle qui permet d'explorer et d'analyser une collection de données suivant 3 perspectives en même temps. Les valeurs de chaque cellule d'un cube de données sont des mesures d'intérêt. Chaque cellule du cube de données est une vue comportant une agrégation d'intérêt.

En statistique, cette technique est un moyen efficace de représenter simultanément deux ou plusieurs caractères observés sur une même population.

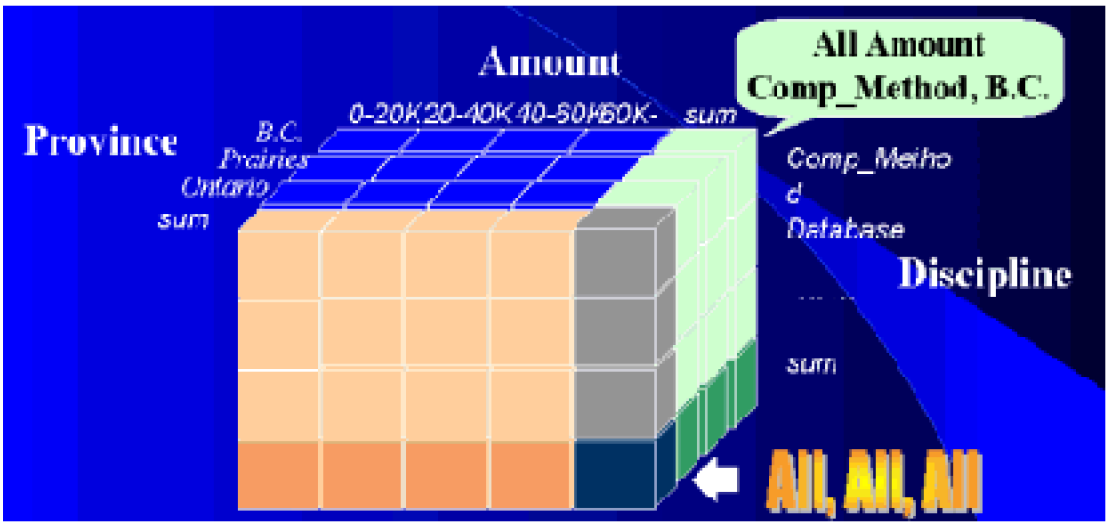


Figure 1.16 Représentation data cube

L'avantage de cette technique se réfère à la possibilité d'accès aux cellules du data

cube, permettant ainsi une navigation multidimensionnelle avec possibilité d'aggrégation, de détail, de sélection, de pivot. Les données sont alors visualisées suivant plusieurs perspectives. L'inconvénient de cette méthode réside en son espace de recherche trop grand.

Le tableau 1.2 présente une synthèse des avantages et des inconvénients des méthodes de visualisation présentée dans cette section.

Tableau 1.2 Avantages et inconvénients des autres techniques de visualisation

Nom	Avantages	Inconvénients
Fish Eye	Facilité de manipulation visuelle (focus, contexte, détail, etc.) Limitation de l'encombrement visuel	Complexité élevée pour de grands graphes Pas de comparaison possible pour deux éléments de la hiérarchie
Cone Tree	Facilité de manipulation visuelle Limitation de l'encombrement visuel	Complexité élevée Impossibilité d'avoir une vue détaillée de plusieurs points
Pixellisation	Possibilité de représentation de très grandes quantités de données	Perte de l'information relative à la distance entre les données dû au tri avant pixellisation Complexité liée aux opérations de tri
Data cube	Navigation multidimensionnelle	Espace de recherche élevé

L'objectif de cette première partie était de présenter différentes techniques de visualisation de données et d'informations. Ces techniques permettent de présenter et se faire des hypothèses sur les données en utilisant des propriétés différentes, avec possibilité ou non de manipulation directe. Certaines techniques de visualisation permettent de représenter des données et de construire un modèle de ces données, on parle de FVD. La section suivante est consacrée à la présentation de quelques unes de ces techniques.

1.3Fouille visuelle de données

1.3.1 Etat de l'art du domaine de la fouille visuelle de données

Les premiers travaux introduisant le terme de FVD datent de la fin des années 1990 [Keim et Kriegel, 1996], [Brunk et al., 1997], [Cox et al, 1997] et [Inselberg, 1998]. Dans un premier temps, toute l'attention a été portée vers le développement de techniques performantes et innovantes d'accès, de représentation graphique et de traitement des données. La masse de données disponible dans le monde ne cesse d'augmenter. La FVD utilise la visualisation comme canal de communication pour la découverte de corrélations dans les données. L'espace disponible sur un écran pour la représentation de ces données est limitée. Une première préoccupation concerne l'amélioration des techniques existantes de stockage, d'accès et de représentation graphique des données. Dans cet ordre d'idées, [Keim, 1996] a utilisé des techniques orientées pixel pour la représentation

des données multidimensionnelles (VisDB). L'outil segments de cercle de [Ankerst et al., 1999] utilise aussi le même principe. Un autre aspect abordé dans ce domaine a conduit à une modélisation de la tâche de FVD [Ankerst, 2000]. Le modèle de tâche de FVD présenté par Ankerst possède 3 variantes (figure 1.17) suivant le mode d'utilisation des représentations graphiques. Pour chacune de ces variantes, on dénote des phases de visualisation des données, d'application de méthodes d'analyse de données avant d'aboutir à la connaissance (modèle des données). En effet, lorsqu'on se sert de la visualisation comme support en fouille de données, après la sélection des données à exploiter (première étape des 3 variantes), une alternative se présente : soit l'utilisateur sélectionne et exécute un algorithme automatique de fouille de données, soit il procède à une visualisation (exploration) de l'ensemble de données. La visualisation peut être suivie de l'application d'une méthode automatique (ou interactive) de construction du modèle des données. L'étape suivante consiste en la visualisation des résultats. On assiste enfin à une évaluation puis à une exploitation de ces résultats considérés comme des connaissances nouvelles. Plus explicitement, les variantes de la figure 1.17 (modèle de tâche de Ankerst) correspondent à :

- l'utilisation de la visualisation pour l'interprétation des résultats d'un algorithme automatique de fouille de données,
- l'utilisation de la visualisation pour une exploration des données qui permet à l'utilisateur d'avoir une idée générale des données, suivi d'une application d'un algorithme automatique de fouille de données à l'ensemble de données et de l'utilisation de la visualisation pour une interprétation des résultats finaux,
- dans le dernier cas, la représentation graphique sert de support aux traitements. Afin de construire le modèle des données, l'utilisateur interagit avec la représentation visuelle et procède à des traitements successifs qui lui permettent de construire un modèle des données.

Nos travaux s'appliquent soit à toutes les variantes du modèle de Ankerst, soit à certaines de ces variantes. Pour chacune de nos contributions, nous préciserons la (les) variante(s) du modèle concernée(s).

Pour les deux premières variantes du modèle de Ankerst, la plupart de techniques de représentations graphiques utilisées ont été présentées en début de ce chapitre. Nous allons à présent dresser un état de l'art des différentes techniques de visualisation qui permettent de découvrir des corrélations dans les données de façon interactive (variante 3 du modèle de Ankerst), de construire un modèle de données et de représenter les résultats issus de la construction du modèle des données.

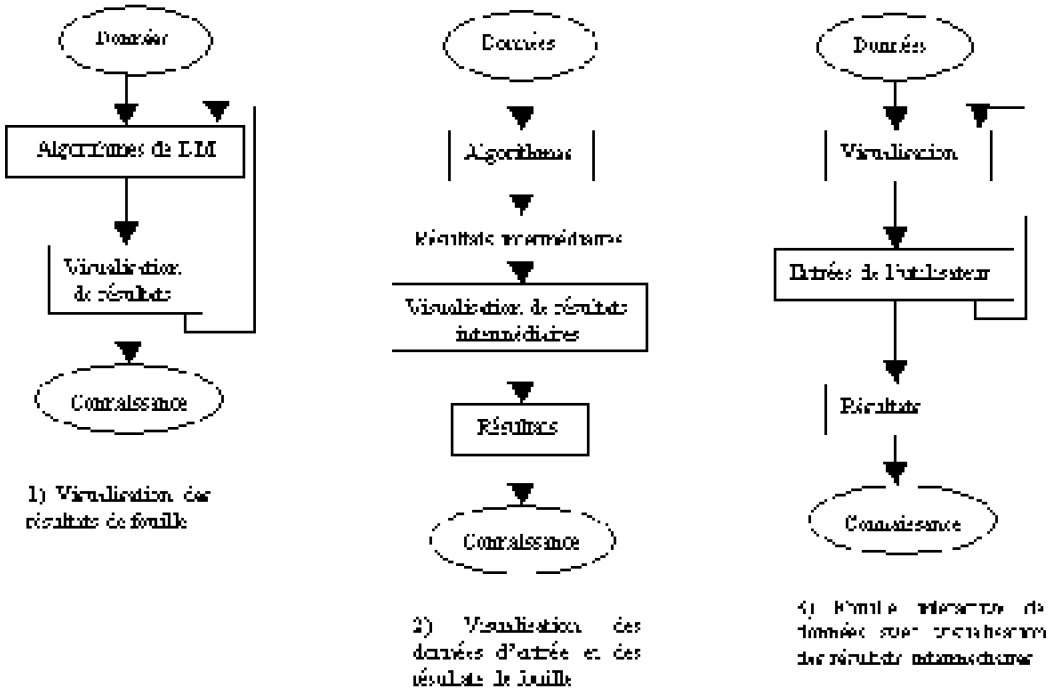


Figure 1.17 Fouille de données et visualisation

1.3.2 Techniques orientées pixel et construction d'un modèle de données

Les méthodes de visualisation qui utilisent cette technique sont : PBC [Ankerst, 1999], segments de cercle [Ankerst et Keim, 1996]) et DTViz [Han et Cercone, 2001].

1.3.2.1 Segments de cercle

Toute la base de données est représentée dans un cercle. Le cercle est divisé en segments, un segment pour chaque attribut. Dans le segment, chaque valeur d'attribut est représentée par une valeur de pixel. L'arrangement des pixels va du centre du cercle vers l'extérieur.

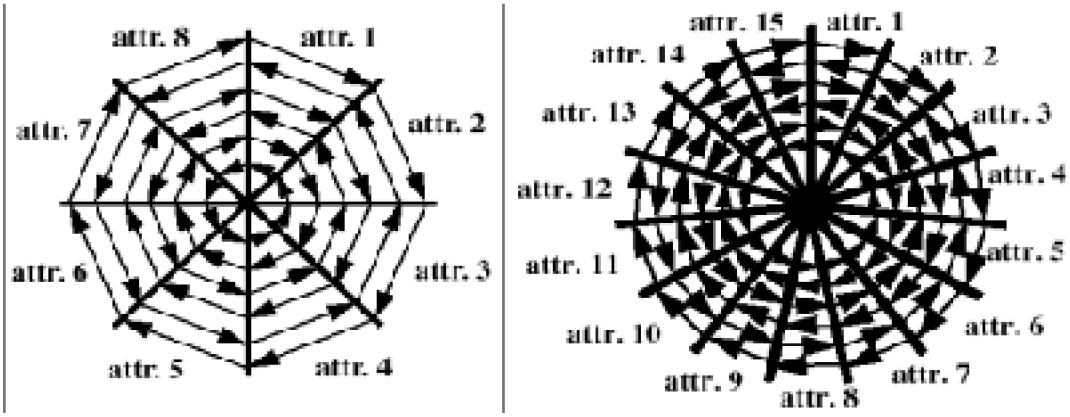


Figure 1.18 Représentation segments de cercles

Si on opère un changement dans le parcours utilisé pour remplir les pixels dans la représentation sous forme de segments de cercle, on obtient une représentation sous forme de barres rectangulaires.

Dans ces modes de visualisation, les données sont triées avant d'être affichées sous forme de pixels.

1.3.2.2 Barres rectangulaires

La figure 1.19 illustre la représentation en barres rectangulaires.

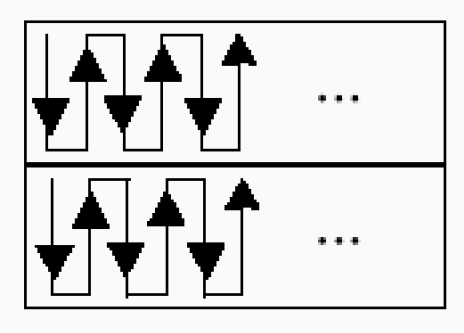


Figure 1.19 Représentation en barres rectangulaires

Segments de cercle et barres rectangulaires permettent de visualiser des ensembles de données relativement grands et d'appliquer l'algorithme PBC décrit dans le paragraphe 1.3.2.3. L'inconvénient de ce type d'approche tient au fait que suite au tri des données avant visualisation, l'information portant sur la distance des données est perdue. De plus, cette opération de tri constitue un coût supplémentaire dans les traitements par rapport aux techniques de représentation de type matrice de scatter plot.

1.3.2.3Perception-Based Classification (PBC)

PBC est un algorithme interactif de construction d'arbres de décision qui utilise le principe de la pixellisation. La phase initiale de PBC permet de représenter graphiquement l'ensemble de données d'apprentissage (segments de cercle ou barres rectangulaires) et d'initialiser un arbre de décision au nœud racine, correspondant à l'ensemble de données d'apprentissage.

De façon concrète, la représentation graphique d'un ensemble de données aboutira à la figure 1.20 par exemple.

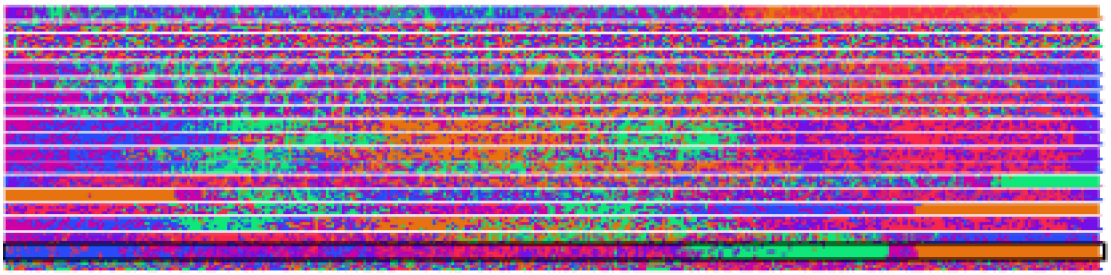


Figure 1.20 Exemple de représentation d'un ensemble de données

La visualisation permet de sélectionner de façon interactive des données et de procéder à des coupes. A partir de la représentation de la figure 1.21, pour la construction d'un arbre de décision par exemple, on peut procéder à des coupes interactives : binaires ou n aires mono variées. L'idée ici est de concevoir de façon interactive un modèle des données. A cet effet, on utilise la stratégie suivante : recherche de la meilleure partition pure, s'il n'en existe pas : recherche de la plus grande partition dominante, s'il n'en existe pas : recherche de l'ensemble de partitions dominantes.

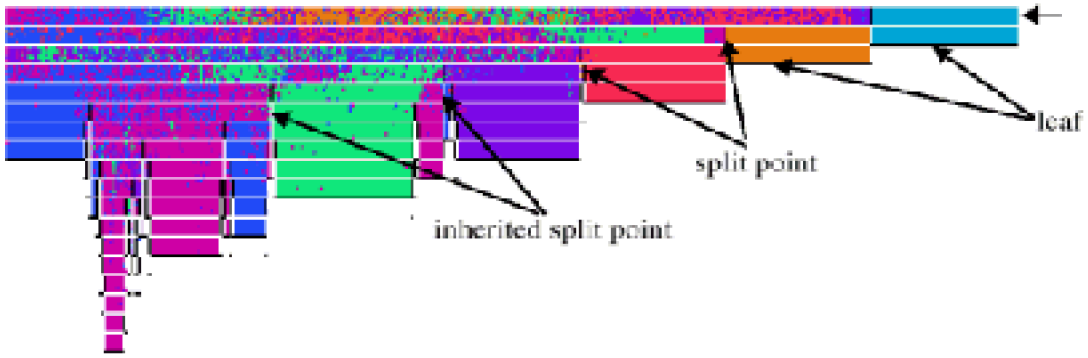


Figure 1.21 Représentation du processus de coupes successives

L'algorithme de construction interactive d'arbres de décision (CIAD) [Poulet, 2002b] et le module UserClassifier [Ware et al, 2001] de WEKA utilisent le même principe de construction d'arbres de décision mais sont basés sur des représentations matricielles. Les sections détaillent ces deux approches.

1.3.3Techniques matricielles et construction interactive d'un modèle des données

1.3.3.1 CIAD (Construction Interactive d'Arbre de Décision)

CIAD est un outil permettant la construction interactive d'arbres de décision. Cette technique utilise des matrices de scatter plot comme technique de visualisation et permet pour des ensembles de données avec un nombre de dimensions (n inférieur à 20) une projection de n*(n-1)/2 matrices. Pour n>20, une représentation par défaut de l'ensemble de données est fournie avec une combinaison de 20 attributs au maximum. La première étape de traitement consiste à représenter graphiquement l'ensemble de données à traiter. La figure 1.22 représente une vue de l'ensemble de données segmentation de

l'UCI [Blake et Merz, 1998] avec CIAD.

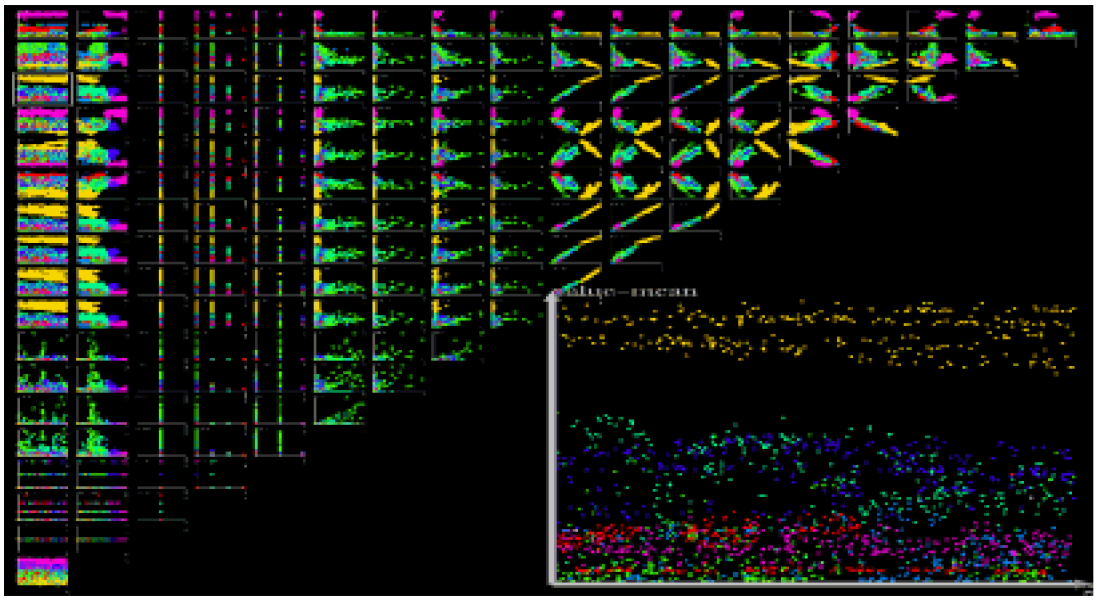


Figure 1.22 Représentation de l'ensemble de données segment avec CIAD

La couleur représente la classe. Les coupes effectuées pour la construction du modèle de données sont de type oblique en 2 dimensions donc sur deux variables. Ces différentes coupes sont effectuées grâce aux capacités humaines en reconnaissance de formes. Les étapes successives de ce traitement sont illustrées par la figure 1.23.

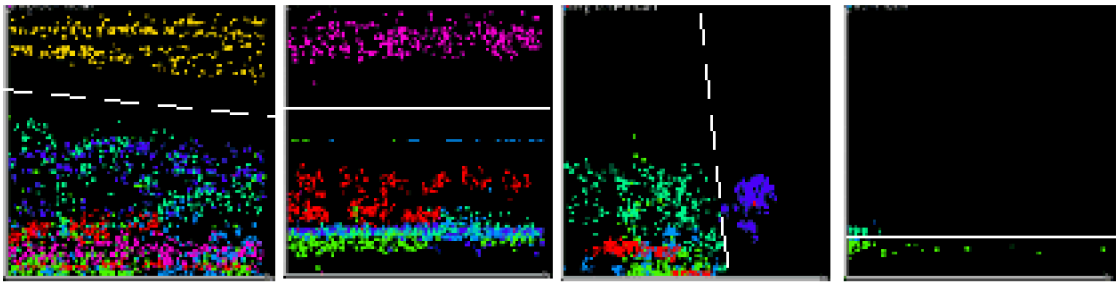


Figure 1.23 Construction interactive du modèle de données avec CIAD

Les 4 premières coupes représentent les classes 2, 7, 6, 3. Ces classes représentent 57% des individus de l'ensemble de données. CIAD peut être exécuté en modes 100% interactif, mixte ou alors 100 % automatique. Par rapport aux méthodes automatiques, CIAD permet d'obtenir une précision équivalente avec des tailles d'arbres inférieures.

1.3.3.2 UserClassifier de WEKA

Le module UserClassifier de WEKA est une implémentation de PBC qui utilise aussi des matrices 2D pour la construction interactive d'arbres de décision. UserClassifier à l'étape initiale de présentation de données ne permet pas d'avoir une vue globale de l'ensemble de données à traiter. Une seule matrice 2D est présentée l'écran. Pour le traitement de grands ensembles de données, la notion de contexte est perdue, il n'est pas possible de

visualiser toutes les paires possibles d'attributs en même temps à l'écran. Moyennant des efforts, il est possible d'accéder à toutes les paires de combinaison possibles d'attributs une par une, ce qui n'est pas le cas avec CIAD qui ne peut aller au-delà d'un certain nombre de dimensions (limite dûe à l'utilisation de la représentation sous forme de matrice de matrices de scatter plot). Les coupes opérées avec le module UserClassifier sont rectangulaires, polygonales ou alors sous forme de polylignes (voir la figure 1.24).

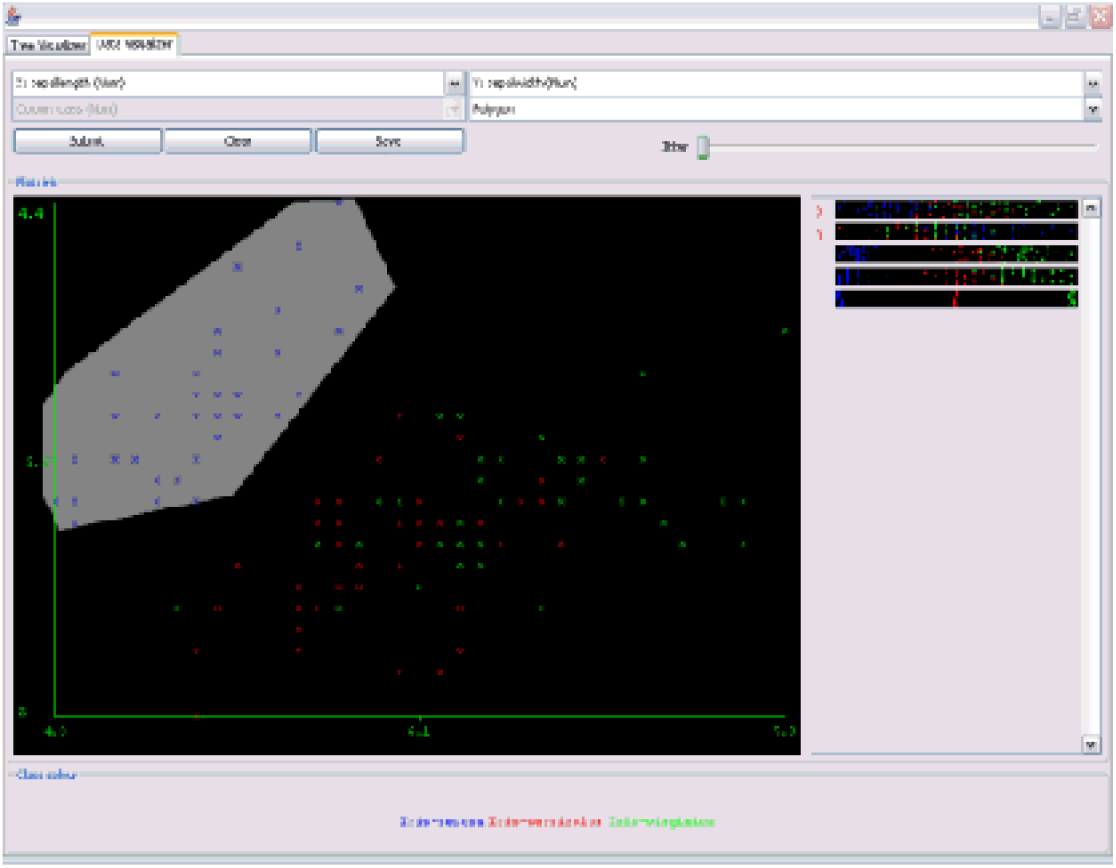


Figure 1.24 Représentation et construction interactive du modèle de données avec UserClassifer de WEKA

UserClassifier hérite de tous les inconvénients de la représentation graphique sous forme de matrice de scatter plot, notamment, l'impossibilité de traiter des ensembles de données pourvus d'un nombre élevé d'individus. De plus, il n'existe pas de mécanisme d'aide aux utilisateurs durant la construction du modèle des données.

Les différentes variantes du modèle de Ankerst montrent aussi que l'étape de construction du modèle de données est suivie d'une étape de post traitement au cours de laquelle l'utilisateur peut avoir recours aux techniques de visualisation. A cet effet, il existe des techniques de visualisation telles que CUBEVIS [Poulet, 2001] ou Grand tour [Asimov, 1985]. Nous nous limitons ici à l'étape de construction du modèle de données.

Le modèle de tâche en FVD montre que la construction du modèle de données peut se faire de façon automatique (deux premières variantes du modèle de Ankerst) ou alors de manière interactive (troisième variante du modèle de Ankerst). Cet état de l'art est

essentiellement basé sur la construction interactive du modèle de données qui possède de nombreux avantages par rapport aux algorithmes automatiques couplés ou non aux méthodes de représentation graphique. L'exécution des algorithmes automatiques d'analyse de données nécessite une étape préalable de paramétrage, ce qui n'est pas le cas en construction interactive du modèle de données. En effet, l'algorithme automatique se comporte comme une boîte noire recevant en entrée des données et fournissant en sortie un modèle de ces données. L'utilisateur ne participe pas à la construction de ce modèle, ce qui pourrait avoir une incidence sur le degré de confiance qu'il accordera au résultat. La vision humaine peut servir à capturer des corrélations complexes dans les ensembles de données au travers de représentations graphiques. Si l'utilisateur de l'outil de fouille interactive de données est un spécialiste du domaine des données, il peut utiliser ses connaissances du domaine de données durant le processus de fouille et non seulement au moment de l'interprétation des résultats (cas des algorithmes automatiques). La confiance au modèle de données ainsi construit est élevée car l'utilisateur a participé à sa construction. Le temps de traitement avec l'algorithme interactif peut s'avérer long, surtout pour de grands ensembles de données.

1.4Conclusion

Partant des méthodes de représentations graphiques pour les statistiques, nous avons introduit la fouille visuelle de données dans ce chapitre. L'objectif était de mieux situer le contexte des travaux qui seront explicités dans la suite de ce document.

En effet, dans un environnement de FVD, les données peuvent être représentées graphiquement soit par un schéma, soit par une combinaison de schéma pour une vue globale des données. La FVD (toutes les variantes du modèle de Ankerst) permet d'explorer et d'interagir (3^{e} variante du modèle de Ankerst) avec les données afin de découvrir les connaissances cachées dans ces données. Les recherches dans le domaine de la FVD sont concentrées sur la mise au point de techniques performantes et innovantes. Mais les domaines connexes de ces recherches ne sont pas trop explorés par exemple la conception centrée utilisateur. Ceci pourra avoir comme conséquence de nombreux problèmes relatifs aux besoins spécifiques des utilisateurs. Afin de pallier à ce problème, nous nous intéressons à la qualité des logiciels de FVD et au guidage des utilisateurs.

Par exemple le module UserClassifier de Weka ne dispose d'aucun mécanisme d'aide. En cas d'erreur, l'utilisateur doit reprendre ses différents traitements à l'état initial.

Le chapitre 2 présente à cet effet un état de l'art sur la qualité externe et interne des logiciels.

Chapitre 2 : La qualité des logiciels

2.1 Introduction

L'état de l'art du domaine de la FVD montre que très peu de travaux concernent l'étude qualitative des outils de FVD. Selon l'AFNOR (NF X50-120), la qualité est l'aptitude d'un produit ou d'un service à satisfaire les besoins des utilisateurs. Ce chapitre introduit, présente et illustre la nécessité d'une amélioration de la qualité des logiciels de FVD. Notre démarche trouve ses fondements d'une part dans la crise du logiciel qui a engendré le génie logiciel (GL) et d'autre part dans l'ergonomie des logiciels, l'étude des facteurs humains et les IHM. En effet, suite à la crise du logiciel caractérisée par une augmentation des coûts de production des logiciels, une difficulté de leur évolution, leur non fiabilité, le non respect des spécifications et des délais, le génie logiciel a été créé. Ce domaine de recherche a ainsi vu le jour entre le 7 et le 11 Octobre 1968 a Garmisch-Partenkirchen, sous le parrainage de l'OTAN. L'objectif du GL est d'optimiser les coûts de développement du logiciel améliorant ainsi la qualité des produits finaux. En GL, divers travaux ont mené à la définition de la qualité des logiciels en terme de facteurs qui dépendent du domaine d'application et des outils utilisés. La qualité d'un logiciel peut être influencée par sa portée, la tâche de conception et les hommes. En ce qui concerne le facteur humain, il est nécessaire de comprendre les attentes et les besoins des utilisateurs finaux. Divers champs de recherche proposent des méthodes d'assurance qualité des logiciels. La non qualité peut entraîner de nombreuses erreurs à l'issue de l'étape de conception des systèmes informatiques. Il est beaucoup plus difficile de corriger des erreurs à l'issu de l'étape de conception. L'objectif de ce chapitre est de faire un tour d'horizon des méthodes de vérification de la qualité des logiciels interactifs afin de mieux situer les apports de notre contribution décrite dans les chapitres 3 et 4 de la deuxième partie. Après une présentation de la problématique de nos travaux, la troisième section de ce chapitre définit et décrit très explicitement les différents facteurs de qualité des logiciels. Suite à cette section, un état de l'art concernant les études qualitatives en FVD est présenté. Partant de cet état de l'art, l'accent est porté sur divers domaines de recherche présentant des approches d'études qualitatives des logiciels et pour conclure nous introduirons la nécessité d'une analyse de la situation de travail en FVD.

2.2 Problématique

Les utilisateurs finaux influencent le développement des produits. Si le produit final correspond à leurs besoins, envies et caractéristiques, il sera certainement adopté et utilisé. En effet, efficacité fonctionnelle, facilité d'acquisition d'informations, vitesse de navigation, efficience de navigation, vitesse d'entrée de données, satisfaction subjective ou facilité d'apprentissage constituent des critères de préférence des logiciels par les utilisateurs. Très peu d'études statuent sur ces facteurs externes de qualité en FVD. Pourtant, le développement et l'utilisation des techniques de FVD, interactives, pose le problème de l'interaction homme machine en général et plus particulièrement les problèmes d'application de recommandations ergonomiques et de psychologie cognitive. En effet, les systèmes interactifs permettent de développer des capacités techniques et cognitives chez les utilisateurs. La technologie de l'outil, c'est à dire les paramètres ou

caractéristiques de l'interaction, fait partie des capacités techniques. Les capacités techniques doivent être prises en compte tout au long du cycle de conception de l'outil, de son interface avec l'utilisateur, afin d'en faciliter l'usage, c'est à dire sa capacité de contrôle par l'utilisateur. Les propriétés cognitives font référence en général au comportement de l'utilisateur face à la tâche qui utilise la technologie pour être réalisée. Il existe des propriétés cognitives propres aux méthodes de visualisation en fouille de données. Ces propriétés sont étroitement dépendantes des propriétés technologiques. Elles sont perçues par l'utilisateur qui se forge des croyances sur la technologie utilisée. Concrètement, nous pouvons citer comme exemples illustratifs de ces différentes propriétés des facteurs subjectifs et/ou objectifs tels que : la cohérence visuelle, la détection de la position courante des traitements, la simplicité de l'environnement, la définition claire des fonctionnalités ou l'utilisation d'un langage compréhensible.

L'un des problème traité ici concerne donc l'analyse et l'évaluation des systèmes de FVD afin d'accéder à leur qualité et d'élaborer des recommandations pour le développement d'outils de ce type. A la suite de cette section, nous allons définir plus amplement la qualité du logiciel.

2.3 Définition de la qualité du logiciel

Pour [IEEE 610.12, 1990], la qualité est la mesure dans laquelle un système, une composante ou un processus répond aux besoins ou aux attentes d'un client ou d'un utilisateur, aux exigences énoncées et implicites de tous les intervenants.

Le logiciel est issu d'un cycle comprenant une analyse des besoins, une spécification de fonctionnalités, une mise en œuvre et une phase de tests. Pour assurer sa qualité, il est nécessaire d'en connaître les facteurs. Il existe deux facteurs de qualité des logiciels : la qualité externe et la qualité interne.

2.3.1 La qualité externe du logiciel

La qualité externe est relative au point de vue utilisateur et concerne tout d'abord les fonctionnalités offertes par le logiciel. Il s'agit de voir dans quelles mesures le produit final satisfait aux spécifications. Il s'agit aussi de jauger les réactions du logiciel aux imprévus ainsi que sa fiabilité. Le second aspect concernant la qualité du logiciel du point de vue utilisateur est le critère de performance qui est caractérisé par la qualité des algorithmes utilisés, l'ergonomie et la facilité d'utilisation.

2.3.2 La qualité interne du logiciel

La qualité interne est relative au point de vue développement. L'attention est focalisée sur les programmes sources, le langage de programmation choisi, la modularité du code obtenu, l'évolution du logiciel. La qualité du point de vue développement est fonction de l'indépendance du produit final en ce qui concerne le genre d'environnement d'exécution et l'interopérabilité (la capacité à interagir avec d'autres applications).

2.3.3 Conclusion

La qualité des logiciels définie et décrite sommairement dans cette section possède de nombreux attributs. Le paragraphe suivant définit plus amplement ces différents attributs de qualité des logiciels.

2.4 Description détaillée des facteurs de qualité du logiciel

Les caractéristiques de la qualité du logiciel (facteurs externes et internes) peuvent être regroupées en quatre grandes catégories : l'utilité, l'utilisabilité, la maintenabilité et la portabilité.

2.4.1 Utilité

L'utilité fait référence à l'adéquation fonctionnelle et regroupe les caractéristiques suivantes : fiabilité, efficacité et aspect humain. La fiabilité ou robustesse du logiciel peut être définie comme étant la probabilité que le logiciel se comporte tel qu'espéré durant un instant donné, c'est aussi l'aptitude du logiciel à fonctionner dans des conditions anormales. Il s'agit plus explicitement de la précision des traitements, de l'intégrité du logiciel c'est-à-dire l'aptitude du logiciel à protéger son code et ses données contre des accès non autorisés.

L'efficacité fait référence à l'utilisation optimale des ressources matérielles : économie de mémoire, rapidité d'exécution. Les aspects humains regroupent la facilité d'apprentissage, la facilité d'utilisation, de préparation des données, la documentation fournie à l'utilisateur, l'accessibilité, la facilité d'interprétation et de correction des erreurs.

2.4.2 Utilisabilité

Le grand dictionnaire terminologique de l'Office de la langue française définit l'utilisabilité comme étant : «la qualité d'un matériel ou d'un logiciel qui est facile et agréable à utiliser et à comprendre, même par quelqu'un qui a peu de connaissances en informatique».

La norme [ISO 9241-11, 1998] quant à elle, définit l'utilisabilité comme le degré selon lequel un produit peut-être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité, efficience et satisfaction, dans un contexte d 'utilisation spécifié.

Un produit est un objet fabriqué en grande série, ayant une utilité fonctionnelle, s'inscrivant dans une activité pour permettre à son utilisateur appartenant à un certain public d'atteindre certains objectifs, le producteur ne pouvant plus intervenir sur l'objet diffusé pour l'adapter aux conditions spécifiques d'utilisation.

L'efficacité invoque la précision et l'intégralité avec lesquelles des utilisateurs donnés peuvent atteindre des buts donnés dans des environnements particuliers.

L'efficience fait allusion aux ressources déployées en fonction de la précision et de l'intégralité des buts atteints.

La satisfaction fait référence au confort et à l'acceptabilité du système pour ses utilisateurs et pour les personnes qui sont affectés par le système.

2.4.3 Maintenabilité

Un logiciel non maintenable rend difficile l'ajustement de son interface utilisateur par exemple. La maintenabilité regroupe la facilité de vérification, la clarté et la facilité d'adaptation. La facilité de vérification concerne la facilité de préparation des tests, l'auto documentation, la possibilité de vérifications formelles statistiques et la structuration. La clarté concerne la structuration, la concision et la lisibilité du code source. La facilité d'adaptation regroupe l'extensibilité, la structuration et la documentation technique. L'extensibilité fait référence à la facilité avec laquelle un logiciel se prête à une modification ou à une extension.

2.4.4 Portabilité

La portabilité concerne la facilité avec laquelle un logiciel peut être transféré sous différents environnements matériels et logiciels. Ce facteur de qualité fait aussi référence à l'utilisation d'un langage standardisé, à l'indépendance du matériel et à l'indépendance du système d'exploitation.

2.4.5 Conclusion

L'utilisabilité et l'utilité font l'objet d'étude des domaines de recherche tels que l'ergonomie des logiciels, l'étude des facteurs humains ou les interfaces homme machine. En ce qui concerne la portabilité et la maintenabilité, elles concernent l'étape de test et de maintenance de la discipline génie logiciel.

A ce niveau, on pourrait se poser la question de savoir pourquoi cet intérêt pour la qualité des logiciels. En effet, il arrive que les logiciels conçus ne répondent pas aux besoins des utilisateurs finaux ou que ces logiciels contiennent beaucoup d'erreurs. Le paragraphe suivant décrit quelques exemples illustratifs.

2.5 Conséquences de la non qualité des logiciels

La qualité des systèmes se résume par leur facilité d'utilisation, d'apprentissage, la vitesse de navigation, la capacité pour les utilisateurs à travailler sans faire des erreurs et leur satisfaction. La non qualité des systèmes informatiques a engendré des pertes inestimables. Nous pouvons citer par exemple l'explosion d'Ariane 5 qui a coûté un demi milliard de dollars, le 4 juin 1996. Cette explosion était due à une erreur dans une composante dont le fonctionnement n'était pourtant pas indispensable durant le vol [Jézéquel et Meyer, 1997]. Plus récemment en France, le 17 octobre 2004 suite à une panne de deux serveurs d'acheminement d'appels, l'opérateur téléphonique Bouygues Telecom n'avait aucune couverture réseau pendant 15 à 48h. Le système de réservation SOCRATE de la SNCF à plantages répétitifs ou la perte de la sonde Mariner vers Vénus à cause d'une erreur de programme FORTRAN en sont d'autres exemples. Il s'avère donc nécessaire de s'assurer de la qualité des systèmes informatiques.

Dans le cadre de nos travaux, nous nous intéressons aux facteurs de qualité du

logiciel visibles par les utilisateurs. De ce point de vue, deux critères résument la qualité des logiciels, il s'agit de l'utilité et de l'utilisabilité (facilité d'utilisation). L'utilité fait référence aux fonctionnalités du logiciel tandis que l'utilisabilité concerne les usages. L'utilité et l'utilisabilité constituent les deux principales dimensions de l'acceptabilité des logiciels [Nielsen, 1993a] représentée par la figure 2.1.

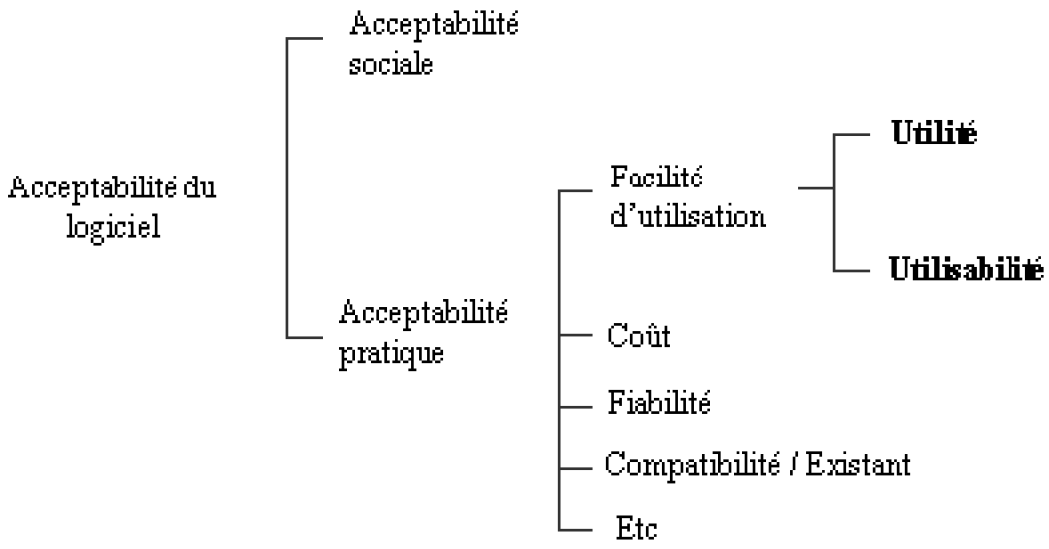


Figure 2.1 Attributs de l'acceptabilité des systèmes [Nielsen, 1993a]

Afin d'être acceptable, un logiciel doit être performant mais aussi utile et utilisable. En pratique [Barthet, 1988], la tentation est forte pour les informaticiens de présenter les systèmes interactifs selon leur logique de fonctionnement. Cette remarque reste d'actualité aujourd'hui, du moins, pour le domaine de l'ECD. En effet, il suffit de parcourir les actes des grandes conférences du domaine de l'ECD pour découvrir que l'évaluation de la qualité des outils obtenus dans ce domaine est à forte dominance fonctionnelle. Les habitudes, tendances, exigences des utilisateurs ne sont pas prises en compte. Pourtant, la compétence technique des logiciels de fouille de données est une condition nécessaire mais pas suffisante à l'évaluation de la qualité des ces outils. Selon la norme IEEE 830, une spécification d'exigences comprend :

- les fonctionnalités qui regroupent l'ensemble de services et fonctions que le système doit fournir,
- les interfaces externes qui sont relatives à l'identification des interactions avec les utilisateurs et les autres systèmes avec lesquels le nouveau système doit s'intégrer,
- la performance qui fait référence aux contraintes d'opération du système en disponibilité et en temps réponse,
- les attributs du système qui représentent les caractéristiques intrinsèques telles que la portabilité, l'exactitude, la maintenabilité, la sécurité, etc.
- les contraintes de conception qui peuvent être définies comme étant les contraintes sur la façon de développer le système.

Mettre des outils fonctionnellement performants à la disposition des utilisateurs n'implique pas qu'ils s'en serviront et surtout qu'ils en feront bon usage. Comme l'indique la figure 2.2, face à un outil et une tâche à réaliser, l'utilisateur final peut avoir une réaction positive, acceptant ainsi d'utiliser l'outil ou une réaction négative : il va alors soit utiliser partiellement l'outil soit ne pas l'utiliser.

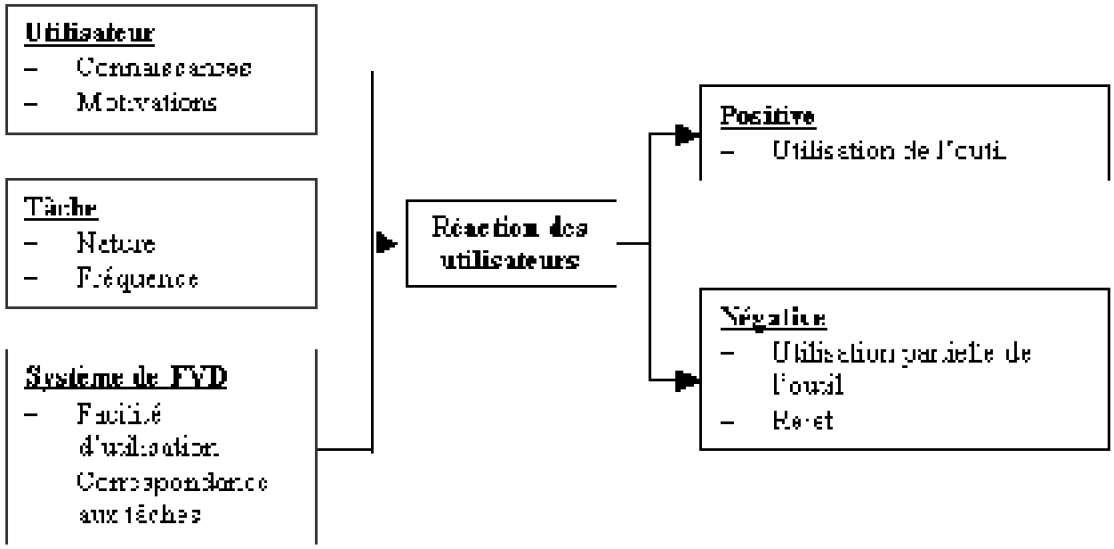


Figure 2.2 Acceptabilité des logiciels

Le but de cette étude qualitative rappelons-le est de définir une méthode permettant de découvrir, de traiter et de prévenir les problèmes graves susceptibles de se produire à l'issue de la conception d'un environnement de FVD afin d'en améliorer la qualité, en fonction des utilisateurs (de leurs actions, de leurs besoins, de leurs compétences, de leur profil, …) et en fonction de la tâche de FVD. La découverte des problèmes de qualité est faite à travers une étude de systèmes existants. Il s'en suit le développement de moyens de traitement des problèmes découverts. La prévention des problèmes est faite à travers des recommandations spécifiques au domaine de la FVD. En effet, au tout début d'un projet informatique, les usagers expriment leurs besoins en ce qui concerne le produit final. Il est nécessaire que ces besoins soient utiles, correctement et complètement exprimés.

A la suite de cette section, nous décrivons divers domaines de recherche qui proposent des techniques de détection des défaillances dans les systèmes informatiques. Les travaux en génie logiciel introduisent cet état de l'art.

2.6 La qualité des logiciels selon divers domaines de recherche

2.6.1 Qualité du logiciel et génie logiciel

D'après la norme [IEEE 610.12, 1990], le génie logiciel désigne l'application d'une approche systématique, disciplinée et quantifiable au développement, à l'opération et à la maintenance du logiciel.

Le génie logiciel est une discipline créée afin de faire face à la crise de l'industrie du logiciel à la fin des années 70, crise dont les signes les plus évidents étaient l'augmentation des coûts de production du logiciel, le non respect des délais de livraison, la non fiabilité des programmes, un écart important entre les besoins exprimés par les utilisateurs et les fonctions offertes par les systèmes informatiques.

Le génie logiciel consiste en un ensemble de techniques qui organise le travail de développement de logiciels de façon à assurer la qualité des produits construits. L'activité de conception d'un logiciel est régie par le cycle de vie du logiciel : l'ensemble des phases qui constituent son développement et son utilisation. Classiquement, on distingue 5 phases :

- la phase d'*analyse* : durant cette étape, les fonctions du futur logiciel sont décrites,
- la phase de *conception* qui s'intéresse à la description de la réalisation des fonctions du système informatique,
- la phase de *codage* durant laquelle les algorithmes décrits en phase de conception sont programmés,
- la phase de *tests* qui permet de comparer le comportement effectif du système avec le comportement prévu à la phase d'analyse,
- la phase d'*exploitation* correspond à l'étape de maintenance du logiciel. On ajuste, on corrige, on améliore et on ajoute de nouvelles fonctions au système.

Une étude préalable précède la première étape du cycle de vie du logiciel. L'étude préalable est constituée de l'analyse des besoins ou des exigences (« requirements » en anglais). On distingue :

- les *exigences fonctionnelles* qui expriment les fonctions que le système doit être capable d'effectuer,
- les *exigences non fonctionnelles*, quant à elles décrivent la qualité du futur système. Il existe plusieurs types d'exigences non fonctionnelles suivant le domaine d'étude. Nous pouvons citer par exemple : les *exigences d'utilisabilité* (qui se réfère à l'ergonomie du logiciel), les *exigences de fiabilité (génie logiciel)*, les *exigences de performance* (temps de réponse, quantité de mémoire nécessaire, temps de récupération des données après une erreur) etc.

En résumé, l'assurance qualité en GL comporte plusieurs facteurs : la performance du logiciel, la performance et le coût du matériel, la facilité de maintenance, la sécurité, la surêté, la fiabilité et l'interopérabilité. Le respect des exigences fonctionnelles et non fonctionnelles permet d'assurer la qualité des logiciels en GL. Dans ce domaine, les tests de logiciels permettent de découvrir et de traiter les défaillances des logiciels d'un point de vue technique. Dans la section suivante, une définition de la notion d'IHM est proposée.

2.6.2 Qualité du logiciel et interaction homme machine

L'Interaction Homme Machine ou Interaction Personne-Système se réfère à l'ensemble des phénomènes cognitifs, matériels, logiciels et sociaux mis en jeu dans

l'accomplissement de tâches sur support matériel (machine, système) [Calvary, 2002].

En IHM, l'accent est porté sur les facteurs humains pour l'analyse, la conception et l'évaluation des systèmes informatiques. Les facteurs humains pour le développement de logiciels constituent un modèle conceptuel très utile pour les développeurs de systèmes centrés utilisateurs. Les recherches de ce domaine sont fondées sur des résultats obtenus en psychologie expérimentale.

Pour concevoir des logiciels qui prennent en compte des facteurs humains et des IHM, il est nécessaire de procéder à une analyse de la tâche et une analyse de la communauté d'utilisateurs qui s'en servira. Ces différentes analyses sont basées sur le modèle de la tâche (donc de l'interaction) et le modèle de l'utilisateur, le respect de règles et principes ergonomiques. Les règles et les principes ergonomiques constituent une aide à déterminer les besoins, oriente certains choix de conception et enfin guide l'implémentation des fonctionnalités précises. En dernier ressort, on procède à l'évaluation, c'est-à-dire à la validation des choix de conception et d'implémentation par des tests d'usage sur prototypes.

La prochaine section est consacrée à la présentation du domaine de l'ergonomie du logiciel nécessaire à la conception des logiciels enrichis sous l'angle IHM.

2.6.3 Qualité du logiciel et ergonomie des logiciels

Selon le Petit Robert, l'ergonomie vient du grec ergon qui signifie travail. Il s'agit d'une étude scientifique des conditions (psychologiques et socio-économiques) de travail et de relation entre l'homme et la machine.

Selon la Société d'Ergonomie de Langue Française (1988), "L'ergonomie est la mise en œuvre de connaissances scientifiques relatives à l'homme et nécessaires pour concevoir des outils, des machines et des dispositifs qui puissent être utilisés avec le maximum de confort, de sécurité et d'efficacité pour le plus grand nombre."

Pour être ergonomique, un logiciel doit répondre aux critères d'utilisabilité et d'utilité. L'ergonomie est une science qui a pour objet l'étude de l'utilisateur, sa tâche et la communication homme machine pour effectuer la tâche. Une science possède des concepts, théories et méthodes. Du point de vue théorique, les modèles ergonomiques offrent un cadre de pensée et non des modèles opérationnels pour décrire, interpréter et prédire. On distingue l'ergonomie de surface et l'ergonomie de profondeur. L'ergonomie de surface concerne la présentation des informations par le système. Il existe des normes, des guides de style qui prodiguent des conseils pour une amélioration de la présentation des informations [Preece, 1993], [Nielsen et Mollich, 1990], [Mayhew, 1992], [Galitz, 1996], [Fernandes, 1995], etc.

L'ergonomie profonde dépend du contexte d'utilisation de l'outil. Afin d'opérer une étude ergonomique profonde, il est nécessaire de bien connaître la situation de travail. L'analyse de la situation de travail est donc nécessaire à cet effet.

Pour mener à bien notre étude qualitative des environnements de FVD, étude visant une assurance qualité des environnements de FVD, nous allons dans le chapitre 3 procéder à une analyse de la situation de travail en FVD. L'analyse de la situation de

travail est issue des recherches du domaine de l'ergonomie des logiciels. Mais avant, la section 2.7 présente différentes méthodes d'évaluation des logiciels.

2.7 Evaluation de la qualité des logiciels

L'évaluation permet d'établir un diagnostic d'usage des systèmes existants, d'effectuer une étude comparative des systèmes informatiques dans un souci de choix, de contrôler la qualité d'utilisation et l'utilité des logiciels, de comparer leurs avantages et leurs inconvénients. Utilisée à bon escient, l'évaluation permet d'identifier les difficultés susceptibles d'être rencontrées.

Pour mieux cerner notre apport en évaluation de la qualité des logiciels de FVD qui est présenté dans les chapitres 3 et 4, cette section présente diverses méthodologies utilisées en évaluation qualitative de logiciels.

En général, plusieurs aspects sont importants dans la qualité des logiciels. Nous pouvons citer :

- l'adéquation du logiciel aux caractéristiques des utilisateurs,
- l'adéquation du logiciel aux caractéristiques des tâches,
- l'adéquation du logiciel au contexte d'utilisation,
- l'adéquation du logiciel à l'organisation à laquelle appartiennent ses utilisateurs.

Afin de vérifier ces différentes adéquations, il existe de nombreuses méthodes. Ces méthodes peuvent soit requérir la participation des utilisateurs, soit s'appliquer aux caractéristiques des interfaces, soit être basées sur des « log files », soit requérir la participation des développeurs. Dans notre étude, nous nous intéressons aux deux premières catégories de méthodes.

2.7.1 Evaluation du point de vue des utilisateurs finaux

L'évaluation du point de vue des utilisateurs se distingue de l'évaluation par les experts (concepteurs (GL), ergonomes, etc.) et de l'évaluation par des directives.

Au début de l'évaluation avec des utilisateurs, plusieurs objectifs sont formulés. L'idée est de recueillir des données comportementales sur l'utilisation du système. On distingue plusieurs approches : les tests utilisateurs, les diagnostic d'usage, les questionnaires, les entretiens, etc.

Le diagnostic d'usage est une technique utilisée lorsqu'il existe une expérience d'utilisation, le système à évaluer doit être conçu et implémenté. Les méthodes relatives au diagnostic d'usage sont : les incidents critiques [Flanagan, 1954], l'analyse de traces écrites [Seuren, 1996] et les questionnaires.

- Les tests utilisateurs se déroulent en situation réelle ou en laboratoire. Il s'agit d'une démarche expérimentale qui peut nécessiter du matériel de type vidéo ou mouchard électronique.

- Les questionnaires constituent un ensemble de questions posées à l'utilisateur afin de recueillir des données relatives aux impressions de l'utilisateur après utilisation.
- Les rapports verbaux quant à eux permettent de recueillir de données relatives aux interactions des utilisateurs durant cette étape.
- L'observation constitue aussi un outil de collecte de données.
- Le principal avantage des méthodes d'évaluation du point de vue des utilisateurs est la prise en compte des utilisateurs réels et de leurs tâches. Il existe aussi quelques limites : ce type de méthode nécessite un utilisateur expérimenté pour découvrir la plus grande quantité d'erreurs. De plus, elle nécessite que le logiciel soit disponible, et par conséquent ne peut pas s'appliquer dès les phases de définition des besoins.

Les techniques d'évaluation ergonomiques décrites ci-dessous peuvent ou non servir à l'évaluation par des utilisateurs finaux.

2.7.2 Evaluation ergonomique des interfaces

Les différentes techniques décrites dans cette partie sont utilisées par les experts.

2.7.2.1 Cognitive walkthrough

Le principe de cette méthode [Wharton et al., 1994] est le suivant : à partir d'une liste de principes ou d'heuristiques, il s'agit d'inspecter l'interface et d'identifier des problèmes potentiels de qualité d'utilisation (d'utilisabilité). Un évaluateur parcourt l'interface et se comporte comme l'utilisateur final. Son objectif est d'évaluer la facilité d'apprentissage de l'utilisateur. La technique ainsi utilisée est basée sur une liste de questions qui dirigent l'attention du concepteur sur des aspects précis de l'interface importants afin de faciliter la résolution de problèmes et le processus d'apprentissage de l'utilisateur [Lewis, 1990]. L'avantage de cette méthode tient au fait qu'elle n'implique pas des utilisateurs. L'une des limites est qu'elle ne permet pas d'identifier des problèmes liés au domaine d'application.

2.7.2.2 L'évaluation coopérative

L'évaluation coopérative [Monk et al, 1993] est une méthode conçue pour des informaticiens par des psychologues spécialistes des interfaces homme-machine. Cette méthode permet d'obtenir des données portant sur des problèmes susceptibles d'être rencontrés par les utilisateurs avec un minimum d'effort afin d'améliorer le logiciel. Le principal atout de cette méthode est qu'elle peut être utilisée par des concepteurs qui ne possèdent pas de notions en ergonomie. Elle repose sur 3 principales étapes : le recrutement des utilisateurs, la préparation des tâches à exécuter par les utilisateurs et l'interaction avec les utilisateurs qui permet de collecter des données.

L'inconvénient est que cette méthode nécessite une maquette ou le produit final. Elle ne peut donc pas être utilisée dès les phases en amont du processus de conception du logiciel.

2.7.2.3 L'évaluation avec des lignes directrices « guidelines »

Les lignes directrices constituent un ensemble de suggestions pratiques pour les concepteurs d'interfaces entre des systèmes informatiques et les utilisateurs. Les lignes directrices peuvent avoir plusieurs sources :

- les résultats expérimentaux,
- des prédictions issues de théories de l'activité humaine,
- les principes de psychologie cognitive, le jugement d'experts,
- les principes d'ergonomie, l'expérience en génie logiciel, l'expérience pratique, etc …
- Plusieurs principes sous-tendent les lignes directrices :
- la facilité d'apprentissage du logiciel, la facilité d'usage, les fonctionnalités,
- les caractéristiques des utilisateurs (novices, experts ou utilisateurs occasionnels),
- la cohérence,
- l'allocation de fonctions,
- les modèles mentaux,
- la nécessité de fournir plusieurs alternatives, etc.

Le respect d'une règle ergonomique influence la réalisation d'un critère ergonomique, celui-ci influençant le respect d'un facteur d'utilisabilité [Mariage, 2005].

De plus, divers auteurs proposent des lignes directirices pour l'évaluation de la qualité des interfaces : les normes sont relatives à l'utilisation du logiciel dans un certain contexte d'usage. On distingue les différentes normes ISO suivantes pour l'évaluation qualitative :

- ISO 11428 : ergonomie, signaux visuels de danger, exigences générales, conception et essais,
- ISO 14915 : ergonomie des logiciels pour les interfaces utilisateur multimédias,
- ISO/TR 16982 : ergonomie de l'interaction homme-système,
- ISO/TR 19358 : ergonomie, élaboration et mise en oeuvre des tests des systèmes de technologie de la parole,

Les critères de Bastien et Scapin [Bastien et Scapin, 1993], [Scapin et Bastien, 1997] concernent les aspects de l'interface utilisateur tels que le guidage, la charge de travail, le contrôle explicite, l'adaptabilité, la gestion des erreurs, l'homogénéité/ la cohérence, la signifiance des codes et des dénominations ou la compatibilité.

Les heuristiques de Nielsen quant à elles sont relatives à la visibilité du statut du système, la compatibilité entre système et monde réel, le contrôle par l'utilisateur et la liberté de l'utilisateur, la cohérence et les normes, la prévention des erreurs, la reconnaissance plutôt que le rappel, la flexibilité et l'efficacité d'utilisation, l'esthétique et la conception minimaliste, l'aide aux utilisateurs pour la reconnaissance, le diagnostic et la réparation des erreurs, l'aide et la documentation.

En plus des critères ergonomiques de [Bastien et Scapin, 1993b], des heuristiques de

Nielsen [Nielsen et Mollich, 1990] et des normes, il existe aussi des principes ergonomiques [Crampes, 1995] et des directives [Dumas, 1999], [Brown, 1988], les critères de [Schneiderman, 1992] qui concernent la durée d'apprentissage, la performance, le taux d'erreur, la satisfaction subjective, la rétention temporelle.

Les principes ergonomiques de [Crampes, 1995] ont été mis en valeur par des ergonomes et concernent les caractéristiques de l'interaction suivantes : la cohérence, la concision, le retour d'informations, la structuration des activités, la flexibilité et la gestion des erreurs. Les principes généraux de conception suivants sous-tendent ces principes : l'emploi des métaphores, l'étude des activités des utilisateurs, la cohérence de l'interface et sa transparence.

Les directives de [Dumas, 1999] quant à elles sont basées sur des incitations. En effet, des incitations efficaces aident l'utilisateur à se rappeler comment exécuter des procédures.

Les directives de [Brown, 1988] sont essentiellement basées sur les éléments de l'interface, les formats d'affichage, le langage, la couleur, les graphiques, le dialogue, les entrées de données, les dispositifs de contrôle et d'affichage, les messages d'erreurs et aide en ligne et l'implémentation de l'interface.

Les méthodes basées sur des directives sont simples à mettre en œuvre, elles ne nécessitent ni analyse ni étude préalable de la tâche. Cependant, elles ne concernent que l'interface et négligent l'utilisateur et son travail, ce qui entraîne une faible fiabilité des résultats qui dépendent en plus de l'évaluateur. Il s'agit donc de méthodes incomplètes pour l'évaluation de la qualité.

Il importe de souligner que toutes les différentes techniques d'évaluation ergonomique des logiciels nécessitent d'être adaptée avant toute application au domaine de la FVD.

De toutes les méthodes décrites dans cette section, nous allons présenter beaucoup plus amplement l'évaluation heuristique et l'évaluation par des recommandations ergonomiques.

2.7.3 Evaluation heuristique

L'évaluation heuristique a pour objectif de détecter les problèmes d'utilisabilité et de design d'une interface [Nielsen et Philipps, 1993b]. Nielsen a défini des heuristiques et les problèmes recensés sont formulés en terme de non respect de ces heuristiques. L'évaluation heuristique est un processus itératif effectué par des experts, ayant l'habitude de la liste d'heuristiques utilisées qui guident l'évaluation. Cette méthode peut être appliquée dès que l'interface est conçue. Les problèmes de qualité d'utilisation détectés sont des déviations par rapport à ces heuristiques qui sont : les dialogues simples et naturels, parler le langage de l'utilisateur, minimiser la charge de mémoire de l'utilisateur, la consistance, le feed-back, les sorties clairement signalées, les raccourcis, les bons messages d'erreurs, prévenir les erreurs, l'aide et la documentation.

Les avantages de cette méthode sont qu'elle est peu onéreuse, intuitive, qu'il est facile de motiver les gens pour la réaliser, qu'elle ne nécessite pas de planning et qu'elle

peut être utilisée très tôt dans le cycle de vie. Elle ne requiert pas la présence des utilisateurs et peut être réalisée individuellement.

En ce qui concerne les inconvénients, l'évaluation heuristique ne fournit pas d'aide pour résoudre les problèmes détectés. La méthode est aussi biaisée par la compétence des évaluateurs : plus les évaluateurs sont compétents, plus la méthode a de chances de donner de bons résultats. Cette méthode ne peut être efficace que si elle est appliquée parallèlement par plusieurs évaluateurs. Les problèmes liés au développement de l'application sont difficilement identifiables avec cette approche.

2.7.4 Evaluation par des critères ergonomiques de Bastien et Scapin

Les critères ergonomiques de [Bastien et Scapin, 1993b] émanent de l'examen de près d'un millier de recommandations issues d'études empiriques et de pratiques courantes pour la conception de systèmes interactifs. Ces critères ont été validés, testés et comparés aux normes ISO 9241-10. Mais elles ne concernent que les interfaces textuelles et graphiques.

L'évaluation avec des guides de recommandation s'appuie aussi sur l'expérience et les connaissances des experts. L'utilisation des guides qui contiennent des connaissances exprimées sous forme de règles renforce la technique d'évaluation. Pour avoir des résultats effectifs, cette technique d'évaluation nécessite au moins trois experts [Pollier, 1991] et débute après la réalisation d'une maquette, du prototype ou du produit final. La démarche d'évaluation est la suivante : l'expert parcourt l'interface et détecte les recommandations ergonomiques qui n'ont pas été respectées. Une première étude a été réalisée en ce sens par [Pollier, 1991]. Afin d'améliorer cette évaluation, [Bastien et Scapin, 1993] ont défini les critères ergonomiques qui soutiennent l'évaluation des interfaces. Ces critères offrent un ensemble de principes dont il doit vérifier l'application sur l'interface et guident ainsi l'expert. Les critères ergonomiques élémentaires sont les suivants : guidage, groupement/distinction par la localisation, groupement/distinction par le format, feed-back immédiat, clarté, concision, actions minimales, charge mentale, actions explicites, contrôle utilisateur, flexibilité, prise en compte de l'expérience utilisateur, protection contre les erreurs, qualité des messages, correction des erreurs, homogénéité/consistance, signifiance des codes et compatibilité.

Par rapport à l'évaluation heuristique, en utilisant des recommandations ergonomiques, l'expert détecte beaucoup plus de problèmes d'utilisabilité, ce qui réduit donc le nombre d'experts nécessaires à la détection des problèmes. L'inconvénient majeur de cette approche tient au fait que la méthode ne peut pas permettre la détection de tous les problèmes d'utilisabilité car elle est basée sur l'expérience, le travail et les connaissances des experts.

2.7.5 Conclusion

L'objectif de cette section était de présenter des techniques d'évaluation de logiciel. Il est aussi à noter que de nombreuses classifications de techniques d'évaluation de logiciels sont disponibles dans la littérature [Farenc, 1997], [Balbo, 1994], [Coutaz, 1990], [Howard, 1987], [Senach, 1990], [Mariage, 2005], etc.

[Coutaz et al., 1993] distingue dans sa classification les évaluation prédictives et les évaluations expérimentales. Les méthodes d'évaluation prédictives regroupent les différentes techniques utilisées tout au long de la conception des logiciels. Ces méthodes ne nécessitent ni utilisateurs finaux, ni système implémenté. L'idée est de pouvoir recenser toutes les erreurs avant que le système ne soit implémenté de façon à limiter les modifications.

Les méthodes expérimentales sont réalisées soit sur le système entier, soit avec une maquette ou un prototype. Ces évaluations peuvent nécessiter la présence des utilisateurs et permettent d'observer et d'analyser l'utilisation qui est faite par les utilisateurs du système.

Pour [Senach, 1990], il existe des évaluations empiriques et des évaluations analytiques. Pour les méthodes empiriques, l'évaluation est faite avec des utilisateurs réels et des tâches réelles. L'objectif est de recueillir des données comportementales sur l'utilisation du système. Cette approche nécessite l'affectation d'une tâche à réaliser et le développement complet ou partiel du logiciel.

Les méthodes d'évaluation analytique consistent en une étude des interfaces selon un ensemble de lignes directrices afin de contrôler qu'elles possèdent bien certaines qualités et de détecter les problèmes qu'elles peuvent poser.

Selon [Karat, 1988], les méthodes d'évaluation sont classifiées suivant qu'elles s'appliquent directement sur les utilisateurs ou qu'elles sont basées sur leurs tâches.

Les méthodes qui impliquent directement les utilisateurs sont basées sur l'idée suivant laquelle il est possible de connaître la qualité d'une interface en analysant les données recueillies auprès des utilisateurs après utilisation.

Les méthodes basées sur les tâches analysent plus ou moins formellement les besoins des utilisateurs. La tâche évaluée est réelle mais, l'utilisateur est simulé. Cette méthode utilise des modèles simplifiés de l'opérateur humain et des théories sur la performance de l'opérateur humain [Howard, 1987].

Ces différentes méthodes ont au moins l'un des avantages suivants : elles permettent la prise en compte de la réalité de l'utilisateur et de sa tâche, elles peuvent intervenir très tôt dans le processus de développement du logiciel, elles sont simples à mettre en œuvre et elle ne nécessitent ni analyse ni étude préalable de la tâche.

En ce qui concerne leurs inconvénients, la portée de la couverture du système est faible. Il s'avère impossible de faire réaliser toutes les tâches possibles. Ces techniques nécessitent des expérimentateurs expérimentés, l'interface doit être disponible. L'évaluation basée sur des modèles engendre un appauvrissement de la réalité dû à l'effet de la modélisation, de plus, les modèles sont lourds à réaliser et à utiliser.

Les méthodes d'évaluation basées sur des experts et la simulation des utilisateurs néglige les utilisateurs et leurs travaux, elles possèdent une faible fiabilité des résultats, car ils dépendent de la qualité de l'expert. A présent, nous allons nous intéresser aux études qualitatives en FVD.

2.8 La qualité des logiciels de fouille visuelle de données

2.8.1 La qualité des modèles de données : estimation de l'erreur

L'analyse des données est une étape du processus d'ECD. A cette étape, on dispose de données structurées. On peut procéder soit à une classification, soit à une estimation, soit à une prédiction, soit à une segmentation, soit à une recherche de règles d'association dans ces données. L'objectif de ces traitements est de retrouver un modèle des données. On parle d'apprentissage supervisé ou d'apprentissage non supervisé selon que les exemples d'apprentissage soient étiquetés ou non.

L'analyse de données aboutit à la création du modèle des données, l'étape suivante consiste en l'évaluation de la qualité de ce modèle. L'évaluation permet de mesurer la justesse et la précision du modèle, c'est-à-dire de voir dans quelles mesures le modèle confirme les hypothèses de départ. Il est aussi question de savoir si le modèle est facile à comprendre, valide sur de nouvelles données, utile, nouveau. Dans le cadre de nos travaux, nous nous situons dans le domaine de l'apprentissage supervisé (les exemples d'apprentissage sont étiquetés) et plus particulièrement de la classification supervisée.

L'évaluation de la qualité du modèle des données consiste à estimer l'incertitude autour de l'estimateur d'erreur future. L'erreur de prédiction est la mesure standard de qualité des modèles d'apprentissage supervisé. L'estimation d'erreur en analyse de données a deux objectifs : la validation qui permet d'ajuster des paramètres de l'algorithme utilisé et le test qui permet d'évaluer les performances et de comparer les algorithmes.

L'estimation d'erreur peut être basée sur des échantillons ou des pénalités. En ce qui concerne l'estimation basée sur des échantillons, on peut rechercher l'erreur de test avec la méthode de holdout, la validation simple. On peut aussi procéder à une validation croisée.

L'estimation basée sur des pénalités quant à elle utilise des critères tels que AIC (Akaike information criterion), BIC (Bayesien information criterion) ou MDL (minimum description length).

Une autre méthode naïve d'évaluation de la qualité du modèle des données serait d'utiliser tous les exemples d'apprentissage pour entraîner et calculer le taux d'erreur sur cet ensemble d'entraînement. Mais, certaines méthodes d'apprentissage tendent à s'ajuster aux données d'entraînement. Il est donc nécessaire d'avoir un ensemble de test indépendant de l'ensemble d'entraînement pour mesurer le taux d'erreur. Des méthodes de ré échantillonnage peuvent donc être utilisées pour obtenir un estimateur non biaisé. Lorsque l'on dispose de très peu d'échantillon, il est difficile de déterminer si le taux d'erreur obtenu est précis ou si la situation dans laquelle on aboutit est due au hasard, des méthodes telles que le leave one out ou la validation croisée peuvent être utilisées.

Dans le domaine beaucoup plus spécifique de la classification supervisée qui nous intéresse, l'évaluation de la qualité des modèles d'analyse de données se fait suivant deux approches : la validation croisée et le boostrap. L'idée de ces deux méthodes est

d'estimer le taux d'erreur de classification. A cet effet, on dispose d'un ensemble d'entraînement (apprentissage) et un ensemble de validation (test). Soit *D* un ensemble de donnée et *x* un entier. Le principe de la validation croisée est le suivant : découper E en x parties égales ($D_1,\ldots,D_x$), pour tout D_i, construire un modèle *M* avec l'ensemble $D-D_i$, évaluer l'erreur e_i de *M* avec D_i, retourner la moyenne des erreurs $e_i = \sum_{i=1}^{x} e_i / x$.

Soit un ensemble d'entraînement *T* de *n* éléments, le principe de l'estimation avec Bootstrap est le suivant : choisir *K* ensembles de *n* éléments avec remise à chaque sélection. Calculer le taux d'erreur et sa variance avec les éléments non sélectionnés pour l'entraînement.

Dans le processus de validation croisée, les données sont divisées de manière répétitive en un ensemble d'apprentissage et un ensemble test sur lequel la précision est mesurée. En général, une fourchette de l'ensemble de départ est utilisée pour l'apprentissage et le reste pour le test. Ce processus est répété n fois et la précision est la moyenne des précisions obtenues pour chaque test.

L'erreur de prédiction obtenue par validation croisée ou boostrap est la mesure standard pour la mesure de la qualité des modèles d'apprentissage. Cette mesure propre au domaine de l'ECD ne permet pas par exemple d'évaluer la qualité de l'interface utilisée pour la fouille de données encore moins l'acceptabilité de cette interface. Nous pensons que l'erreur de prédiction est une condition nécessaire mais pas suffisante pour juger de la qualité des outils de FVD. A cet effet, nous allons passer en revue dans la section 2.8.2 les méthodes utilisées en interaction homme machine, génie logiciel et ergonomie cognitive. Si l'on se réfère à la définition de la qualité des logiciels, les méthodes décrites dans cette section font partie des facteurs internes de qualité. Pour terminer, cet état de l'art, nous présentons d'autres approches d'évaluation de facteurs internes de qualité des logiciels. En visualisation pour la fouille de données, les travaux de [Grinstein et al, 1997] ont permis l'évaluation qualitative experte et fonctionnelle des méthodes de représentation graphique de données par rapport à la capacité mémoire des ordinateurs, à leur vitesse d'exécution et à leurs capacités graphiques. Il existe aussi des travaux traitant de l'analyse qualitative d'outils en analyse de données [King et al, 1998], [Collier et al., 1999]. Cette analyse concerne les aspects techniques des algorithmes (performances, temps d'exécution, précision). Pour les besoins de ce type d'évaluation, des efforts de création d'entrepôts de données ont été l'objet des projets tels que l'UCI Machine Learning Repository [Blake et Merz, 1998], le Kent Ridge Bio Medical Dataset Repository [Finyan et Huiqing, 2002], Statlog [Metal, 2005].

2.8.2 Etude de la qualité externe des logiciels de fouille visuelle de données

En ECD, la validation croisée, le holdout ou le boostrap constituent des techniques d'évaluation de la qualité des modèles des données qui sont construits à travers des interfaces. Afin de faciliter la modélisation des données, il est nécessaire que l'interface soit de bonne qualité. Dans le processus de développement de logiciels, comme spécifié ci-dessus, il existe des facteurs visibles par les développeurs et des facteurs visibles par les utilisateurs finaux. Dans ce travail, nous désignons par défaillance du logiciel la

non-conformité avec les attentes des utilisateurs. Les techniques d'étude de la qualité des logiciels d'analyse de données décrites dans la section 2.8.1 sont visibles par les développeurs. En ECD, très peu de travaux sont consacrés aux autres aspects concernant les défaillances du logiciel, notamment l'interaction homme machine, l'analyse du contexte, des actions, de l'activité, l'ergonomie des interfaces et cognitive. En général, l'évaluation de la qualité des logiciels en vue de la détection des défaillances de logiciels peut être faite en amont du processus de conception du logiciel c'est-à-dire après analyse des besoins, elle peut être intégrée dans le cycle de développement itératif du logiciel (génie logiciel), l'étude de la qualité du logiciel peut aussi avoir lieu après développement ou prototypage (test du logiciel, évaluation ergonomique). Il existe de nombreuses méthodes d'évaluation de logiciels. Une classification de ces méthodes d'évaluation peut être faite en fonction de leurs objectifs, du moment où l'évaluation a lieu ou du type d'utilisateurs impliqués dans l'évaluation. En génie logiciel, on parle de test du logiciel. Le test de logiciel a pour objectifs de détecter les déviations par rapport aux spécifications, détecter des erreurs, augmenter la confiance dans le programme, déterminer un niveau de fiabilité dans le logiciel, évaluer les performances ou évaluer le comportement en charge.

Du point de vue IHM et génie logiciel, les aptitudes attendues de l'évaluation de logiciels sont de plusieurs ordres : la satisfaction des besoins des utilisateurs, la fiabilité, la pérennité, l'interopérabilité, la conformité aux standards et un bon ratio coût / performance.

Pour matérialiser concrètement ces différentes méthodes d'amélioration de la qualité des logiciels, il existe des normes et des critères de développement de systèmes de qualité. Nous pouvons citer les standards, par exemple la norme ISO 9241 [Iso, 1999] et HFES/ANSI 200, les guides de style, par exemple le guide de Sun [Sun, 1999], [Constantine, 2001] et [Detweiler et Omanson, 1996], les compilations de règles [Vanderdonckt, 1994] et les critères liés au contexte de l'activité : l'activity checklist de [Kaptelinin, 1999]. L'un des travaux précurseurs du domaine de la qualité du logiciel a donné naissance au modèle de McCall [McCall et al., 1977] qui recense une cinquantaine de critères permettant d'exprimer la qualité du logiciel. Partant de la liste de McCall, [Ghezzi et al., 1991] a établi une liste plus exhaustive de critères de qualité des logiciels. Cette liste comprend seize critères. Il existe d'autres modèles tels que celui de [Murine et Carpenter, 1984]. Un autre ensemble de facteurs de qualité des logiciels a été réalisé par [Boëhm, 1978]. Ces facteurs concernent les fonctionnalités et les performances du logiciel du point de vue génie logiciel et de l'utilisabilité.

Plus récemment, l'évaluation heuristique [Nielsen, 1993a] et les recommandations ergonomiques [Bastien et Scapin, 1999] mettent beaucoup plus l'accent sur l'évaluation ergonomique des IHM. L'étude des facteurs humains et des IHM complète la qualité des logiciels dans les fondements théoriques de ces travaux qui sont basés sur un ensemble de critères. Des travaux de psychologie cognitive relatifs à la visualisation de données [Healey, 1996] proposent aussi des primitives intéressantes dans un contexte d'évaluation qualitative.

Mais, certaines des techniques décrites ci-dessus sont assez générales. Pour des nouveaux systèmes tels que ceux de la FVD, une piste de recherche consiste à retrouver

les caractéristiques de qualité puis à voir comment les évaluer. Une autre piste de recherche peut consister à adapter les normes, les standards ou critères existants aux nouveaux systèmes. On obtient alors des recommandations de qualité spécifiques, des critères de qualité spécifiques et des questionnaires spécifiques. Dans le cadre de la FVD, les questions de qualité externe des logiciels restent d'actualité : que faut-il faire pour garantir la qualité d'un environnement de FVD ? Quels principes faut-il appliquer ? Quand et comment les appliquer ?

2.9 Etude qualitative et détection de défaillances dans les logiciels de FVD

L'évaluation de la qualité du logiciel qu'elle soit ergonomique, enrichie sous l'angle IHM ou alors orientée génie logiciel (test de conception) dépend du positionnement de la démarche dans le cycle de vie du logiciel. L'évaluation peut être intégrée au cycle de développement du logiciel ou peut avoir lieu après conception ou prototypage. On distingue deux grandes catégories d'approches d'évaluations de logiciels [Coutaz, 1990] : les approches empiriques qui regroupent les tests de conception et les diagnostics d'usage et qui sont centrées sur les utilisateurs. La seconde approche est de type analytique. L'approche analytique est centrée sur les experts et non sur l'utilisation de l'outil. Ce type d'approche regroupe les modèles d'évaluation de type informelle (à caractère théorique) et les modèles d'évaluation formelle. Le modèle d'évaluation informel fait intervenir plusieurs spécialistes, pour obtenir une expertise partielle de chaque expert et la combiner. Cette évaluation peut être heuristique ou alors basée sur une grille d'évaluation.

Le modèle d'évaluation formelle contient des modèles prédictifs et des modèles de qualité. Si l'on désire classer les méthodes d'évaluation de logiciels en fonction du moment pendant lequel elles sont utilisées, on va distinguer dans la première catégorie les approches telles que l'évaluation formative [Howard, 1987] ou l'évaluation prédictive [Coutaz et al., 1993]. Tout comme l'évaluation formative, ce type d'approche ne nécessite pas de système implémenté.

En ce qui concerne la seconde catégorie : les méthodes d'évaluation de prototypes ou produits finaux, nous pouvons citer les méthodes sommatives [Howard, 1987] ou l'approche dite expérimentale [Coutaz et al., 1993] pour l'évaluation de systèmes réels (maquette, prototype ou système final).

Il existe donc des techniques d'évaluation de logiciels qui impliquent directement les utilisateurs [Karat, 1988] et d'autres techniques basées sur la tâche [Karat, 1988].

L'idée premièreici est de découvrir, de traiter et de prévenir les problèmes graves susceptibles de se produire à l'issue de la conception d'un environnement de FVD. A cet effet, il est nécessaire de procéder à une inspection experte et à une évaluation utilisateur. Toute la difficulté inhérente à une telle démarche consiste en la définition des méthodes d'analyse et d'évaluation adaptées aux experts, aux utilisateurs et à la FVD. Le travail présenté dans cette thèse a conduit à la définition de recommandations pour une inspection experte et des métriques d'analyse et d'évaluation spécifiques au domaine de

la fouille visuelle de données. Ces métriques ont deux objectifs : elles pourront servir d'indications aux concepteurs des environnements de FVD et de feuille de route aux évaluateurs.

2.9.1 Processus de détection des défaillances

En général, les méthodes d'évaluation de la qualité d'utilisation permettent soit de mesurer l'aisance des utilisateurs, soit de faire une enquête d'usage ou d'évaluer l'interface. Chacune de ces méthodes permet une évaluation partielle. Pour les besoins d'élaboration du diagnostic des systèmes de FVD, notre objectif est de procéder à une inspection experte de tous ces aspects en utilisant une seule méthode et en procédant à une enquête auprès des utilisateurs finaux. Pour l'inspection experte, nous nous servons de l'analyse de la tâche et des utilisateurs pour adapter les recommandations de qualité d'utilisation existantes au domaine de la FVD, voir la figure 2.3.

En effet, les techniques d'inspection experte existantes reposent sur des standards généraux qui ne répondent pas aux spécificités de la FVD. Pour l'adaptation, une analyse de la tâche a été nécessaire. Après l'analyse de la tâche de FVD, nous nous sommes intéressés aux méthodes de test de performances des utilisateurs que nous avons étendus aux fonctionnalités des logiciels de FVD. A cet effet, dans les premiers balbutiements de nos travaux [Fangseu Badjio et Poulet, 2004a], nous avons procédé à une adaptation des recommandations (« guidelines ») existantes aux spécificités du domaine pour aboutir à des recommandations spécifiques à la FVD partant des recommandations ergonomiques existantes et d'une analyse exhaustive de la situation de travail en FVD. Les recommandations issues de ce processus ne couvrent cependant pas l'ensemble des besoins des logiciels de FVD, elles ne concernent que l'ergonomie des interfaces. L'analyse de ces logiciels avec ces recommandations n'est donc pas très fine. Les domaines de recherche tels que le GL, l'ergonomie des logiciels et l'ergonomie cognitive, la théorie de l'activité et la théorie de l'action nous ont été d'une très grande utilité pour la définition d'une méthode beaucoup plus exhaustive. La résultante de ces investigations est un ensemble de recommandations pour une inspection experte des outils de FVD qui est détaillée au chapitre 3 et un ensemble de métriques d'analyse des environnements de FVD dont une extension peut servir et a servi à l'évaluation par des utilisateurs des outils de ce type, présenté au chapitre 4.

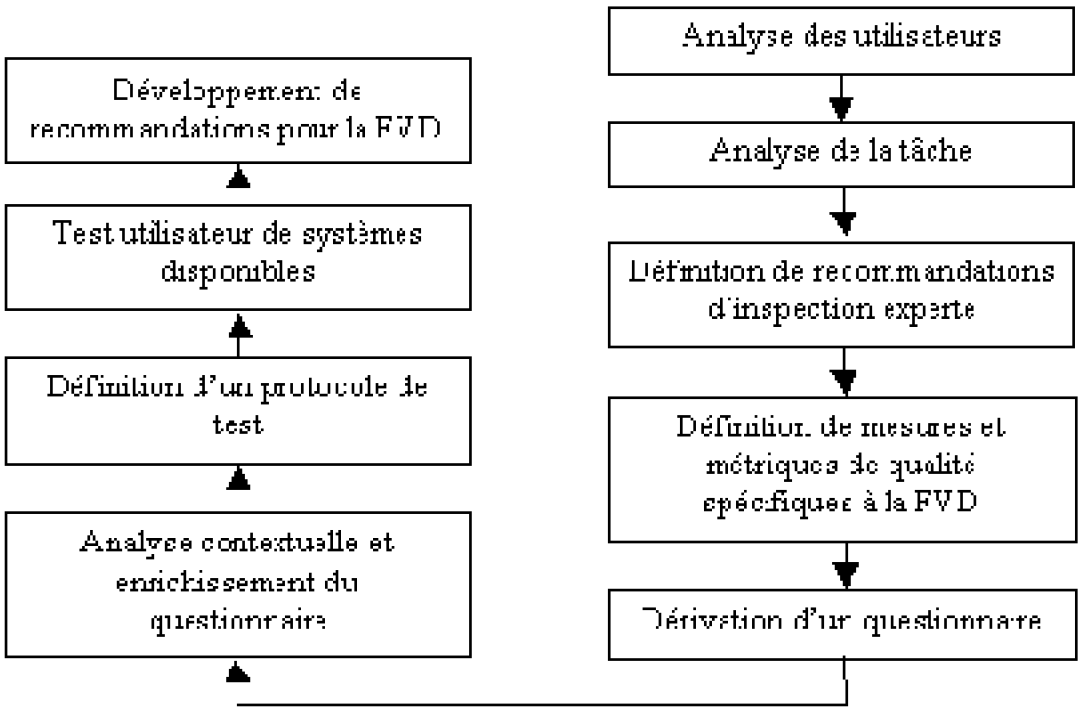


Figure 2.3 Processus de diagnostic des environnements de FVD

Le processus de définition de recommandations (figure 2.3) pour le développement d'outils de FVD est initialisé par la définition de critères d'analyse et d'évaluation de logiciels spécifiques au domaine de la FVD. Au préalable, l'analyse des utilisateurs et de la tâche de FVD a permis de définir des recommandations pour une inspection experte de ces environnements. L'idée ici est de retrouver les besoins des utilisateurs d'un environnement de FVD et de voir dans quelles mesures les environnements existants répondent à ces besoins. La filière ergonomie de logiciels propose à cet effet d'opérer une analyse de la situation de travail. Ces analyses de la tâche et des utilisateurs de FVD qui feront l'objet du chapitre 3 ont permis de définir un ensemble de métriques de qualité en FVD. A partir de ces métriques, nous avons dérivé un questionnaire d'évaluation des outils de FVD qui a servi grâce au protocole de test au diagnostic des outils de FVD comme l'indique le chapitre 4.

Dans la suite de cette section, nous allons procéder à une synthèse des méthodes de détection des défaillances dans les logiciels.

2.9.2 Synthèse : détection des défaillances en FVD

Afin de diagnostiquer les outils de FVD, les avantages des modèles issus de champs de recherche tels que l'ergonomie de logiciels, les IHM, le génie logiciel, etc., nous ont permis d'analyser et de comprendre finement les utilisateurs ainsi que la tâche de FVD. Les avantages des tests d'utilisabilité (basés sur les facteurs humains et les IHM) nous ont poussé à penser dans un premier temps à utiliser cette approche [Fangseu Badjio et Poulet, 2004a]. Ces avantages sont notamment la possibilité pour l'évaluateur d'observer

l'utilisateur dans un contexte réel d'utilisation, ainsi, les problèmes identifiés sont ceux que l'utilisateur rencontre réellement lorsqu'il se sert du logiciel. De plus, les difficultés freinant l'utilisateur dans sa tâche permettent d'identifier objectivement ses problèmes. En dernière position, des mesures relatives à la qualité peuvent être effectuées pendant le test. Néanmoins, ce type de test peut difficilement couvrir l'ensemble des fonctionnalités du logiciel. L'utilisabilité permet d'opérer un diagnostic d'usage. Durant l'évaluation des outils de FVD, il s'avère aussi nécessaire de procéder à une prédiction de la performance, réflexion épistémique, explication du fonctionnement humain ou aide à la conception d'interface. Les modèles existants nécessaires à ces différents traitements ne sont pas dédiés à la FVD et par conséquent ne prennent pas en compte les spécificités de ce domaine.

Ainsi, nous avons procédé à une seconde analyse de la situation de travail. Pour nous, l'analyse de la situation de travail consiste à étudier le contexte dans lequel l'outil de FVD est utilisé. Il s'agit plus précisément de l'étude des phénomènes susceptibles d'être observés lorsqu'un utilisateur travaille sur un environnement de FVD afin de définir les moyens à mettre en œuvre pour l'assister. [Ankerst, 2000] a proposé un modèle de tâche en FVD, l'analyse de la tâche de FVD du chapitre 3 a été basée sur ce modèle. En ce qui concerne l'analyse des utilisateurs, dans ce même chapitre, nous proposons un modèle de l'utilisateur basé sur un système multi-agent, ce modèle permet de prendre en compte les spécificités de tout utilisateur. Tout au long du diagnostic des systèmes de FVD, les points qui nous intéressent sont les suivants : le logiciel satisfait-il les besoins de l'utilisateur dans un contexte d'utilisation déterminé ? Le degré de réalisation des objectifs poursuivis dans l'interaction est-il élevé ? Les ressources disponibles pour atteindre les objectifs sont-elles efficaces ? L'utilisation du logiciel, dans l'accomplissement de la tâche, est-il satisfaisant ?

Pour les utilisateurs, les critères de préférence des logiciels sont :

- l'efficacité fonctionnelle, l'efficacité invoque la précision et l'intégralité avec lesquelles des utilisateurs donnés peuvent atteindre des buts donnés dans des environnements particuliers,
- la précision, intégralité et buts des utilisateurs,
- la vitesse de navigation,
- l'efficience de navigation, l'efficience fait allusion aux ressources déployées en fonction de la précision et de l'intégralité des buts atteints,
- les ressources déployées (fonction de précision et buts atteints),
- la vitesse d'entrée de données,
- la satisfaction, la satisfaction adresse le confort et l'acceptabilité du système (de l'interface) pour ses utilisateurs et pour les personnes qui sont affectés par le système,
- le confort et acceptabilité du logiciel,
- l'apprentissage, la facilité d'apprentissage d'une interface peut se traduire notamment par l'utilisation de conventions propres à un ensemble d'interfaces connues

(cohérence entre les interfaces) éliminant ou diminuant ainsi la nécessité d'apprendre des notions nouvelles.

La première partie du chapitre 3 s'articule autour de la modélisation de l'utilisateur de FVD. Nous montrons dans quelles mesures ce modèle utilisateur pourrait assister le diagnostic des systèmes de FVD, comment sera opéré sa mise en œuvre, son initialisation, son utilisation et sa maintenance. Deuxièmement, nous décrivons le modèle de tâche.

2.10 Conclusion

La non qualité des systèmes informatiques engendre des pertes inestimables. Il s'avère nécessaire d'étudier la qualité de tout système. Nous nous intéressons au cas de la FVD. Afin de mieux cerner nos travaux en amélioration de la qualité des outils actuels de FVD, le présent chapitre a présenté différentes méthodes d'évaluation qualitative des logiciels. Dans la plupart de ces méthodes, l'évaluation porte essentiellement sur les éléments de l'interface utilisateur et sur l'utilisation de cette interface. Nous avons aussi constaté que les méthodes d'évaluation existantes étaient assez générales et n'adhéraient pas aux spécificités de la FVD. A cet effet, il s'est avéré nécessaire de procéder à une analyse de la situation de travail en FVD afin de définir une méthode qui sied le mieux à ce domaine. En effet, l'évaluation qualitative peut être faite par les concepteurs du logiciel, les experts en ergonomie ou les utilisateurs finaux. Il existe des méthodes pour chaque catégorie d'évaluateur.

Faisant suite à l'état de l'art présenté dans ce chapitre, nous proposons une méthode d'inspection experte des outils de FVD dans le chapitre 3. Dans le chapitre 4 de ce mémoire, nous proposons une méthode d'analyse qualitative des outils de FVD dont une extension servira comme nous le verrons au diagnostic des logiciels de ce type.

Partie 2 : Contributions en analyse qualitative pour la FVD

Publications

Fangseu Badjio E., Poulet F.: *Towards usable visual data mining environments*, to appear in proc. of **HCII'05**, the 11th International Conference on Human-Computer Interaction, Las Vegas, Nevada, USA, Jul 2005.

Fangseu Badjio E., Poulet F.: *Ergonomic Criteria for Visual Data Mining*, International Symposium of Visual Data Mining (VDM) of IEEE 9th International Conference on Information Visualization (**IV@VDM'05**), Poster, London, UK, Jul 2005 (accepted).

Fangseu Badjio E., Poulet F.: *Visual data mining tools: quality metrics definition and application*, in proc. of **ICEIS'05**, the 6th International Conference on Enterprise Information Systems, Miami, Florida, USA, May 2005, pp.98-103, 2005.

Fangseu Badjio E., Poulet F. : *Définition des spécificités de la fouille visuelle des données pour une évaluation de l'interaction homme machine*, in proc. of 3e Atelier Visualisation et Extraction de Connaissances, **EGC'05**, Paris, 2005, pp.7-14, 2005.

Fangseu Badjio E.: *Quality evaluation of visual data mining tools*, in proc. of **AC'05**,

the IADIS International Conference Applied Computing 2005, pp.133-138, 2005.

Fangseu Badjio E., Poulet F.: *Usability of Visual Data Mining Tools*, in proc. of **ICEIS'04**, the 6th International Conference on Enterprise Information Systems, Porto, Portugal, Apr.2004, Vol.5, pp.254-258, 2004.

Fangseu Badjio E. : *Qualité de l'Interaction Homme Machine en Fouille Visuelle de Données*, **INFORSID'04**, Biarritz, Mai 2004, 543-544.

Fangseu Badjio E., Poulet F. : *Utilisabilité d'un environnement de fouille de données*, in proc. of **SFC'03**, Meetings of the French-speaking Classification Society, Neuchâtel, Suisse, pp.117-120, 2003.

Chapitre 3 : Méthode d'inspection experte en FVD

3.1 Introduction

Dans ce chapitre, nous présentons une nouvelle technique basée sur l'ergonomie des logiciels pour une inspection experte des outils de FVD en vue de la détection de défaillances dans les outils de ce type. Rappelons brièvement que le terme ergonomie vient du grec ergon qui signifie travail et nomos qui signifie loi, règle. L'ergonomie s'intéresse à l'aménagement des systèmes personne machine c'est-à-dire aux conditions de travail de telle sorte que les utilisateurs puissent travailler dans des conditions optimales de sécurité, confort, santé, satisfaction et efficacité. Selon [Meinadier, 1991], l'ergonomie des logiciels consiste à optimiser la manière dont l'information est traitée et présentée par l'ordinateur pour correspondre aux objectifs des utilisateurs.

Un logiciel ergonomique est utile et utilisable. Un système utilisable est facile d'utilisation, adapté à tous les profils d'utilisateurs qui s'en servent et facile d'apprentissage. Il est donc nécessaire de connaître les utilisateurs. En ce qui concerne l'utilité, il doit y avoir une adéquation entre les utilisateurs et leurs tâches, le logiciel doit répondre à leurs besoins, d'où la nécessité de bien cerner la tâche des utilisateurs.

Dans notre contexte, l'ergonomie sera utilisée dans l'optique de diminuer les erreurs commises dans les systèmes existants de FVD, de réduire le temps d'apprentissage et d'améliorer la qualité d'utilisation (utilisabilité) de ces outils. La démarche ergonomique repose sur l'analyse de la situation de travail qui consiste à étudier le contexte dans lequel l'utilisateur se sert du logiciel. Il s'agit plus précisément de l'étude des phénomènes susceptibles d'être observés lorsqu'un utilisateur travaille sur l'environnement. Nous pensons qu'une méthode d'inspection, d'analyse ou d'évaluation de logiciels issue de l'analyse de la situation de travail permettra d'assurer la qualité du logiciel à toute étape de son cycle de vie.

. Nous devons insister sur le fait que les approches ergonomiques ne possèdent pas seulement des avantages. En effet, leurs limites portent essentiellement sur la portée de leur couverture durant l'analyse, l'évaluation ou le diagnostic, elles ne couvrent pour la

plupart que l'interface utilisateur. L'idée pour nous est de définir une méthode qui se serve des connaissances ergonomiques et qui puisse couvrir les interfaces utilisateur – système de FVD, système de FVD – ensemble de données, système de FVD – outils graphiques, etc.

L'objectif de ce chapitre est de définir un ensemble de directives pour le développement de systèmes de FVD de qualité. La qualité externe des logiciels de FVD a été jusqu'à très récemment relayée au second plan des préoccupations en recherche. Nous avons déjà vu dans les chapitres 1 et 2 que l'étude des facteurs humains, de l'ergonomie des logiciels, du génie logiciel peut constituer un point de départ en ce qui concerne l'amélioration de la qualité des outils de FVD.

Les directives définies dans ce travail peuvent être appliquées à toute étape du cycle de conception des logiciels de FVD. Pour mener à bien une telle réalisation, le point de départ a été l'analyse de la situation de travail en FVD qui nous a permis d'étudier le processus de FVD et d'observer par exemple que le choix de la méthode d'analyse des données (1ère et 2e variantes du modèle de Ankerst) peut ne pas être très évident pour le spécialiste des méthodes d'analyse, encore moins pour le spécialiste des données. Afin que l'analyse des données se fasse dans les meilleures conditions, il est nécessaire de prévoir des mécanismes d'aide aux utilisateurs qui contribuent aussi à l'ergonomie du logiciel. Il s'avère nécessaire que les outils de FVD soient faciles à utiliser pour permettre de mener à bien l'analyse de données. Les directives définies serviront au diagnostic des systèmes existants afin de développer des techniques de FVD acceptables par tout type d'utilisateur. En effet, le diagnostic permet de détecter les erreurs potentielles des logiciels par application soit de règles, de guides de styles, de guides de recommandation ou par évaluation heuristique.

Après un état de l'art des théories et des modèles en ergonomie dans la première partie de ce chapitre, la seconde partie s'articule autour de la modélisation de l'utilisateur. Nous montrons comment pourra être opéré sa mise en œuvre, son initialisation, son utilisation et sa maintenance. Ensuite, nous décrivons le modèle de tâche de FVD qui inclut des mécanismes de perception visuelle. Enfin, nous présentons l'ensemble de mesures d'inspection spécifiques au domaine de la FVD, nous concluons ces travaux et en perspectives nous introduisons une mise en œuvre possible du modèle de l'utilisateur.

3.2 Etat de l'art : fondements théoriques en analyse de la situation de travail

Avant de commencer cet état de l'art, il importe de souligner qu'il existe des méthodes non ergonomiques visant à l'analyse de la situation de travail. L'une des limites de ces méthodes est qu'elles ne prévoient pas le traitement des erreurs humaines, elles sont essentiellement conçues selon les critères de performance. Elles ne portent donc pas sur les caractéristiques des utilisateurs, elles décrivent le travail prescrit sans prendre en compte le travail réel. Le travail prescrit correspond à ce que l'opérateur doit faire et le travail réel à ce qu'il fait effectivement.

Pourtant, pour effectuer une tâche, l'utilisateur effectue une activité qui est

déterminée par ses caractéristiques psychologiques (expérience de la tâche, motivation à utiliser l'outil, caractère occasionnel ou permanent d'utilisation), physiologique (âge, sexe, état de fatigue, …) et psychosociologique (motivations, statut, …). Les interfaces de logiciels reposent sur les opinions et jugements des utilisateurs, d'où la nécessité d'une étude ergonomique et le choix d'une analyse basée sur l'ergonomie des logiciels.

Dans le domaine de l'ergonomie des logiciels, il existe un grand nombre de techniques susceptibles d'être utilisées en analyse de la situation de travail. Il n'existe cependant pas de techniques qui soient plus valables que les autres dans l'absolu. En effet, ces techniques peuvent se compléter et ne sont pas exclusives. Selon [ISO/TR 16982, 2002], il peut s'agir de l'observation des utilisateurs, de mesures relatives aux performances, d'incidents critiques, de questionnaires, de la pensée à haute voix, de la conception et de l'évaluation collaborative, de la méthode de créativité, des méthodes basées sur des documents, des approches basées sur des modèles, de l'évaluation par expertise et de l'évaluation automatisée. Il est donc nécessaire de procéder à une analyse de la situation en fouille visuelle de données afin de définir une méthode beaucoup plus spécifique à ce domaine.

Les progrès en informatique ont permis de contribuer significativement à l'aisance de l'activité humaine. Il ressort de différents travaux [Bastien et Scapin, 1993], [ISO, 1998] que pour favoriser l'activité, il est nécessaire de guider, conseiller et favoriser l'acquisition de connaissances par l'utilisateur en vue d'une certaine expertise relative à l'accomplissement des tâches qui lui sont soumises. La FVD, réservée dans un premier temps aux experts des méthodes d'analyse de données est désormais accessible à d'autres types d'utilisateurs. Pour aboutir à des modèles de données sur un environnement de ce type, il existe plusieurs types d'actions possibles et plusieurs types d'opérations relatives à ces actions. La pertinence des modèles de données ainsi obtenus dépend des compétences des utilisateurs. Cependant, rien n'a été fait afin que sur un tel environnement on puisse calculer et proposer des connaissances pertinentes aux utilisateurs suivant leurs profils, adaptant le processus de fouille à leurs compétences. Pourtant, des travaux [Hatcheut et al., 2005] montrent que les performances des utilisateurs varient selon le dispositif d'aide à leurs activités. Par exemple, une interface appropriée à un expert du domaine de la découverte de connaissances dans les données conviendrait très peu à un novice, à un utilisateur occasionnel ou à un spécialiste du domaine de données. Ceci dit, durant l'inspection et le diagnostic des outils de FVD, nous allons nous intéresser aux utilisateurs des environnements de FVD, à leurs compétences et aux dispositifs de FVD de façon à promouvoir le développement cognitif relatif aux interactions des utilisateurs.

Du point de vue ergonomique, la modélisation de l'interaction homme machine (en FVD) consiste à modéliser l'utilisateur, la tâche et l'application. La modélisation de l'utilisateur concerne la caractérisation de son niveau d'expertise qui peut être débutant, confirmé ou expert. Il peut aussi s'agir de la modélisation de ses processus cognitifs : facilité d'apprentissage, connaissance, croyances ou des processus psychologiques.

En ce qui concerne la représentation cognitive des utilisateurs, elle peut se faire à travers le modèle du processeur humain, la théorie de l'activité et de l'action. Il s'agit plus précisément du dispositif nécessaire aux utilisateurs dans l'accomplissement de leurs

tâches. La théorie de l'activité en général et plus particulièrement la théorie de l'action [Norman, 1986] qui en découle constituent le cadre théorique de cette modélisation de l'utilisateur.

En effet, la théorie de l'activité [Kaptelinin, 1995], [Leont'ev, 1978] développée par des théoriciens soviétiques est un ensemble de principes de base qui constituent un système conceptuel général. L'activité est dirigée par un objet (sa motivation). Divers courants de pensées régissent la théorie de l'activité. Nous pouvons citer par exemple le courant qui stipule qu'un individu face à son activité se crée des représentations et les modifie, ce qui peut se traduire dans notre contexte par le fait qu'un individu pourrait après une séquence de fouille visuelle de données acquérir des heuristiques lui permettant de manipuler d'autres systèmes de fouille visuelle. Un autre courant de pensée, la théorie de l'action [Norman, 1986], issue de la théorie de l'activité décompose en hiérarchies l'activité. On a l'activité proprement dite qui est en relation avec des buts, des motivations et les actions qui peuvent servir à plusieurs activités et qui s'effectuent par des opérations, les opérations étant des actions élémentaires. En nous basant sur ces différentes théories, l'idée est de tenir aussi compte du caractère improvisé du comportement humain [Suchman, 1987] dans les différentes actions qu'il peut accomplir.

Notre objectif est donc de définir une méthode qui permette de relever les besoins des utilisateurs en FVD durant un diagnostic ou une inspection des outils de ce type. Nous allons aussi proposer une modélisation de l'utilisateur qui permette de prendre en compte l'évolution de ses connaissances, de ses compétences et du contexte de l'action. Le modèle de l'utilisateur permet de l'aider dans le cadre de ses activités et reflète les caractéristiques cognitives de celui-ci. Les recherches du domaine des tuteurs intelligents dans les années 1980 ont permis l'essor du modèle utilisateur qui est aussi très utilisé en IHM et fait généralement référence à trois différents concepts, il peut s'agir :

- de la représentation faite par l'utilisateur d'un logiciel informatique,
- des connaissances dont dispose le logiciel informatique de son utilisateur,
- de l'ensemble de connaissances que devrait avoir un utilisateur pour pouvoir utiliser le logiciel de façon optimale.

Dans l'état actuel des recherches en ECD en général, les modèles de données ne dépendent pas du profil des utilisateurs. Tous les utilisateurs (experts ou non) qui désirent retrouver des pépites de connaissances dans un ensemble de données utilisant le même algorithme et les paramètres identiques obtiennent les mêmes résultats. Pourtant, non seulement dans une organisation nécessitant un système d'ECD pour l'exploitation de ses données il existe des utilisateurs de type différent mais aussi, les modèles de données issus de la fouille dans de grands ensembles de données s'avèrent différents et se distinguent en compréhensibilité, pouvoir prédictif, lisibilité et coût de calcul. La prise en compte des caractéristiques et compétences des utilisateurs s'impose.

En guise d'exemple, si on se situe dans le cadre beaucoup plus restrictif des méthodes de visualisation pour l'exploration des données, un utilisateur peut préférer par exemple les matrices 2D et un autre utilisateur les coordonnées parallèles. Un système auquel est couplé un modèle utilisateur pourra décider implicitement de la méthode de

visualisation adéquate à un type d'utilisateur en se basant sur son modèle. Ainsi, la familiarité de l'utilisateur avec l'outil et le système cognitif de raisonnement en rapport avec les propriétés sémantiques des méthodes graphiques seront aisément pris en compte.

Dans cette étude, l'analyse de la situation de travail en FVD a pour but de définir des critères ergonomiques spécifiques au domaine de la FVD pour l'évaluation des systèmes de ce type. La section suivante présente une classification de méthodes basées sur des directives pour l'évaluation des logiciels. L'objectif ici est de pouvoir situer l'évaluation à base de critères dans le champ de recherche lié à l'évaluation des logiciels.

3.3 Evaluation de logiciels à l'aide de directives

On distingue deux méthodes d'évaluation de logiciels à l'aide de directives : l'évaluation par inspection et l'évaluation empirique [Senach, 1990]. Pour l'évaluation par inspection, il existe trois approches :

- l'inspection est fondée sur des procédures de représentation graphique des problèmes d'utilisabilité,
- l'inspection est basée sur des modèles prédictifs des performances de l'utilisateur,
- l'inspection est faite par des connaissances expertes (inspection heuristique). Le principal intérêt de l'inspection heuristique est son degré de détail. En effet, cette méthode garantit une analyse exhaustive de l'ensemble du logiciel.

L'évaluation empirique consiste à interpréter les performances des usagers, à qui l'on prescrit une tâche et plus généralement à interpréter leurs comportements, attitudes ou opinions. Cette méthode d'évaluation est aussi appelée enquête d'usage.

Dans le cadre de nos travaux, nous nous proposons de définir une méthode d'inspection des logiciels de FVD (dans le présent chapitre) et une méthode d'analyse et d'évaluation empirique de ces logiciels (chapitre 4). L'idée est de pouvoir utiliser ces différentes méthodes dès les phases en amont du processus de conception.

La méthode d'inspection sera destinée aux experts (concepteurs, analystes de données) tandis qu'une extension de la méthode d'analyse (méthode d'évaluation) servira au diagnostic des environnements de FVD par des utilisateurs finaux.

Ces méthodes font appel aux connaissances de la situation de travail en FVD. La section suivante est dédiée à l'analyse de la situation de travail en FVD.

3.4 Analyse de la situation de travail en FVD

L'objectif de l'analyse de la situation de travail est de mieux connaître les utilisateurs et les tâches de FVD afin de définir les directives qui conviennent le mieux à l'inspection, l'analyse et l'évaluation des outils de ce type. Le point de départ d'une telle réalisation est une analyse préalable des besoins qui nous a permis non seulement de connaître les utilisateurs mais aussi de proposer un modèle pour supporter l'activité du « fouilleur de

données ».

Pour les besoins de modélisation, le choix a été porté sur l'utilisation des diagrammes de séquences UML pour représenter les différents modèles qui illustrent nos travaux. En effet, les éléments de formalisation des diagrammes de séquences UML apportent une vue intéressante du système modélisé. Chaque objet contenu dans un rectangle du diagramme UML représente un élément du système, par exemple le « fouilleur de données ». L'échange de messages entre deux éléments du système est représenté par une flèche qui permet d'indiquer le sens de la lecture. Cette flèche est explicitée par un verbe ou une forme verbale.

3.4.1 Modèle utilisateur de FVD

Il ressort de l'analyse préalable des besoins, que les utilisateurs potentiels d'un environnement de FVD peuvent être soit experts du domaine des données, soit experts en analyse de données, ils peuvent aussi appartenir à une organisation quelconque nécessitant des outils d'analyse de données pour le traitement de leurs données. La FVD nécessite des connaissances diverses. Il s'impose la nécessité de prodiguer de l'aide et des conseils aux différents utilisateurs sachant que les utilisateurs acquièrent des compétences en fouille tout au long des traitements.

L'aide à promulguer aux utilisateurs dépend de leurs profils et de leurs connaissances. Afin d'intégrer le profil et les connaissances de l'utilisateur tout au long du processus de fouille, la proposition faite dans ce travail est qu'à l'initialisation du système de fouille, le modèle de l'utilisateur corresponde à ses compétences générales ou son profil. Ce modèle doit être maintenu par des informations issues des interactions de l'utilisateur sur l'environnement de fouille. De façon concrète, le modèle utilisateur proposé met en relation les différentes dimensions du modèle de la tâche, du contexte, des stratégies possibles d'organisation des traitements et de l'aide à promulguer aux différents utilisateurs.

L'idée ici est de pouvoir adapter de façon implicite non seulement les modèles obtenus par la fouille mais aussi le processus de fouille de données selon le type et les compétences de l'utilisateur. Pour la maintenance de ce modèle, les informations ayant trait à l'historique des interactions de l'utilisateur sur l'environnement de fouille sont utilisées. L'initialisation du modèle utilisateur est la première étape nécessaire à cet effet, nous la décrirons dans le prochain paragraphe.

3.4.1.1 Acquisition du modèle utilisateur de FVD

L'acquisition du modèle utilisateur se fait lors du premier contact de l'utilisateur avec le logiciel. Cette phase a pour but d'identifier l'utilisateur ainsi que ses besoins. A cet effet, une adaptation des spécifications de IMS LIP (Information Management System – Learning Information Package) [IMS-LIP, 2003] peut être utilisée. Les informations susceptibles d'être retenues de cette spécification sont : l'identifiant, les buts visés par la fouille visuelle, les compétences, le poste occupé (activités dans l'entreprise), les informations de sécurité, les qualifications qui regroupent les certificats, les licences, les hobbies, le statut et le rôle.

Le modèle utilisateur ainsi créé doit pouvoir être mis à jour tout au long des interactions de l'utilisateur sur l'environnement de fouille. Afin de maintenir ce modèle, nous proposons l'utilisation d'un système multi-agent (SMA). Partant de la définition d'un agent logiciel de [Ferber, 1995], nous pouvons définir un agent comme une entité autonome, c'est-à-dire capable d'agir sur elle-même et sur son environnement en vue de réaliser ses objectifs. L'agent dispose d'une représentation partielle de cet environnement. Dans un environnement multi-agent, l'agent peut communiquer avec les autres agents, son comportement est la conséquence de ses observations, de ses connaissances et des interactions avec d'autres agents.

L'utilisation des SMA comme support à la modélisation de l'utilisateur consiste à définir une société d'agents et les interactions possibles entre eux. Le premier avantage d'une telle approche repose sur cette définition des interactions entre agents qui permet d'opérer des traitements en parallèle. Le second avantage fait référence à l'autonomie des agents. Comme indique la définition d'un agent, il s'agit d'une entité capable d'agir sur elle-même. L'acquisition de connaissances concernant les préférences et les aptitudes des utilisateurs tout au long des traitements pourra être faite sans aide extérieure par un agent. Les prises de différentes décisions (consistant à proposer ou non suivant les cas une assistance à l'utilisateur) peuvent aussi être déléguées à des agents. L'aspect adaptatif et l'autonomie des SMA permettent au fur et à mesure des traitements d'acquérir des connaissances relatives à la performance des utilisateurs et de mettre à jour les informations de leur modèle. Après l'acquisition du modèle utilisateur, il est nécessaire de le maintenir. Cette étape fait l'objet de la prochaine section.

3.4.1.2 Maintenance du modèle utilisateur de FVD

Afin de mieux situer le contexte dans lequel s'opère la maintenance du modèle utilisateur, nous proposons d'abord un modèle de séquence des instructions exécutées sur l'environnement de FVD.

Modélisation du système de fouille visuelle de données

La figure représentée ci-dessous matérialise donc la spécification des différentes interactions sur l'environnement de FVD.

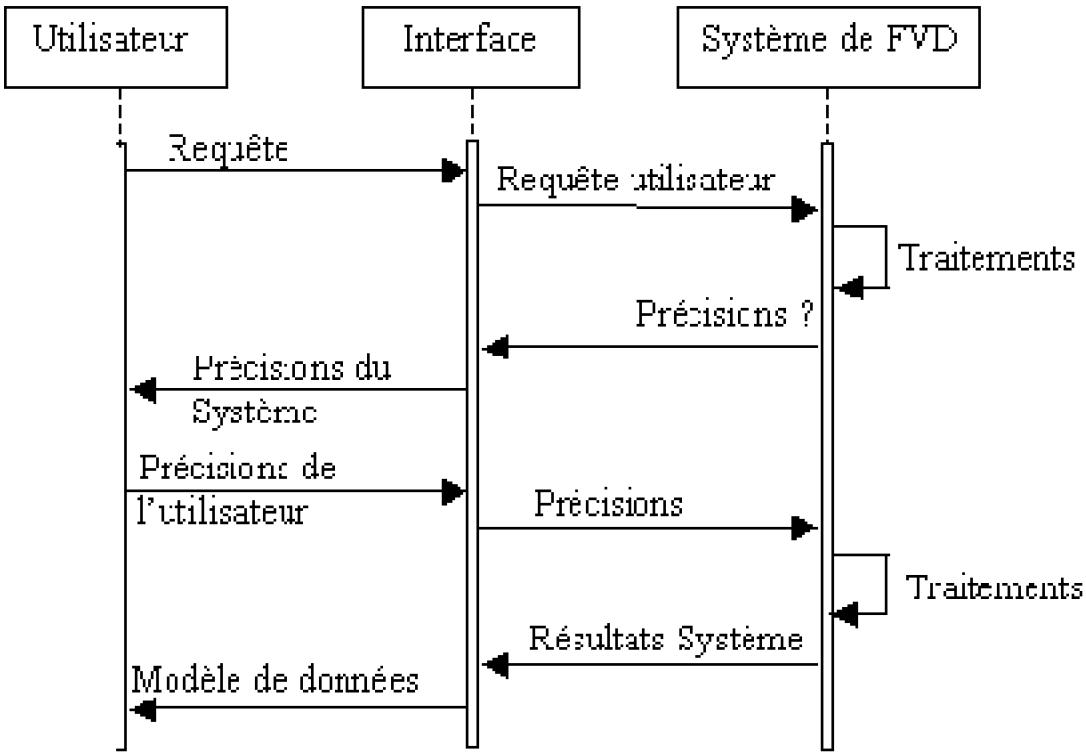


Figure 3.1 Diagramme de séquences de l'activité de FVD

Trace de scénario

L'utilisateur agit sur une interface. L'entité chargée de la gestion de l'interface transmet les requêtes de l'utilisateur au système de FVD. Le système de FVD traite la requête de l'utilisateur, au cours de ce traitement, il peut s'avérer nécessaire que l'utilisateur fournisse des précisions au système.

Tout comme pour un problème de décision, l'utilisateur perçoit le problème de FVD à travers la représentation graphique des données, sachant qu'il peut ne pas savoir au départ quel est le chemin à parcourir qui va de ce point de départ vers la solution. Il se fixe alors des buts à atteindre par rapport à la représentation des données. Cette étape est suivie de la planification des actions à accomplir et de l'évaluation des résultats.

Tout au long de ces différentes interactions, un apprentissage visant la mise à jour du modèle utilisateur peut s'opérer comme l'indique la section suivante qui présente l'approche utilisée pour la maintenance du modèle utilisateur.

Phase de maintenance du modèle utilisateur de FVD

Le modèle de l'utilisateur proposé est maintenu par un SMA greffé au système de FVD. En effet, il est possible de sauvegarder, de mettre a jour et de supprimer des informations du modèle. A cet effet, une base de cas permet de répertorier les différents traitements effectués par l'utilisateur sur l'environnement de fouille. Le diagramme de séquence du système final se présente comme suit (figure 3.2) :

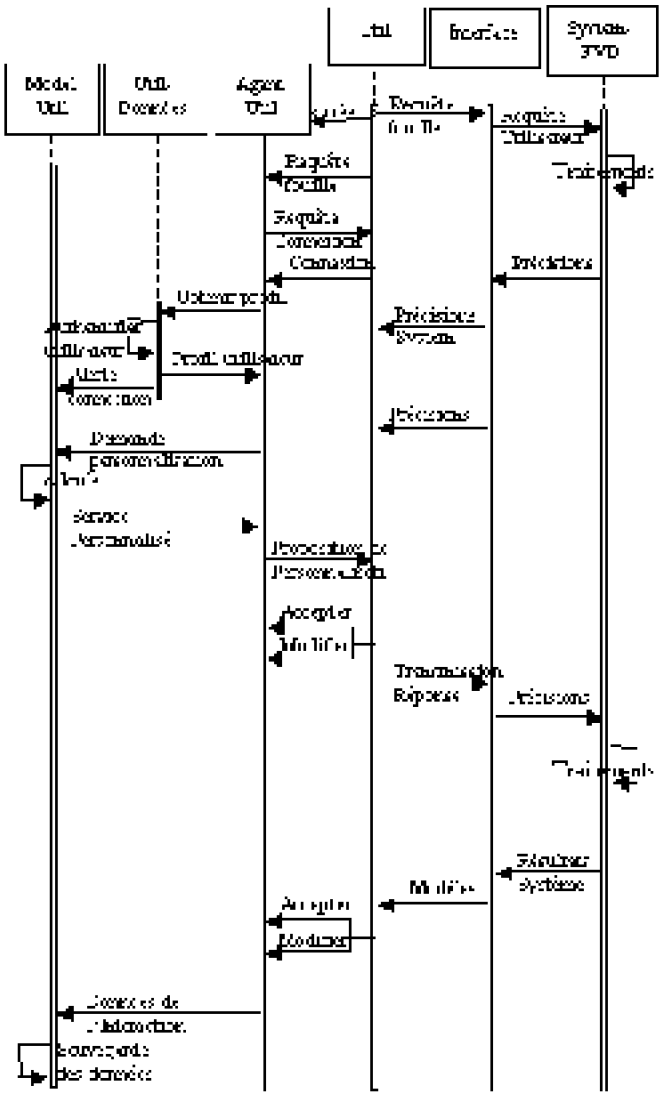


Figure 3.2 Diagramme de séquences de la phase de maintenance du modèle utilisateur

Trace de scénario

La requête de FVD exprimée par l'utilisateur sur l'environnement entraîne la création d'un agent (AgentUtil) qui se met à l'écoute de toutes les actions de l'utilisateur sur l'environnement de fouille et lui demande de s'identifier.

La base de cas des interactions faites par l'utilisateur permet d'effectuer des calculs de similitudes et de lui proposer de l'aide tout au long des différentes étapes de l'activité de fouille.

L'agent (AgentUtil) propose cette aide à l'utilisateur qui peut soit l'accepter ou intégrer son contenu dans son activité. La décision de l'utilisateur est transmise à l'entité chargée de la gestion de l'interface qui la transmet au système de fouille visuelle de données qui effectue les traitements et transmet les modèles des données obtenus à l'utilisateur.

Les informations suivantes sont nécessaires à la gestion du modèle utilisateur : le

chemin parcouru par l'utilisateur de l'étape initiale jusqu'à l'accomplissement de sa tâche, le nombre d'accès au module d'aide et les choix opérés quant à l'aide contextuelle promulguée.

L'implémentation et le déploiement dans un système du modèle utilisateur ainsi développé permettent de répondre plus explicitement aux besoins effectifs des utilisateurs finaux.

3.4.1.3 Discussion

La modélisation de l'utilisateur permet de mieux gérer ses connaissances et de lui proposer des modèles de données adaptées à ses besoins. Cette étape permet aussi de relever les points essentiels nécessaires à la réalisation d'un système de FVD facile et agréable à utiliser. C'est-à-dire un système qui puisse s'adapter de façon implicite aux besoins des utilisateurs. L'analyse, l'évaluation et l'inspection des outils de FVD permettront de voir dans quelles mesures l'utilisateur est assisté dans sa tâche et ce facteur constituera un point important de l'étude de la qualité de ces outils. Mais, en plus de cette modélisation de l'utilisateur, il est indispensable d'effectuer une modélisation de la tâche allouée aux utilisateurs. Pour illustrer la nécessité de l'analyse de la tâche de FVD, couplée à l'analyse des utilisateurs, nous allons nous référer aux travaux de [Hoc, 1991]. En effet, l'auteur remarque que en conception de systèmes informatiques, les experts des systèmes à concevoir participent à l'analyse, à la conception et au développement de ces systèmes. Mais la plupart du temps, l'utilisation du système final revient à un opérateur dont l'expertise est variable. La responsabilité du succès ou de l'échec de la tâche reviendra à cet opérateur. Il s'avère nécessaire que les modes de raisonnement du système ne soient pas étrangers aux utilisateurs finaux. Il faut à cet effet envisager la conception de systèmes avec une modélisation de l'expertise humaine dans le domaine et une modélisation de l'opérateur.

L'analyse de la tâche qui œuvre à la compréhension du travail de FVD fait l'objet du paragraphe suivant. En effet, cette analyse permet de détecter des besoins pertinents en assistance.

3.4.2 Modèle de tâche de FVD

Le modèle de tâche permet de définir tous les pré requis nécessaires à la bonne marche d'une application. En effet, l'information contenue dans le modèle de tâche permet d'identifier toutes les interactions et les besoins en présentation et traitement des données dans l'environnement. A travers le modèle de tâche de FVD, nous voulons étudier le support ou l'aide à apporter aux utilisateurs de FVD.

L'idée dans cette partie est de prendre en compte le contexte de la tâche, les objectifs et les actions de tout type d'utilisateur pour une assistance à la conception du modèle des données. Par rapport au modèle utilisateur, en fouille de données, les recherches jusqu'à l'heure actuelle ont permis d'exploiter beaucoup plus le modèle de la tâche [CRISP-DM, 2000]. La tâche peut être définie comme un but à atteindre dans des conditions déterminées.

La section 1.3.1 du premier chapitre de ce mémoire présente le modèle de tâche de FVD décrit par Ankerst. Afin d'aboutir au modèle des données dans un environnement de FVD, les utilisateurs procèdent par plusieurs étapes, à savoir :

1. sélection des données à exploiter,
2. passage à l'étape 3 ou visualisation des données,

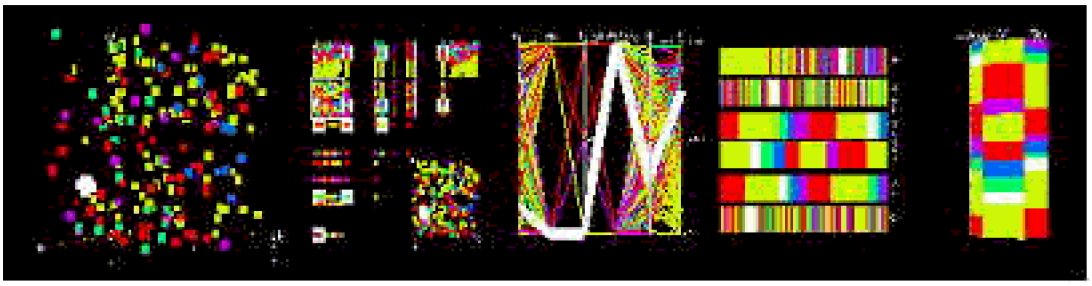


Figure 3.3 Visualisation de données

1. sélection du module d'analyse parmi ceux proposés par le système, l'environnement peut disposer de méthodes d'analyse de données automatiques et interactives. Pour la fouille interactive par exemple, l'utilisateur devra à partir de la représentation des données de la figure 3.3, sélectionner la représentation matricielle et procéder à des coupes successives afin d'aboutir à une partition pure comme l'indique la figure 3.4. Pour chaque coupe (ligne séparatrice représentée en blanc), l'utilisateur essaye d'isoler la sous partition la plus pure.

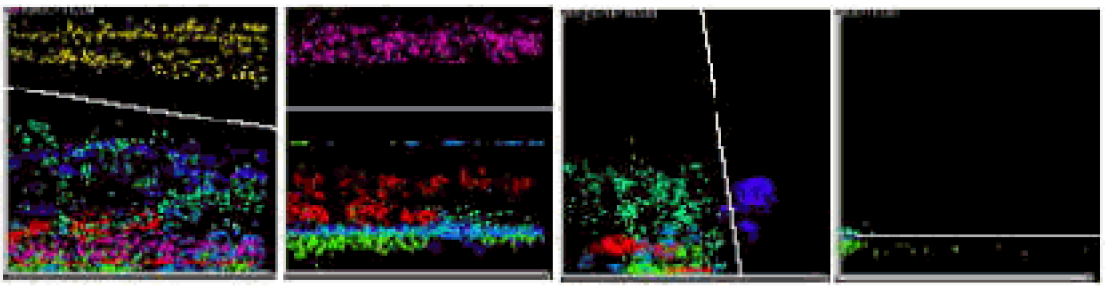
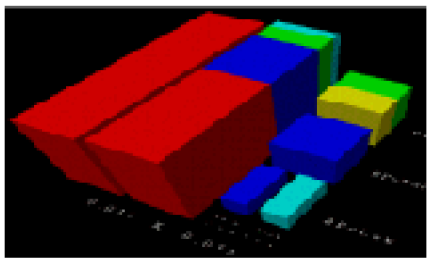


Figure 3.4 Etapes de la fouille interactive

1. visualisation des résultats,



```
petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
|  petalwidth <= 1.7
|  |  petallength <= 4.9: Iris-
|  |  |  petalwidth <= 1.5: Iris-
.0)
|  petalwidth > 1.7: Iris
virginica (46.0/1.0)
```

(1) (2)

Figure 3.5 Résultats de la FVD : (1) représentation 3D, (2) arbre de décision textuel

1. évaluation des résultats (qui doivent être facilement interprétables), suivie d'un éventuel retour à l'étape 1 ou 2,
2. exploitation des résultats considérés comme des connaissances nouvelles, retour à l'étape 1 ou arrêt.

A ce niveau, il semble important de revenir sur le fait que la FVD est basée sur des primitives graphiques. La visualisation des données nécessite de donner une matérialisation visuelle à des entités qui ne possèdent pas nécessairement une géométrie et une interprétation de cette matérialisation afin de mettre en avant certaines de leurs propriétés. L'utilisateur perçoit la forme des données, analyse les variations de couleurs et les partitions de données pures à travers une représentation graphique. Il est donc primordial que l'utilisateur sache déceler les différentes propriétés d'une représentation graphique qui lui est présentée, d'où l'importance pour lui de maîtriser les caractéristiques tant sémiologiques que cognitives du système des signes et des composantes sur lesquelles sont construites les représentations graphiques. Le risque est grand pour un utilisateur non expert du domaine de la visualisation de ne pas se retrouver dans ces systèmes de signes. L'idéal serait de lui proposer des mécanismes d'aide. Dans cette optique, l'idée de ce travail est de s'intéresser durant l'analyse et le diagnostic des systèmes de FVD plus particulièrement aux structures des représentation des données (et donc aux capacités humaines en reconnaissance de formes). Concrètement, il s'agit de voir dans quelles mesures les techniques mises en œuvre permettent de capter l'attention des utilisateurs sur des points saillants de la représentation graphique sachant que cette étape contribue à la perception et à la cognition. La perception étant indissociable de l'action [Hascoët et Beaudoin-Lafon, 2001], on va aussi s'intéresser à l'interaction qui fait suite à la phase de cognition dans le processus de FVD et du raisonnement humain en général. En effet, le processus de raisonnement humain repose sur l'alternance des phases de définition des objectifs à atteindre et des phases de recherche et de test des solutions potentielles. L'étape 3 du modèle de FVD présenté dans le paragraphe 3.4.2 montre que l'utilisateur doit procéder au choix de la méthode d'analyse de données à utiliser par exemple. Il serait très intéressant de l'aider pour ce faire. De plus, les données disponibles à travers le monde ne cessent d'augmenter, il serait aussi intéressant de pré traiter ces données avant l'application des outils de FVD afin de faciliter l'activité des fouilleurs de données.

A la lumière de ce qui précède, lors du diagnostic ou de l'inspection des outils de FVD, nous allons tout d'abord nous intéresser aux caractéristiques des représentations graphiques qui sont décelées facilement. En effet, l'être humain est capable de gérer les variations dans les représentations graphiques par les mécanismes d'attention.

L'analyse de la tâche en FVD nous a aussi permis de constater qu'il est nécessaire d'étudier les mécanismes attentionnels et la perception visuelle en FVD pour réduire la charge cognitive des utilisateurs. Les propriétés de ce point de vue, pouvant contribuer à une assistance aux concepteurs des modèles de données utilisant la visualisation comme canal de communication sont décrites dans cette section.

Fort des différentes observations issues de l'analyse des utilisateurs et de la tâche de FVD, nous allons à présent introduire des recommandations pour la FVD dont le non respect pourra être interprété comme un problème de qualité. La méthode proposée peut être utilisée à tout étape du cycle de conception du logiciel.

3.5 Inspection experte : directives pour la FVD

L'analyse de la situation de travail en FVD, une analyse des travaux de Nielsen [Nielsen, 1993a], des critères de Bastien et Scapin [Bastien et Scapin, 1993], des différentes normes ISO pour la qualité des logiciels, des différents guides de styles et recommandations de qualité de logiciels [Preece, 1993], [Nielsen et Mollich, 1990], [Mayhew, 1992], [Galitz, 1996], [Fernandes , 1995], etc. constituent les fondements des directives pour l'inspection des environnements de FVD présentées dans cette section et adhérent aux spécificités de ce domaine.

3.5.1 Guidage

3.5.1.1 Définition

Le guidage concerne les moyens mis en œuvre pour conseiller, orienter, informer et conduire l'utilisateur lors de ses interactions avec l'ordinateur [Bastien et Scapin, 1993].

Un bon guidage facilite l'apprentissage et l'utilisation du système en permettant à l'utilisateur de savoir à tout moment où il se trouve dans une séquence d'interactions, ou dans l'accomplissement d'une tâche, de connaître les actions permises ainsi que leurs conséquences et d'obtenir de l'information supplémentaire.

3.5.1.2 Exemple concret de mise en œuvre du guidage en FVD

Les progrès réalisés dans le domaine de l'ECD ont permis d'intégrer plusieurs méthodes d'analyse de données dans un même environnement. Le processus de FVD prévoit qu'après la sélection des données à traiter, l'utilisateur choisisse une méthode d'analyse des données. Le choix de la méthode d'analyse des données à exécuter n'est pas aisé. La figure 3.6 présente un exemple concret du choix de la méthode d'analyse des données dans l'environnement WEKA [Witten et Eibe, 2005], un logiciel libre de l'université de Waikato qui contient une collection d'algorithmes d'apprentissage machine, de fouille de

données et de FVD. Dans ce logiciel, comme l'indique la figure, il existe des modules de prétraitement des données, de classification, de clustering, de découverte de règles d'association, de sélection d'attributs et de visualisation.

La première étape de traitement avec WEKA consiste au prétraitement des données, ensuite l'utilisateur choisit un module parmi les modules disponibles (classification, clustering, découverte de règles d'association, etc.) après ce choix, l'utilisateur choisit une méthode parmi celles disponibles dans le module choisit. Si par exemple le module choisi par l'utilisateur est Classifier et le sous module Rules (figure 3.6), sans connaissance préalable relative aux performances des 10 algorithmes de ce module, il aura à les exécuter tous avant de choisir le plus approprié ou en choisira un sous ensemble de manière aléatoire. Comme dans la plupart d'outils de fouille, aucune aide n'est mise à la disposition des utilisateurs afin qu'ils puissent opérer convenablement ce choix.

Une mise en œuvre du guidage serait d'aider les utilisateurs à opérer ce choix (chapitre 5).

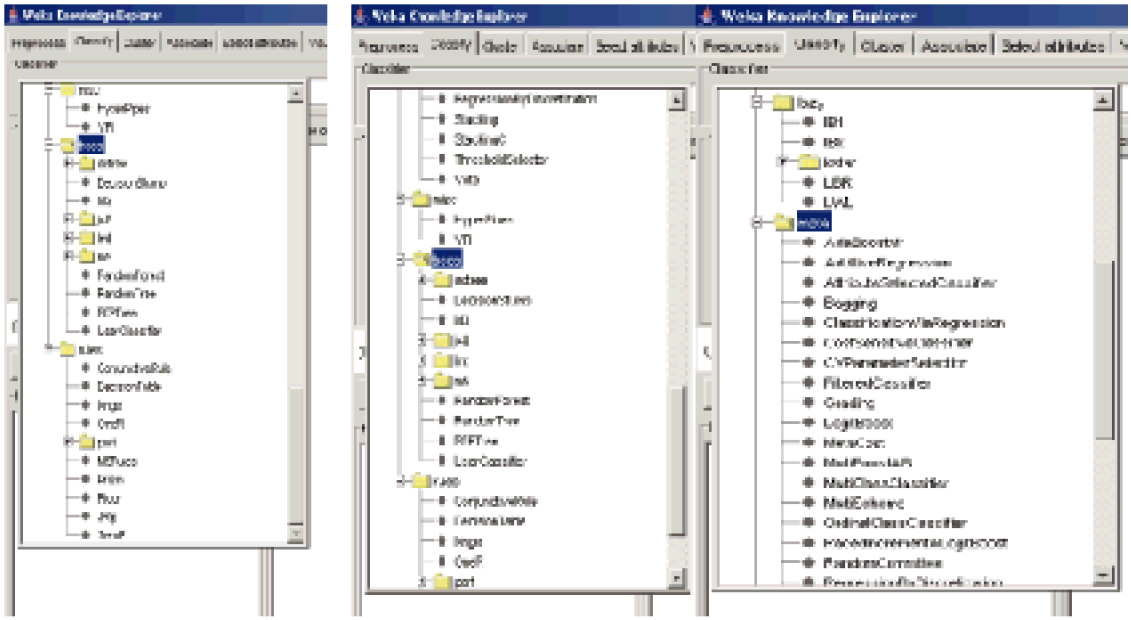


Figure 3.6 WEKA

3.5.2 Gestion des erreurs

3.5.2.1 Définition

La gestion des erreurs est l'ensemble de moyens permettant d'une part d'éviter ou de réduire les erreurs, d'autre part de les corriger lorsqu'elles surviennent.

3.5.2.2 Mise en œuvre

Il s'agit ici de mettre en place des moyens pour détecter et prévenir les erreurs. Par exemple, toutes les actions possibles sur l'interface doivent être envisagées et plus particulièrement les appuis accidentels des touches du clavier afin que les entrées non

attendues soient détectées. Au cas ou par exemple l'exécution de la méthode d'analyse de données choisie par l'utilisateur n'arrive pas à terme, il faut donner la possibilité à l'utilisateur d'exécuter un autre algorithme sans toutefois que le système ne plante. Pour la possibilité d'exécution d'un autre algorithme par l'utilisateur, le module de choix de l'algorithme à exécuter évoqué au niveau de la mise en œuvre du guidage doit pouvoir retourner non seulement l'algorithme le plus efficace à la résolution du problème mais aussi la liste d'algorithmes classés suivant des critères d'évaluation d'algorithmes.

3.5.3 Adaptabilité

3.5.3.1 Définition

L'adaptabilité est la capacité du système à s'adapter aux besoins de l'utilisateur sans intervention explicite de sa part ou sa capacité à réagir selon le contexte et selon les besoins et les préférences des utilisateurs.

3.5.3.2 Mise en œuvre

Pour la mise en œuvre de ce critère, nous avons pensé à la possibilité pour l'utilisateur de personnaliser son interface. La personnalisation de l'interface utilisateur a pour but la prise en compte des stratégies et ou des habitudes de travail de l'utilisateur en fonction de la tâche à exécuter. Nous avons aussi pensé au développement de moyens disponibles pour la prise en compte du niveau d'expérience de l'utilisateur (débutant, expérimenté, occasionnel) ainsi que son profil.

3.5.4 Réutilisation des données d'apprentissage

3.5.4.1 Définition

La réutilisation des données d'apprentissage consiste à considérer les données en sortie du système de fouille comme étant des données d'entrée pour d'autres procédures ou la même procédure.

Ce critère trouve une de ses justifications dans le fait que l'on reconnaît une bonne méthode d'analyse de données [Han et Kamber, 2001] à sa capacité à fournir un modèle de données quelque soit l'étape de traitement dans lequel il se trouve.

Ce critère contribue ainsi à la réalisation de l'efficacité (capacité du système à atteindre un objectif donné).

3.5.4.2 Mise en œuvre

En fouille de données, lorsque l'ensemble de données à traiter est volumineux, le temps nécessaire à la découverte de connaissances est important. En l'état actuel des compétences d'outils d'ECD, lorsqu'un utilisateur arrête volontairement ou non un processus de fouille avant son terme, il doit tout recommencer à l'état initial. Il s'agit ici de lui permettre de continuer dans le processus sans toutefois avoir à recommencer à l'étape

initiale.

3.5.5 Multiplicité du rendu

3.5.5.1 Définition

Nous définissons la multiplicité du rendu comme étant la capacité du système à fournir plusieurs choix (alternatives) de solution pour un problème donné.

Pour justifier ce choix de critère, nous pouvons nous baser sur des résultats tels que le No free lunch theorem [Wolpert et Macready, 1997], les besoins en flexibilité des systèmes personne-machine.

3.5.5.2 Mise en œuvre

Il existe de nombreuses techniques de représentation graphique de données : techniques de base, géométriques, symboliques, graphes, hiérarchiques, en trois dimensions. Tout le monde s'accorde sur le fait qu'aucune de ces méthodes n'est meilleure que les autres dans tous les cas de figure. Pour un même ensemble de données, il peut s'agir concrètement de prévoir plusieurs méthodes de visualisation possibles.

3.5.6 Aide en ligne

3.5.6.1 Définition

L'aide en ligne concerne la documentation mise à la disposition de l'utilisateur.

3.5.6.2 Mise en œuvre

Une mise en œuvre de ce critère peut être le développement de menus contextuels visant à renseigner l'utilisateur, lui fournir des explications associées à la méthode de visualisation utilisée ou une aide à la décision pour le choix de critères (paramètres) nécessaires à la bonne exécution de sa tâche de fouille.

3.5.7 Feedback

3.5.7.1 Définition

Après l'accomplissement d'une action, une réponse doit être fournie à l'utilisateur le renseignant sur l'action accomplie et sur son résultat, ceci, avec un délai de réponse approprié et homogène selon les types de transactions.

3.5.7.2 Mise en œuvre

Le processus de fouille de données peut s'avérer long, surtout lorsque les données sont volumineuses et complexes. Une information indiquant à l'utilisateur que les traitements sont en cours et l'état d'avancement des traitements devrait être fourni à l'utilisateur.

3.5.8 Plasticité

3.5.8.1 Définition

La plasticité est la capacité du système à s'adapter aux variations des ressources interactionnelles et de calcul (par exemple les variations dans les ensembles de données traités, la prise en compte d'un modèle partiel des données obtenu en cours de traitements) tout en conservant la continuité ergonomique.

3.5.8.2 Mise en œuvre

Pour la mise en œuvre du critère de plasticité, l'accent peut être porté sur le passage d'une étape à l'autre du cycle de fouille de données qui doit être perceptible par le système ainsi que le passage à l'analyse d'une base de données différente de celle utilisée au cours de l'exécution précédente.

3.5.9 Curabilité

3.5.9.1 Définition

Capacité pour l'utilisateur à corriger une situation non désirée.

3.5.9.2 Mise en œuvre

Le nombre d'erreurs enregistrées lors de l'exécution de l'outil, le temps nécessaire à la correction des erreurs sont des facteurs pris en compte habituellement au cours de l'évaluation de l'utilisabilité des IHM. Le développement de moyens de curabilité des erreurs permet de mieux répondre aux attentes des utilisateurs et des tests d'utilisabilité.

3.5.10 Conclusion

Les critères et les mises en œuvre proposés dans la section ci-dessus peuvent permettre à des utilisateurs d'analyser et d'améliorer les outils de FVD. Ceci se fera de plusieurs façons, par exemple en répondant aux besoins d'aide à donner aux utilisateurs d'environnements de FVD non spécialistes des méthodes d'analyse de données, ce qui évitera des retours sur conception qui engendrent des pertes de temps et un coût considérable de production d'outils sans toutefois garantir les performances.

Les critères ainsi présentés émanent d'une adaptation des critères existants ayant été éprouvés. A ce niveau, le problème que pose ces critères réside dans leur non validation expérimentale. Le paragraphe suivant présente un essai de validation expérimentale de ces critères.

3.6 Conclusion

L'objectif du travail décrit dans ce chapitre était de définir une méthode basée sur

l'ergonomie des logiciels pour une inspection experte des outils de FVD. A cet effet, nous avons procédé à des analyses de la situation de travail en FVD. Au fil de ces analyses, il nous a semblé important de proposer un modèle utilisateur qui puisse être implémenté et qui serve à aider les utilisateurs. La figure 3.7 présente une matérialisation concrète de la phase d'acquisition du modèle utilisateur. Une perspective de ce travail serait d'intégrer ce modèle de l'utilisateur dans les outils de FVD et de tester sa valeur ajoutée.

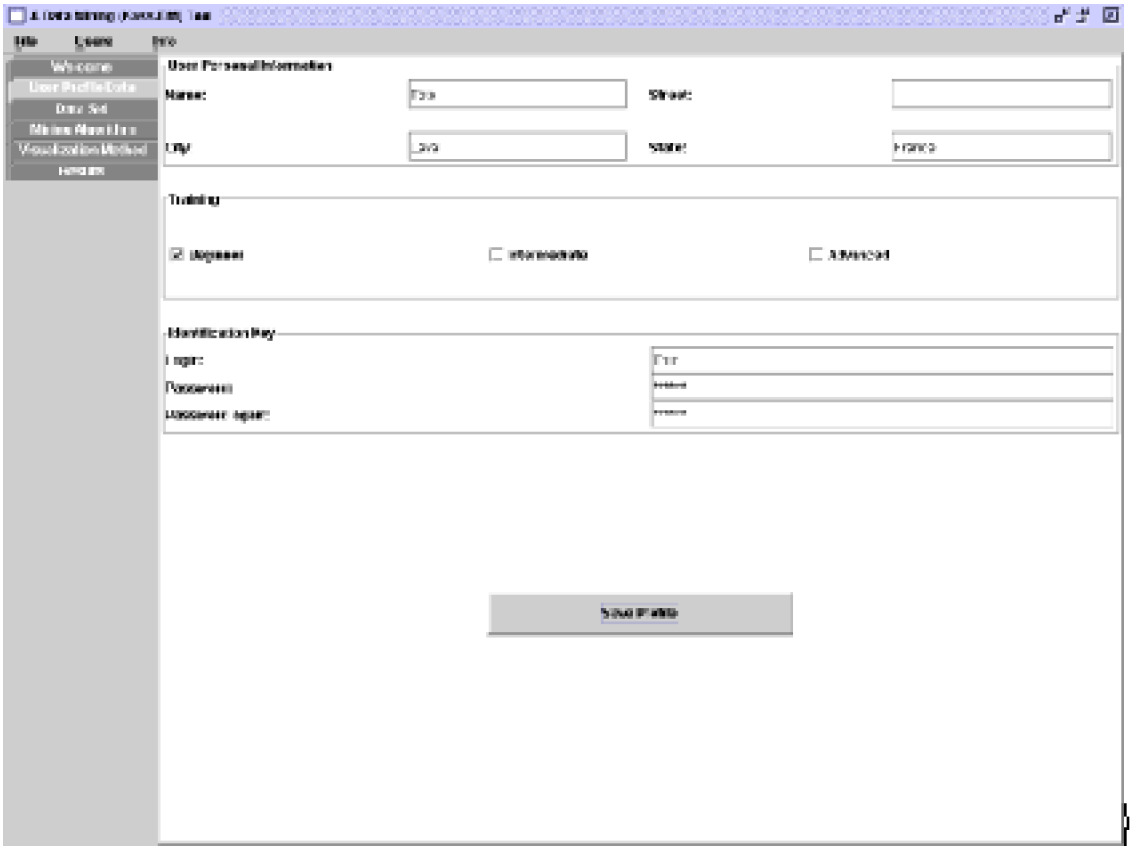


Figure 3.7 Acquisition du modèle utilisateur

La seconde perspective concerne la standardisation des différents critères d'inspection répertoriés dans ce chapitre. Il s'agit plus particulièrement d'une validation expérimentale par un ensemble d'experts des directives proposées. Ces experts utiliseront à cet effet leurs connaissances du domaine pour valider ou invalider les directives, permettant de déterminer à quel point les outils de FVD satisfont les besoins des utilisateurs et les exigences fonctionnelles.

Pour les besoins de cette validation, nous avons défini un questionnaire (tableau 3.1)

en vue d'un éventuel sondage des experts. L'idée ici est qu'ils puissent donner leur avis quant à l'impact réel des critères recensés sur la qualité d'un environnement de FVD.

Les directives définies dans ce chapitre sont destinées aux experts qui développent ou procèdent au choix d'outils d'analyste de données. Il importe aussi d'obtenir le point de vue des utilisateurs finaux d'outils de FVD. A cet effet, dans le chapitre 4, nous allons nous intéresser aux différents moyens susceptibles de favoriser une telle préoccupation.

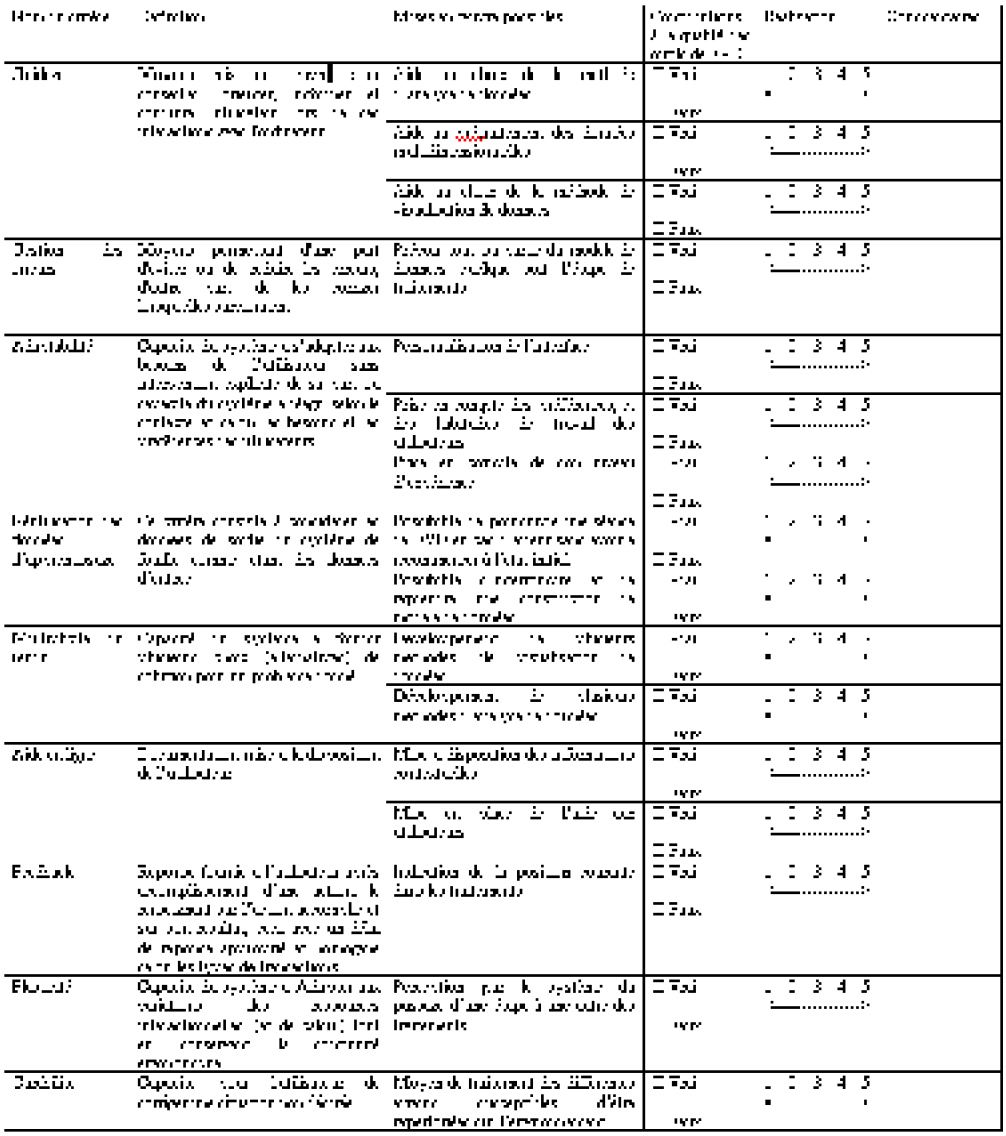


Tableau 3.1 Questionnaire pour sondage des experts en analyse de données quant à la validité scientifique des directives définies

Chapitre 4 : Métriques et mesures de qualité en FVD

4.1 Introduction

Les méthodes d'évaluation de la qualité des logiciels peuvent requérir l'intervention des utilisateurs finaux ou s'appliquer aux caractéristiques du logiciel. Au chapitre 3, un ensemble de directives pour l'inspection experte des outils de FVD a été proposé. Cette méthode s'applique aux caractéristiques des outils de FVD. Il s'agit d'une adaptation de différentes normes et recommandations après une étude approfondie de la situation de travail en FVD. Dans ce chapitre, l'objectif est de développer une méthode qui puisse permettre une évaluation des outils de FVD par des utilisateurs finaux. En effet, les normes ISO 9241, ISO/IEC 9126, ISO 13407 et les critères de [Nielsen et Phillipps, 1993b], [Bastien et Scapin, 1993], [Smith et Mosier, 1986] et les recommandations générales telles que [Vanderdonckt 1994], [Senach 1990] proposent des attributs qui caractérisent la qualité d'utilisation des logiciels. Ces attributs comprennent : la facilité d'apprentissage, la satisfaction des utilisateurs, la compréhensibilité, l'efficacité, l'opérabilité, l'attractivité etc… Pour une analyse des logiciels, il existe aussi des attributs d'utilité des logiciels pouvant être couplés aux attributs qui caractérisent la qualité d'utilisation. Du point de vue des utilisateurs finaux, comment accéder et juger de la pertinence de ces différents attributs dans un environnement de FVD de façon concrète et valide ? Il est à noter que retrouver et corriger les erreurs du logiciel après livraison est 80 fois plus coûteux qu'en phase de conception [Boehm et Basili, 2001] et que du point de vue de l'utilisateur, l'interface est l'élément le plus important du logiciel puisque il s'agit de l'outil de médiation pour l'accomplissement de sa tâche [Costabile, 2001].

Nous essayons d'apporter des éléments de réponse à cette question dans ce chapitre. L'idée ici est de développer une méthode d'analyse qualitative qui puisse être étendue à l'évaluation et aider à la prise en considération des facteurs humains très tôt dans la phase de conception du logiciel. En effet, la philosophie de conception centrée sur l'utilisateur a été proposée par [Gould et Lewis, 1985] et s'appuie sur les principes suivants :

- attention immédiate et continue aux utilisateurs,
- conception intégrée (tout évolue en même temps : interface, manuel d'utilisateur, etc.),
- évaluation immédiate et continue auprès des utilisateurs,
- conception itérative.

L'approche proposée s'insère dans le cadre du respect du principe 3 ci-dessus. En effet, l'évaluation immédiate des logiciels en cours de conception par des utilisateurs nécessite d'avoir une méthode ou une technique adéquate. Une extension de la méthode d'analyse que nous présenterons servira comme nous le verrons à l'évaluation des outils de FVD par des utilisateurs ce qui contribuera à la définition de recommandations qui prises en compte dans les phases en amont du processus de conception permettra de corriger à moindre frais les erreurs susceptibles de se produire après développement des logiciels et qu'il serait beaucoup plus coûteux de corriger en fin de conception des systèmes à

venir. L'idée ici est donc de pouvoir vérifier la qualité des logiciels de FVD en requérant surtout la participation des utilisateurs finaux à l'évaluation.

Comme nous l'avons souligné, la méthode d'analyse d'outils de FVD que nous proposons s'appuie sur de nombreux critères et guides de style expérimentalement validés mais qui ne sont pas adaptés aux spécificités de la FVD. Ces critères et recommandations permettront de développer une série de questions pour l'évaluation. Pour un besoin de canalisation du processus d'adaptation de critères et de recommandations, nous avons choisi de travailler selon un cadre formel. La prochaine section de cette introduction présente le cadre formel de nos travaux.

4.1.1 Cadre formel : modèle GQM (Goal/Question/Metrics)

Pour la spécification des métriques de qualité, notre cadre formel est le modèle GQM en anglais « Goal/Question/Metrics » de [Basili et al, 1994]. Nous aurions pu choisir un cadre formel différent de GQM qui puisse nous permettre de mieux étayer nos objectifs en études qualitative des logiciels de FVD. En effet, le modèle GQM constitue un support permettant de mieux clarifier les objectifs à atteindre, les différents attributs de ces objectifs sont identifiés par l'intermédiaire de nombreuses questions. Comme l'indique la figure 4.1, de façon descendante, les mesures de qualité sont définies. Durant l'évaluation qualitative, l'explication des mesures ou l'interprétation des résultats obtenus est faite de manière ascendante comme l'indique la figure 4.1.

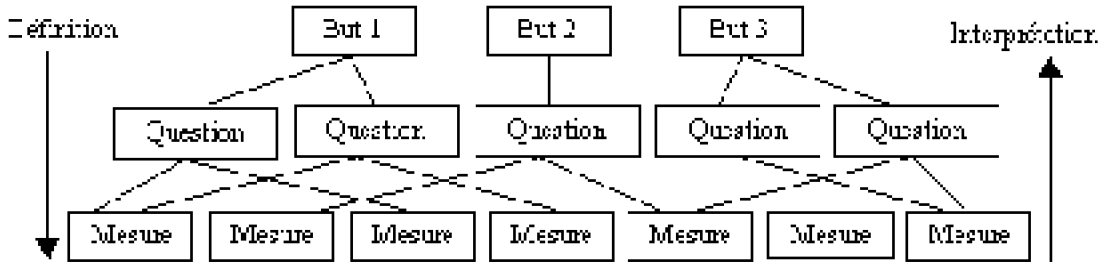


Figure 4.1 Modèle GQM [Basili et al, 1994]

Plus précisément, si on se situe dans le modèle GQM, notre objectif sera de trouver le moyen de spécifier chaque critère de qualité décrit par les différents normes et standards (tableau 4.1) afin de les appliquer aux spécificités de la fouille visuelle ou des critères relevés lors de la modélisation des utilisateurs ou des tâches, répertoriés aussi dans le tableau 4.1. Par exemple afin de mettre en oeuvre le guidage (but), la question sera comment guider nos utilisateurs finaux (question) et une des réponses pourra être la recommandation de l'algorithme de fouille de données à exécuter que nous avons vu dans le chapitre 3. Le présent chapitre utilise donc les connaissances décrites dans le chapitre 3 comme nous le verrons dans le processus de spécification de métriques de qualité de la prochaine section.

4.1.2 Processus de spécification des métriques

Comme l'indique la figure 4.2, le processus de spécification des métriques de qualité en FVD fait suite à l'analyse de la situation de travail de FVD (analyse de la tâche et des utilisateurs). Cette analyse permet d'accéder aux objectifs de qualité spécifiques au domaine traité. A partir de cette étape, il devient possible d'appliquer le cadre formel de GQM, sachant que les objectifs de qualité permettent de dériver les questions et les réponses à ces questions correspondent aux mesures de qualité.

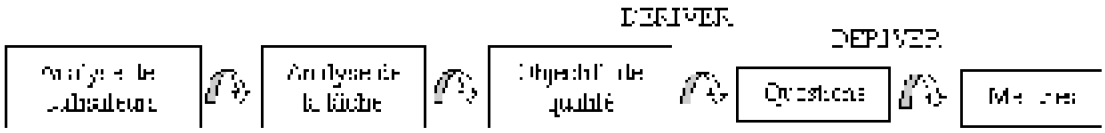


Figure 4.2 Processus de définition de mesures

Dans la suite de ce chapitre, partant des objectifs de qualité de logiciels, nous explicitons la méthode d'analyse des logiciels de FVD proposée, ensuite nous présentons un questionnaire d'évaluation des dites méthodes dérivé de la méthode d'analyse. Une étude de cas nous permet d'appliquer concrètement les critères développés et enfin nous concluons sur les perspectives de ce travail.

4.2 Définition de mesures : application du cadre formel GQM

4.2.1 Objectifs de qualité

Il importe de revenir sur le fait que les normes, les standards et les travaux isolés proposent des recommandations de qualité pour les logiciels en général. Pour les besoins de nos travaux, nous utilisons comme support ces recommandations qui constituent le G (buts) du modèle GQM. Les objectifs de qualité développés dans ces recommandations peuvent se résumer en :

- **le guidage** qui concerne les moyens mis en oeuvre pour conseiller, orienter, informer et conduire l'utilisateur,
- **la charge de travail** qui fait référence aux éléments de l'interface ayant un rôle dans la réduction de la charge perceptive ou mnésique des utilisateurs de même que dans l'augmentation de l'efficacité du dialogue,
- **le contrôle explicite** qui concerne la prise en compte par le système des actions explicites des utilisateurs et le contrôle qu'ont les utilisateurs sur le traitement de leurs actions,
- **l'adaptabilité** qui illustre la capacité à réagir selon le contexte et selon les besoins et les préférences des utilisateurs,
- **la gestion des erreurs** qui concerne les moyens permettant d'une part d'éviter ou de réduire les erreurs, d'autre part de les corriger lorsqu'elles surviennent,
- **la compatibilité** qui se réfère à l'accord entre les caractéristiques des utilisateurs et

des tâches, d'une part, et l'organisation des sorties, des entrées et du dialogue d'une application donnée, d'autre part,

- **les fonctionnalités du logiciel** qui concernent la puissance de l'outil par rapport aux capacités offertes.
- A un plus haut niveau d'abstraction, les objectifs de qualité décrits ci-dessus sont :
- l'efficacité qui fait référence à la capacité d'un dispositif à atteindre un objectif donné et se mesure par la réussite des tâches et la qualité de la performance,
- l'efficience, la capacité de réaliser une tâche donnée avec un minimum d'effort et qui se mesure par le taux et la nature des erreurs d'utilisation, le temps d'exécution d'une tâche donnée, le nombre d'opérations nécessaires à l'accomplissement d'une tâche, la charge de travail,
- la satisfaction qui concerne le niveau de confort des utilisateurs,
- l'apprenabilité (facilité avec laquelle un utilisateur apprend a se servir du système [Tricot et al., 2003]) et la maintenabilité qui peuvent être mesurées par le niveau d'expertise, l'amélioration et la stabilité de la performance des utilisateurs avec le temps etc.
- Par application du cadre formel de GQM, à partir de ces différents objectifs de qualité, un processus de réflexion et de questionnement se met en place et aboutit à la définition de critères de qualité spécifiques au domaine de la FVD comme décrit dans la prochaine section.

4.2.2 Questions à répondre pour améliorer la qualité des outils de FVD : Q de GQM

Dans le modèle GQM, la seconde étape après avoir défini les buts de notre modèle GQM est la définition d'un ensemble de questions dont les réponses contribueront à l'accomplissement des objectifs de départ (tableau 4.1).

Tableau 4.1 Questions relatives à l'amélioration de l'utilisabilité

	Questions
Guidage	Comment conseiller, orienter, informer et conduire les utilisateurs de la fouille visuelle de données ?
Charge de travail	Comment réduire la charge perceptive ou mnésique des utilisateurs et augmenter l'efficacité du dialogue ?
Contrôle explicite	Quelles sont les actions explicites des utilisateurs en fouille visuelle de données ? Comment rendre l'utilisateur maître de ce contrôle ?
Adaptabilité	Quels sont les contextes possibles ? Quels sont les utilisateurs finaux ? Quels pourraient être leurs préférences ?
Gestion des erreurs	Quelles sont les erreurs susceptibles de se produire ? Comment les éviter ?
Compatibilité	Comment prendre en compte le type des utilisateurs dans l'accomplissement de leurs tâches ?
Fonctionnalités du logiciel	Quelle est la puissance de l'outil par rapport aux capacités offertes ?

Afin de présenter les éléments de réponse aux différentes questions du tableau 4.1, nous avons choisi d'utiliser deux niveaux d'abstraction. Le niveau 1 présente les différentes mesures de qualité à un niveau d'abstraction élevé tandis que le niveau 2 est plus détaillé.

4.2.3 Ensemble de mesures : M de GQM (niveau 1)

L'analyse de la tâche en FVD permet d'apporter des éléments de réponse aux questions du tableau 4.1. Par exemple, pour la réduction de la charge de travail de l'utilisateur, il serait judicieux de prévoir une certaine flexibilité : donner la possibilité de choisir parmi plusieursméthodes de visualisation de données. Plusieurs méthodes d'analyse de données pouvant être disponibles simultanément dans un environnement de fouille, pour le guidage des utilisateurs, il faudrait développer des moyens pour les conseiller, orienter et guider. La recommandation de la méthode d'analyse de données à choisir pour un problème donné constitue une mise en œuvre possible du guidage.

Nous avons aussi intégré l'aspect fonctionnel dans nos critères d'évaluation. On va donc s'intéresser par la suite à la qualité technique des environnements. Les critères définis constituent des exigences qui pourront être prises en considération dès la phase de conception des logiciels.

Tableau 4.2 Mesures d'utilisabilité

Mesures	
Guidage	Recommandation de méthodes d'analyse de données Recommandation de méthodes de visualisation de données Feedback Aide en ligne Menus contextuels Gestion du profil utilisateur
Charge de travail	Réutilisation des données d'apprentissage Multiplicité du rendu Traitement des données de grande dimension Prise en main de l'outil
Contrôle explicite	Recommandation de méthodes d'analyse de données Recommandation de méthodes de visualisation de données Multiplicité du rendu
Adaptabilité	Personnalisation de l'interface utilisateur Prise en compte des habitudes de travail de l'utilisateur et ses préférences Adaptation à la variation des ressources interactionnelles
Gestion des erreurs	Détection et gestion de toutes les actions possibles sur l'interface, reprise sur erreur Possibilité d'exécuter une autre méthode d'analyse de données sans sans provoquer des erreurs du système
Compatibilité	Multiplicité du rendu Personnalisation de l'interface utilisateur Prise en compte des habitudes de travail de l'utilisateur et ses préférences
Fonctionnalités	Portabilité Accès aux données hétérogènes Diversification d'algorithmes Validation de modèles Présentation des résultats Réutilisation des données d'apprentissage

A présent, nous allons procéder à une description plus détaillée des critères du tableau 4.2 dans la prochaine section dédiée aux critères de niveau 2 suivant la hiérarchie définie précédemment.

4.2.4 Ensemble de mesures : M de GQM (niveau 2)

Le tableau 4.3 contient des données descriptives du logiciel et de son évaluateur. Il s'agit notamment du profil de l'utilisateur, du nom de logiciel et l'environnement sur lequel le logiciel a été évalué.

Tableau 4.3 Mesures qualitatives

Profil Utilisateur	
Nom du logiciel	
Environnement d'évaluation	

Il importe de souligner que [Hû et al., 2001] présentent des travaux qui comme les nôtres constituent une adaptation des recommandations générales et de critères disponibles pour le domaine de l'évaluation de produits interactifs pour l'apprentissage

humain. Dans le domaine des environnements virtuels, nous pouvons citer les travaux de [Bach, 2004] qui proposent une adaptation des recommandations générales d'ergonomie pour les interfaces graphiques du domaine des environnements virtuels.

L'organisation de critère choisie pour nos travaux est similaire à celle de [Hu et al., 2001]. La structure utilisée est hiérarchique et arborescente. Nous avons des thèmes, des méta critères et des critères.

En effet, les mesures de qualité présentées dans le tableau 4.2, nous ont permis de définir six thèmes d'analyse de logiciels de FVD. Chacun de ces thèmes est donc représenté sous forme d'un arbre avec des méta critères et des critères (feuilles de l'arbre). A partir de ces différents critères, nous avons dérivé comme nous le verrons un questionnaire d'évaluation. Dans chaque structure arborescente présentée dans les sections 4.2.4.1, 4.2.4.2, 4.2.4.3, 4.2.4.4, 4.2.4.5, 4.2.4.6, les critères soulignés sont spécifiques au domaine de la FVD, les autres critères peuvent servir à l'évaluation de tout type de logiciel.

4.2.4.1 Thème Utilisateur

Ce thème se réfère à l'utilisateur final etpermet d'avoir une perception d'ensemble de l'usage de l'outil de fouille graphique de données, de ses caractéristiques techniques, telles que l'adaptabilité et l'adéquation du système, de sa facilité de communication et de contrôle, de sa robustesse et de son efficacité, de sa facilité de compréhension, ainsi que de son caractère convivial et personnalisé.

Il s'agit plus précisément de répondre à la question : par rapport au profil de l'utilisateur, quel est l'effort qu'il déploie et quelles sont ses impressions générales en ce qui concerne l'utilité, l'efficacité, le pragmatisme, la flexibilité, la clarté, la commodité des différentes actions qu'il a accompli.

La flexibilité permet de prendre en compte les spécificités de chaque utilisateur en proposant plusieurs choix pour un traitement de données. Cette propriété permet aussi de gérer les combinaisons possibles d'actions sur l'interface par exemple l'utilisation du langage naturel et des clics souris.

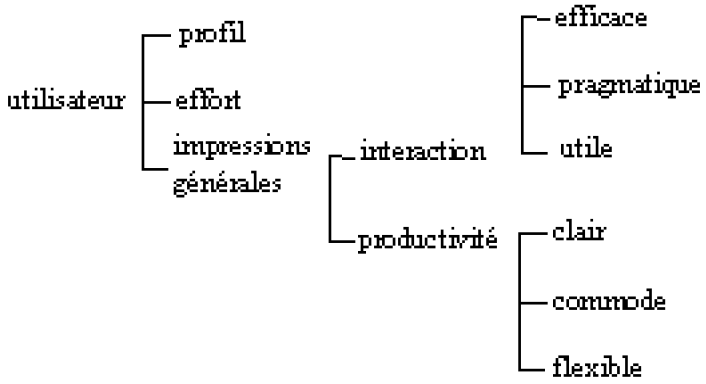


Figure 4.3 Représentation arborescente du thème utilisateur

Les impressions générales des utilisateurs se mesurent par l'efficacité, le pragmatisme, l'utilité de l'interaction et aussi par la clarté, la commodité et la fiabilité de la productivité.

Tous les critères du thème Utilisateur peuvent être appliqués à n'importe quel logiciel, ils ne sont pas spécifiques au domaine de la fouille visuelle de données.

4.2.4.2 Thème Utilisabilité

Le thème Utilisabilité est uniquement basé sur des recommandations générales pour la conception ergonomique de l'IHM qu'on peut retrouver en partie dans les références suivantes : [Nielsen, 1994], [Vanderdonckt, 1994], [Vanderdonckt, 1998], [Scapin et Bastien, 1997]. Parmi les méta critères de ce thème, nous avons répertorié la manipulation, le guidage (moyens mis en œuvre pour conseiller, orienter les utilisateurs) et la navigation.

L'idée ici est de pouvoir réduire les obstacles qui pourraient entraver la tâche des utilisateurs. On vérifiera si le système fournit des réponses dans de brefs délais aux utilisateurs suite à des actions accomplies (feedback immédiat). L'accent sera aussi mis sur la localisation des outils nécessaires à l'accomplissement de leurs tâches par les utilisateurs et à l'adaptabilité de la tâche aux différentes actions.

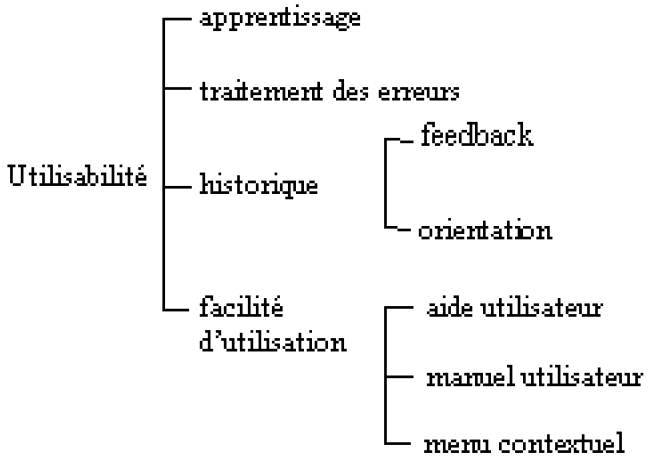


Figure 4.4 Représentation arborescente du thème Utilisabilité

L'Utilisabilité se mesure par la facilité d'apprentissage et de traitement des erreurs, le feedback, l'orientation, l'étude utilisateur, le manuel utilisateur et les menus contextuels. Tout comme pour le thème Utilisateur, les critères du thème Utilisabilité peuvent être utilisés pour l'évaluation de tout type de logiciel.

4.2.4.3 Thème Modèle de Présentation de l'Interface (MPI)

Le thème MPI **permet d'estimer de façon globale l'esthétique et l'aspect attrayant de l'outil et fait référence à l'ergonomie de l'interface. Ce thème complète les aspects relevés dans le thème Utilisabilité dans les besoins de qualité d'utilisation. Les**

critères qui y sont définis peuvent être appliqués à n'importe quel type de logiciel. Un bon usage de l'outil informatique est conditionné par une interface de bonne qualité. Ce thème d'évaluation nous permet d'acquérir le point de vue de l'utilisateur en ce qui concerne la convivialité de l'interface. Les éléments permettant d'évaluer le modèle de présentation de l'interface sont :

- la lisibilité (simplicité, clarté des représentations (textes, images), surcharge, choix typographique (taille, choix de polices)),
- les couleurs (adéquation du choix, nombre, pertinence de l'utilisation des couleurs, lisibilité).

Afin d'améliorer la lisibilité des interfaces produites, les capacités cognitives et perceptives des utilisateurs doivent être prises en compte.

Le MPI se mesure par la typographie, l'utilisation des couleurs, le graphisme, les icônes et par la lisibilité.

Dans leurs travaux, [Hû et al., 2001] disposent d'un thème d'évaluation (éléments de l'IHM) qui est proche du thème MPI, ce thème intègre des aspects multimédia avec le son et la vidéo qui ne sont pas indispensables à un environnement de FVD.

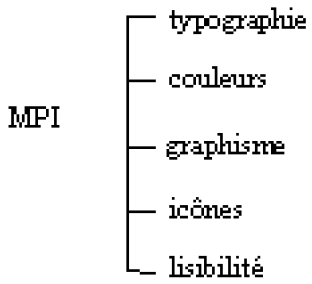


Figure 4.5 Représentation arborescente du thème MPI

4.2.4.4 Thème Qualité Technique

La Qualité Technique permet de mesurer la puissance de l'outil eu égard aux capacités offertes. L'évaluation à ce niveau se réfère aux possibilités d'adaptation à la tâche, à la précision de mise en oeuvre et à la capacité de prédiction de connaissances. Les mesures basées sur la technologie testent aussi le degré avec lequel un système peut manipuler des données de types et de tailles variables. Du point de vue applicatif, ceci peut être examiné sur une série de données de tailles croissantes.

Plus explicitement, la Qualité Technique (figure 4.6) se mesure par l'installation, l'assistance, la portabilité, l'architecture logicielle, l'accès aux formats hétérogènes des données, la flexibilité (diversité d'algorithmes), la validation des modèles de données, la présentation des modèles des données, l'exportation des modèles de données, l'interopérabilité, l'efficience, la robustesse, la réutilisation des données d'apprentissage, le traitement des données multidimensionnelles, la nouveauté des connaissances, la précision des connaissances et la richesse des connaissances. Ces critères sont pour la

plupart spécifiques au domaine de la FVD. Dans les travaux de [Hû et al., 2001], le thème qualité technique nécessaire à l'évaluation des produits interactifs pour l'apprentissage humain fait référence par exemple au temps de chargement d'une image, au temps et à la procédure de téléchargement d'une application. Ces facteurs sont spécifiques à leur domaine d'étude (l'apprentissage assisté par ordinateur).

Pour nous, la qualité technique fera beaucoup plus référence à des aspects spécifiques au domaine de la fouille visuelle de données comme l'indique la figure 4.6, à savoir :

- la portabilité : elle fait référence à la possibilité d'utilisation de l'outil sous plusieurs systèmes d'exploitation,
- l'accès aux données hétérogènes : il s'agit ici de voir si tout format de données est accepté par l'outil,
- la diversification d'algorithmes : elle fait référence aux nombres de méthodes d'analyse de données qui ont été implémentées dans l'outil,
- la validation de modèles : elle concerne la présence ou non de méthodes permettant de valider les résultats issus de la FVD,
- la présentation des résultats : il est question ici de juger de la pertinence de la représentation des résultats de la FVD,
- la réutilisation des données d'apprentissage : après une session de construction de modèle de données interrompue, est ce qu'il est possible de recommencer le processus au point ou l'utilisateur s'est arrêté ?

Par rapport aux travaux de [Marghescu et al., 2004], le thème Qualité Technique englobe les aspects que les auteurs ont regroupé dans leur thème qualité des connaissances. Cependant, ils ne s'intéressent pas à l'accès aux formats de données hétérogènes, à la validation des modèles des données, à la réutilisation des données d'apprentissage et au traitement des données multidimensionnelles.

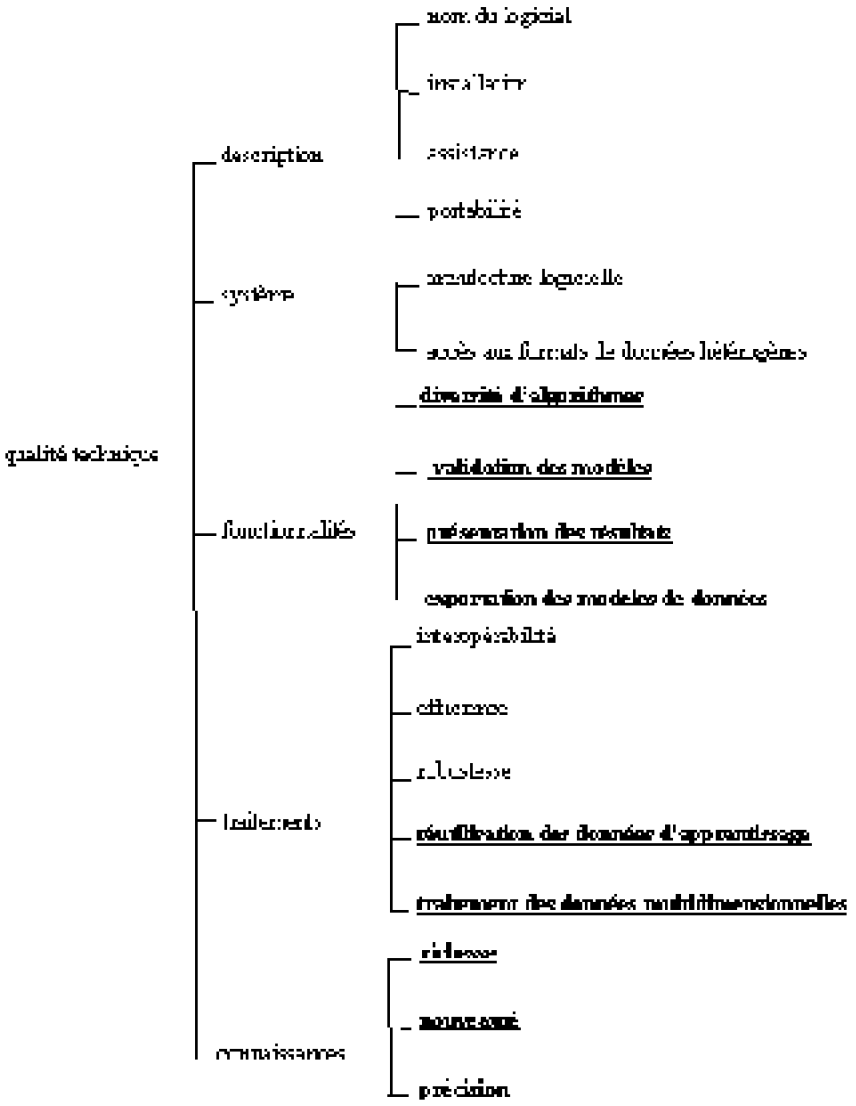


Figure 4.6 Représentation arborescente du thème Qualité technique

Du point de vue évaluation, on va s'intéresser dans un premier temps à l'installation du logiciel, l'environnement de travail, la facilité d'installation et l'assistance aux utilisateurs.

Le second méta critère permet de traiter de l'aspect portabilité. Ensuite, on s'intéresse à l'architecture logicielle sachant que le logiciel doit pouvoir s'exécuter sur diverses plates formes.

Beaucoup plus spécifique à la fouille visuelle de données, le méta critère fonctionnalité permet de mesurer l'aisance des utilisateurs face aux différents choix à opérer sur l'environnement par exemple.

4.2.4.5 Thème Scénario

Le thème Scénariofait référence au comportement dynamique de l'interaction durant le processus de fouille visuelle de données. Il s'agit ici de s'assurer d'une certaine

cohérence dans l'enchaînement des panoramas de visualisation et de s'assurer également que l'interface est intuitive et prédictive aux yeux de l'utilisateur.

En ce qui concerne ce thème, on va dans un premier temps s'intéresser au méta critère fouille visuelle de données. Il s'agira plus précisément de vérifier que des moyens sont bien déployés pour orienter, informer et conseiller les utilisateurs durant la session de fouille. Il s'agit aussi de vérifier que ces moyens sont efficaces et que les utilisateurs ne procèdent pas à plusieurs essais avant de parvenir aux modèles finaux des données.

Le méta critère tâche entière va s'intéresser aux mesures concourrant à la bonne marche de la fouille visuelle des données. Il s'agira par exemple des moyens développés pour prévenir et corriger les erreurs, qui consistent par exemple à des actions incorrectes.

Le thème Scénario fait référence à la qualité de l'interaction dans le logiciel et ses critères peuvent être appliqués à n'importe quel type d'environnement.

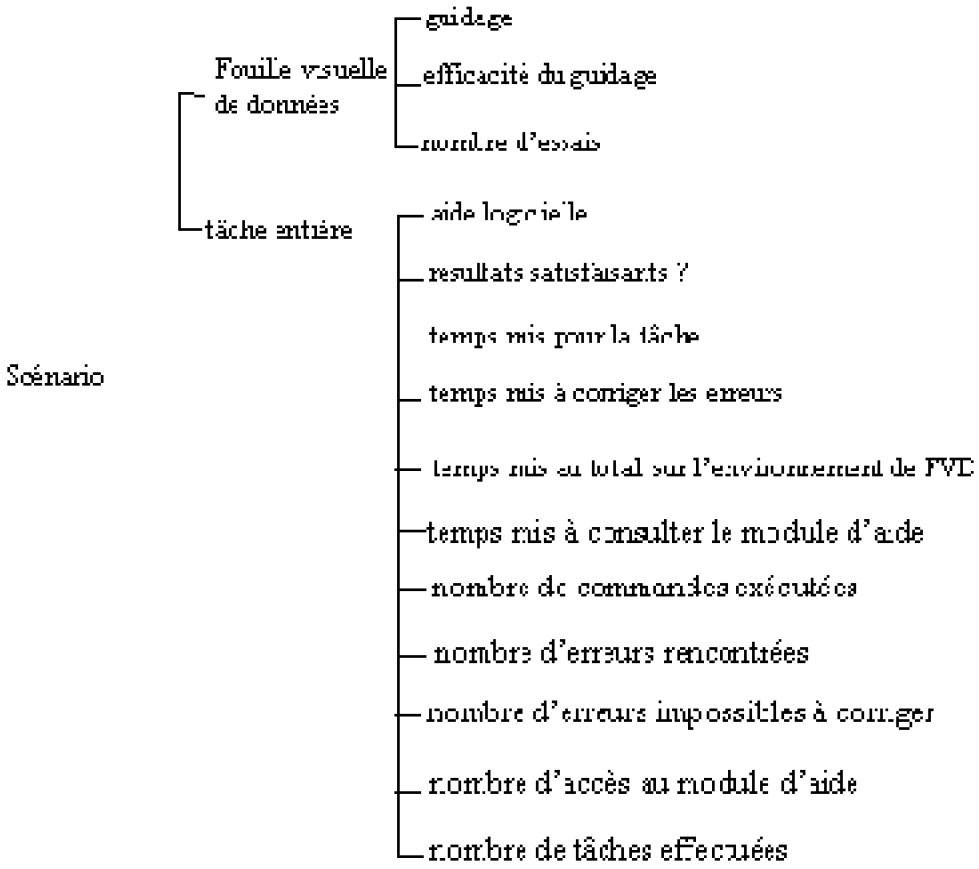


Figure 4.7 Représentation arborescente du thème Scénario

Dans le méta critère tâche entière, on retrouve les critères répertoriés sur le site de l'AS Evaluation [AS Evaluation, 2005] pour l'évaluation des techniques de visualisation. Nous devons insister sur le fait que tous les critères répertoriés sur le site de l'AS Evaluation ne constituent qu'une partie du thème scénario de la méthode que nous proposons. Par rapport à l'AS Evaluation, la méthode proposée est beaucoup plus exhaustive.

4.2.4.6 Thème Visualisation

Le thème Visualisation concerne la pertinence des représentations graphiques et leur structuration par rapport aux objectifs de fouille de l'utilisateur et de son profil.

La visualisation permet de représenter des données sur lesquelles des manipulations (interactions) sont opérées afin d'aboutir au modèle des données. Il s'agit ici de voir dans quelles mesures les représentations graphiques utilisées facilitent la perception et la compréhension des connaissances. L'idée aussi est de savoir si la primitive graphique permet une détection rapide de connaissances, si les couleurs sont utilisées correctement, si les entités graphiques sont détectées sans confusion. Un autre aspect évalué dans ce thème est la charge cognitive relative à l'exécution de la tâche de l'utilisateur. Un moyen de réduction de la charge cognitive des utilisateurs serait la réduction de la densité des informations.

Le critère compatibilité et cohérence permet de vérifier si les chemins valides et minimaux d'interactions sont ceux attendus par l'utilisateur, s'il existe des points de repère pour les utilisateurs en général.

Le thème visualisation regroupe trois types de critères. Des critères propres à tout type de logiciels notamment la rapidité des méthodes graphiques, leur compatibilité et cohérence et l'utilisation de couleurs. Ce thème comprend aussi des critères propres à toute visualisation interactive : vue d'ensemble, zoom, détails à la demande, filtrage, comparaison, historique et extraction. Enfin, nous avons des critères propres aux méthodes de FVD. Comme exemples de ces critères, nous pouvons citer la détection de données manquantes, la multiplicité des représentations graphiques disponibles, la détection des connaissances, le traitement des densités d'informations et de données de très grande dimension.

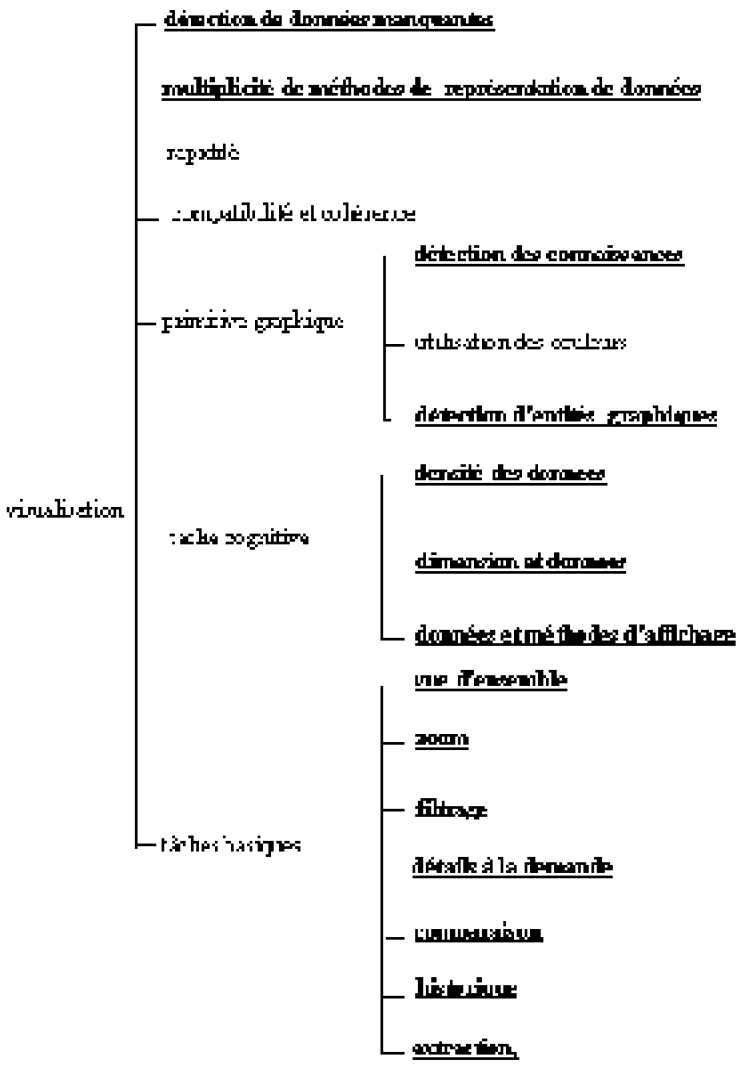


Figure 4.8 Représentation arborescente du thème visualisation

Plusieurs auteurs [Card et al., 1999], [Chen et Czerwinski, 2000] soulignent la nécessité d'une évaluation empirique des techniques de visualisation des données. Selon [Dix et al., 1998], cette évaluation peut être opérée afin de s'assurer que le système est conforme aux besoins des utilisateurs et aux spécifications des développeurs. [Tufte, 1993] et [Bertin, 1981] ont défini des indices de qualité de base en visualisation qui sont intégrés au thème visualisation dont l'arborescence est décrite dans la figure 4.8. Cette arborescence incorpore aussi les tâches de visualisation de [Schneiderman, 1996] et utilise les résultats de [Morses et al., 2000], [Stasko et al., 2000], [Fangseu Badjio, 2005a], [Fangseu Badjio et Poulet, 2005c], [Fangseu Badjio et Poulet, 2005d] et [Fangseu Badjio et Poulet, 2005e].

Dans les travaux de [Marghescu et al., 2004], un des trois thèmes d'évaluation de la qualité d'utilisation en FVD concerne la visualisation et regroupe 4 critères : le paramétrage initial, la disposition graphique des données, les tâches d'exploration et les fonctions de transfert des résultats du processus de fouille.

4.2.5 Discussion : comparaison des métriques d'analyse des logiciels de FVD avec les autres méthodes d'analyse

La méthode d'analyse des outils de FVD ainsi définie comporte 69 métriques et 6 thèmes principaux. Les thèmes se subdivisent en méta critères et critères, avec un maximum de trois niveaux dans cette hiérarchie avec le thème *Utilisateur*, méta critères *Impressions Générales* et critères (*Interaction*, P*roductivité*).

Au niveau 2 (méta critères), nous en avons 12 au total et 13 mesures sont de niveau 1 (critères).

Ces différentes métriques ont les avantages suivants :

- elles peuvent être utilisées très tôt dans le cycle de conception de logiciels de FVD afin d'éviter d'éventuelles problèmes qualitatifs,
- elles s'adressent aux experts (analyse de données, concepteurs d'interfaces, spécialistes du domaine des données, etc.) et peuvent servir à l'évaluation utilisateur,
- elles intègrent les notions d'ergonomie, d'analyse de données, de génie logiciel, d'IHM, etc.
- elles sont basées sur des standards et des méthodes assez éprouvées telles que les heuristiques [Nielsen et Mollich, 1990], les recommandations ergonomiques de [Bastien et Scapin, 1993], les différentes normes ISO ou des travaux tels que : [Smith et Mosier, 1986], [Vanderdonckt 1994], [Senach 1990], [Hû et al., 2001], [AS Evaluation, 2005], [Tufte, 1993], [Bertin, 1981], [Card et al., 1999] et [Chen et Czerwinski, 2000].

Dans le cadre d'une analyse ou d'une évaluation, le risque avec cette méthode assez exhaustive est de se perdre dans la recherche des facteurs de non qualité. Il s'avère donc nécessaire que l'évaluateur se fixe des objectifs bien précis et qu'il délimite clairement les thèmes à étudier au cours d'une évaluation donnée.

Dans la plupart de méthodes traditionnelles, l'analyse et l'évaluation de la qualité des logiciels se réfèrent seulement aux éléments de l'interface utilisateur. Un environnement de FVD dispose d'une interface avec les utilisateurs mais aussi d'une interface avec les données et les méthodes de visualisation de données. Pour une analyse adéquate de la qualité des outils de FVD, il s'impose le besoin de traiter de différents aspects suivants : des aspects orientés visualisation, des aspects orientés données, des aspects orientés utilisabilité et des aspects orientées contexte (activité, événement, régulation). Hormis les aspects orientés contexte qui sont implicitement pris en compte dans les différents thèmes d'analyse que nous avons présentés, tous les autres thèmes sont traités.

Notre objectif à présent est de définir un questionnaire à partir des métriques de qualité relevées dans les sections 4.2.4.1, 4.2.4.2, 4.2.4.3, 4.2.4.4, 4.2.4.5, 4.2.4.6, afin de pouvoir identifier les différents problèmes des utilisateurs.

4.3 Diagnostic des systèmes de FVD : questionnaire destiné aux

utilisateurs

Pour le diagnostic des systèmes de FVD, nous avons choisi de recueillir les impressions des utilisateurs réels soumis à des tâches réelles de FVD.

La fiabilité de la méthode d'évaluation ainsi proposée est théorique car elle utilise entre autres les modèles pour représenter les utilisateurs et les tâches. La modélisation de l'utilisateur abordée au chapitre 3 a permis de détecter les problèmes théoriques que peuvent rencontrer les utilisateurs. Il importe d'avoir confirmation au moment du diagnostic.

Cette section présente la grille d'évaluation des outils de FVD issue des métriques d'évaluation présentées ci-dessus pour le diagnostic de ces outils. L'idée ici pour le diagnostic est d'interviewer les utilisateurs potentiels de ces outils (novices, occasionnels, experts, …) ayant été soumis à des tâches de fouille interactive. Ce questionnaire est donc rempli à l'issue d'une interaction utilisateur – environnement de fouille.

Le barème choisi pour l'évaluation utilisateur avec le questionnaire est un nombre compris entre 1 et 5 qui est interprété comme suit :

- 5 si l'outil évalué répond très bien au critère,
- 4 si l'outil évalué répond bien au critère,
- 3 si l'outil évalué répond moyennement au critère,
- 2 si l'outil évalué répond peu au critère,
- 1 si l'outil évalué ne répond pas du tout au critère.

Il est à noter que le questionnaire permet une évaluation de l'ergonomie du logiciel au travers des thèmes Utilisabilité et modèle de présentation de l'interface. La grille permet aussi d'accéder à l'acceptabilité du système avec le thème utilisateur. Les thèmes qualité techniques, scénario et visualisation traitent à la fois de la qualité d'utilisation et de l'utilité de ces systèmes.

4.3.1 Questions pour le thème Utilisateur

Les critères regroupés dans le thème utilisateur (tableau 4.4) sont subjectifs et permettent d'obtenir les impressions, le sentiment de l'évaluateur vis-à-vis de l'environnement.

Tableau 4.4 Thème utilisateur

	1	2	3	4	5
Adéquation du système à ses besoins					
Facilité de communication					
Facilité de compréhension					
Efficacité					
Pragmatisme					
Utilité					
Convivialité					
Personnalisation					
Flexibilité					

4.3.2 Questions pour le thème Utilisabilité

La facilité d'utilisation, les fonctions qui permettent de soutenir certaines tâches, les avertissements (orientation) et l'aide font partie du thème utilisabilité (tableau 4.5). Les critères de ce thème sont complétés par ceux du modèle de présentation de l'interface (MPI), tableau 4.6 qui s'intéresse aussi aux interfaces utilisateur en FVD. Dans le MPI, il s'agit de voir dans quelles mesures les informations présentées à l'écran sont repérables, mémorables, lisibles et contextuelles.

Tableau 4.5 Thème utilisabilité

	1	**2**	**3**	**4**	**5**
Apprentissage (prise en main)					
Traitement des erreurs					
Feedback					
Orientation					
Aide utilisateur					
Manuel utilisateur					
Menu contextuel					

4.3.3 Questions pour le thème MPI

Les questions relatives au MPI concernent uniquement l'interface utilisateur.

Tableau 4.6 Thème modèle de présentation de l'interface

	1	2	3	4	5
Typographie					
Choix et utilisation – couleurs					
Pertinence – couleurs					
Graphisme					
Icônes					
Lisibilité					
Disposition des éléments à l'écran					

4.3.4 Questions pour le thème Qualité technique

Les questions du thème qualité technique sont pour la plupart spécifiques au domaine de la FVD. Nous avons donc une forte préoccupation à la qualité du modèle des données ou des connaissances et la facilité des traitements.

Tableau 4.9 Thème qualité technique

	1	2	3	4	5
Installation					
Assistance					
Portabilité					
Architecture					
Accès aux données hétérogènes					
Diversification d'algorithmes					
Validation de modèles					
Présentation de résultats					
Exportation de modèles					
Nouveauté - connaissances					
Précision – connaissances					
Richesse - connaissances					
Interopérabilité					
Efficience					
Robustesse					
Réutilisation des données d'apprentissage					
Traitement de données de grande dimension					

4.3.5 Questions pour le thème Scénario

Le thème Scénario permet de revoir les exigences initiales des logiciels en terme de mesures temporelles (temps mis pour effectuer une tâche, temps mis à corriger les

erreurs, etc …) et de compréhension. La compréhension ici est définie par les critères tels que l'efficacité du guidage, l'aide logicielle, le nombre de commandes ou le nombre d'erreurs.

La qualité de l'interaction est ici explorée, sachant que l'interaction peut concerner l'analyse des données ou l'exploration des données.

Ce questionnaire permet d'améliorer les performances des utilisateurs lors de l'accomplissement de leurs tâches, de réduire leurs efforts et de prévenir les erreurs désastreuses.

Tableau 4.7 Thème Scénario

	1	2	3	4	5
Temps mis pour effectuer une tâche					
Temps mis à corriger les erreurs					
Temps mis sur l'environnement					
Temps mis à consulter le module d'aide					
Nombre de commandes exécutées					
Nombre d'erreurs rencontrées					
Nombre d'erreurs impossibles à corriger					
Nombre d'accès au module d'aide					
Nombre de tâches effectuées					
Efficacité du guidage					
Aide logicielle					
Résultats satisfaisants ?					

4.3.6 Questions pour le thème Visualisation

Le thème Visualisation permet de mesurer l'importance des capacités cognitives nécessaires à la création d'un modèle de données à partir de représentations graphiques : détection de données manquantes ou incorrectes, compréhensibilité, détection de connaissances, d'entités graphiques, compatibilité et cohérence des interactions. Dans ce thème, il est aussi possible d'évaluer la mise en œuvre des 7 tâches principales de visualisation décrites par [Schneiderman, 1996] : zoom, filtrage, détail à la demande, historique, extraction, vue d'ensemble, comparaison.

Les différents critères de ce thème permettent implicitement de choisir des

techniques de visualisation en fonction de leur faciliter de représenter des données en 2, 3, N dimensions.

Tableau 4.8 Thème visualisation

	1	2	3	4	5
Détections données manquantes ou incorrectes					
Compréhensibilité					
Rapidité					
Multiplicité du rendu					
Préférence utilisateur pour la multiplicité du rendu					
Détection des connaissances par les primitives graphiques					
Utilisation de couleurs par les primitives graphiques					
Détection d'entités graphiques par les primitives graphiques					
Compatibilité et cohérence des interactions					
Tâche cognitive – densité informationnelle					
Tâche cognitive – traitement des données de grande dimension					
Tâche cognitive – visualisation de données					
Tâche de base – zoom					
Tâche de base – détail à la demande					
Tâche de base – filtrage					
Tâche de base – extraction					
Tâche de base – historique					
Tâche de base – vue d'ensemble					
Tâche de base – comparaison					

4.4 Problèmes de qualité susceptibles d'être répertoriés

Au terme d'un diagnostic utilisateur de logiciels de FVD avec le questionnaire élaboré, on peut s'attendre à plusieurs types de problèmes de qualité. Du point de vue de l'interface, on peut avoir des problèmes de clarté, de lisibilité, de gestion d'erreurs, de désorientation, de cohérence, de prévisibilité et de guidage d'une étape à l'autre des traitements.

En ce qui concerne la qualité de l'interaction, il peut y avoir trop d'actions pour accomplir un but, un temps de réponse important avec un mauvais feedback, un problème de flexibilité et de désorientation.

Les représentations graphiques quant à elles peuvent manquer d'intuitivité, il peut y avoir des erreurs dans le développement de tâches principales de visualisation, les problèmes peuvent concerner le traitement de données de très grande dimension.

Du point de vue des connaissances acquises, elles peuvent ne pas respecter les attentes de richesse, de compréhensibilité, d'utilité et de nouveauté qui caractérisent les résultats des processus d'ECD.

Les processus d'analyse et d'évaluation décrits jusqu'à présent sont purement théoriques. Dans la section suivante, nous allons évaluer la méthode présentée dans le cadre du diagnostic utilisateur des outils de FVD.

4.5 Etude de cas : application du questionnaire d'évaluation

L'objectif de cette étude de cas est d'opérer le diagnostic des outils de FVD utilisant les matrices 2D comme canal de communication. Nous voulons mettre en exergue un ensemble de dysfonctionnements dans les dits environnements, ceci afin de créer des anticipations pour la réalisation des tâches, de conseiller, d'informer et d'orienter les utilisateurs des environnements de fouille visuelle de données de ce type. Pour les besoins de cette évaluation, nous avons établi deux principales hypothèses de travail.

Première hypothèse de travail : si l'ensemble des critères d'évaluation présenté dans la section précédente est pertinent, alors l'évaluation par des utilisateurs de même profil nous permettra de détecter des problèmes de conception et d'utilisation semblables.

Deuxième hypothèse de travail : si les environnements de FVD disponibles sont de bonne qualité, stables et cohérents, alors l'étude de cas ne permettra pas de relever des problèmes de qualité.

4.5.1 Liste des tâches

Pour les besoins de notre évaluation, la tâche prescrite est la construction interactive d'un arbre de décision à partir de représentations des ensembles de données du tableau 7, issus de la base UCI [Blake et Merz, 1998].

Tableau 4.9 Description des ensembles de données

Nom	Nombre d'observations	Nombre d'attributs	Nombre de classes
Australian	690	14	2
Ionosphere	351	32	2
Iris	150	5	3
Letter	5000	40	3
Mushrooms	8124	24	2
Nursery	12960	8	5
Segmentation	2310	11	7
Shuttle	20000	16	26
Vehicle	846	18	4
Yeast	1484	9	10

Les ensembles de données choisis sont de taille variable (nombre d'attributs et nombre de dimensions).

4.5.2 Cadre de l'étude

Cette étude porte sur l'utilisation de WEKA [Witten et Eibe, 2005] et de CIAD [Poulet, 2001], des outils de FVD qui permettent la construction interactive d'arbres de décision. Pour chacun de ces outils, les utilisateurs soumis à l'évaluation en vue du diagnostic ont des profils similaires. Il s'agit plus précisément de 13 utilisateurs dont onze hommes et deux femmes pour WEKA, ces utilisateurs ont suivi une formation préalable en FVD. En ce qui concerne CIAD, le test est effectué par quatre utilisateurs autonomes (spécialistes en fouille visuelle de données). Il s'agit de deux hommes et deux femmes.

Pour ce diagnostic de WEKA et CIAD, nous aurions aimé avoir un nombre beaucoup plus exhaustif d'évaluateurs. Notre hypothèse de travail au départ était d'évaluer ces outils avec des utilisateurs du même profil. Cependant, il s'est avéré impossible pour nous au moment du diagnostic de réunir plus de quatre spécialistes en fouille visuelle de données, raison pour laquelle l'évaluation de CIAD a été faite avec quatre utilisateurs. En ce qui concerne l'évaluation de WEKA, elle s'est faite avec 13 étudiants du niveau de dernière année d'école d'ingénieur qui ont préalablement participé à un cours d'ECD.

4.5.3 Résultats

Le tableau 4.10 présente les résultats de ces deux diagnostics. La première colonne présente les critères mis en exergue dans ces diagnostics et les deux dernières colonnes de ce tableau représentent les moyennes obtenues pour ces critères.

Les critères ainsi mis en évidence sont relatifs aux propriétés techniques et cognitives développés à travers WEKA et CIAD.

Il est à noter que notre première hypothèse de travail a été vérifiée dans les deux diagnostics.

Pour les évaluateurs, les outils sont très utiles (10ème ligne du tableau). Mais, le traitement des erreurs avec WEKA est inexistant. Au cas où l'utilisateur se trompe

pendant une étape de construction de l'arbre de décision, il doit recommencer dès l'étape initiale. Par contre, CIAD permet de corriger les erreurs et de réutiliser des données d'apprentissage, ce qui réduit la charge de travail de l'utilisateur. Mais, l'installation de CIAD n'est pas aisée, CIAD est encore à sa phase expérimentale. Un autre point faible de ces outils réside dans l'impossibilité de traiter les données de formats différents. En effet, un seul format de données est accepté.

La prise en main de CIAD et WEKA est facile (score de 80%). La disposition des éléments à l'écran, le modèle de présentation de l'interface, l'utilisation des graphiques sont assez satisfaisants pour les deux outils.

Les utilisateurs des méthodes CIAD et WEKA ne sont pas guidés (ligne 5 du tableau), il manque à ces outils des modules d'aide en ligne, des informations contextuelles ou un manuel utilisateur.

Cette étude de cas permet de suggérer aux concepteurs de WEKA de travailler l'aspect réutilisation de données d'apprentissage qui permettra de réduire la charge de travail des utilisateurs. Au cas ou ces derniers ne vont pas jusqu'au bout du traitement interactif d'un ensemble de données, ils n'auront pas à recommencer à l'étape initiale. En ce qui concerne CIAD et WEKA, ces résultats permettent d'attirer l'attention des concepteurs sur la nécessité de développer les modules d'assistance aux utilisateurs, des manuels utilisateurs et de proposer plusieurs alternatives possibles en ce qui concerne les méthodes de visualisation et d'analyse de données, les aspects cognitifs de la visualisation pour les utilisateurs et les préférences des utilisateurs. Après avoir proposé plusieurs alternatives aux utilisateurs, il s'avère aussi nécessaire de les guider dans le choix de celle qui convient le mieux à la résolution de leur problème.

Au terme de ce diagnostic de WEKA et CIAD, le point qui attire le plus notre attention concerne les difficultés relevées dans ses outils pour le traitement des données de grandes dimensions pourvues ou non de nombreux objets ou observations (lignes 2, 3 et 4 du tableau des résultats). Ceci constitue un réel problème car à l'heure actuelle, les ensembles de données sont de plus en plus grands. Pour améliorer l'état actuel des recherches dans ce domaine, suite au diagnostic des outils de FVD, nous proposons une contribution au traitement des données de ce type dans le chapitre 6.

4.6 Conclusion et travaux futurs

Faisant suite à la crise du logiciel en fin des années 1960, le génie logiciel a vu le jour. Les méthodes de cette discipline définissent le cycle de vie du logiciel. Mais chaque nouveau logiciel possède ses spécificités. Les recommandations de qualité existantes nécessitent donc d'être réadaptées en vue de s'appliquer à la FVD. En effet, comme tout système interactif, il importe de s'assurer que la part interactive du système de FVD fonctionne correctement et ne provoque pas des erreurs susceptibles de pouvoir porter atteinte au bon fonctionnement du système, mais aussi, il est important de s'assurer que le système de FVD est facile à utiliser, en d'autres termes que les structures d'interaction que le système propose permettent effectivement à l'usager d'effectuer les tâches qu'il entend pouvoir réaliser avec le système.

La FVD étant à la croisée de plusieurs disciplines (interfaces homme machine, psychologie cognitive, intelligence artificielle, etc …), il est assez difficile de définir une démarche qualitative. Toute la difficulté relative à une telle réalisation réside dans la diversité des connaissances nécessaires à l'analyse des outils de FVD. Afin de mieux comprendre les utilisateurs ainsi que les différentes tâches à réaliser en général, le domaine de recherche ergonomie du logiciel propose d'utiliser l'analyse de la situation de travail. Méthode que nous avons adoptée dans le cadre de nos travaux. Plus précisément, après une analyse de la situation de travail en FVD : une étude détaillée du processus de FVD mettant l'accent sur les conditions de fonctionnement, l'effet du contexte, des situations et des tâches, nous avons pu poser les jalons de nos réalisations. Concrètement, nous avons développé une méthode de diagnostic des environnements de FVD. Ensuite nous avons procédé au diagnostic de certains de ces environnements. Enfin, les conclusions relatives à ces diagnostics ont servi de base au développement d'outils d'aide aux utilisateurs des environnements de FVD, outils spécialisés dans ce domaine.

En perspectives aux travaux de diagnostic des outils de FVD que nous décrivons, nous envisageons une automatisation avec pondération des différents critères définis et une possibilité de support au choix du meilleur outil de FVD à utiliser parmi ceux disponibles et évalués par des utilisateurs finaux.

Cette étape de notre contribution vise à concevoir de nouveaux environnements de FVD ou à adapter ceux qui existent déjà afin qu'ils favorisent la découverte des connaissances non triviales et potentiellement utiles dans les données, permettent aux utilisateurs finaux de réaliser leurs tâches sans embûches, compensent les difficultés relatives à l'autonomie des utilisateurs d'un système de FVD, soient adaptés aux objectifs de FVD.

Après le diagnostic des environnements de FVD, le second point étudié concerne l'appropriation des outils de FVD. En effet, le diagnostic des systèmes existants de FVD nous a permis de constater par exemple que dans ces systèmes, tout utilisateur qui effectue une requête obtient le même résultat ou qu'il peut exister de nombreux choix à opérer dans des systèmes de ce type . Pourtant chaque utilisateur qui se sert des environnements de fouille de données a sa singularité ainsi que des besoins particuliers. Une solution à ce problème serait de développer un environnement de fouille de données tenant en compte les besoins spécifiques de chaque type d'utilisateur. Mais cette solution s'avère coûteuse car elle impose l'achat de nouveaux outils aux compagnies déjà pourvues d'outils de fouille de données. Dans les chapitres 5 et 6, nous étudions des approches permettant d'intégrer des supports aux utilisateurs des systèmes existants et nouveaux de fouille de données.

Partie 3 : Contributions à l'amélioration de la qualité des outils de FVD

Publications

E.Fangseu Badjio, F. Poulet, *Prétraitement de grands ensembles de données pour la fouille visuelle*, to appear in proc. of **EGC'06**, Lille, Jan 2006.

Fangseu Badjio E., Poulet F.: *User Guidance: From Theory to Practice, the Case of Visual Data Mining*, to appear in proc. of **IEEE-ICTAI'05**, the 17th IEEE International Conference on Tools with Artificial Intelligence, Hong Kong, China, Nov 2005.

Fangseu Badjio E. : *Retrouver les meilleurs algorithmes pour la classification supervisée des données*, **INFORSID'05**, Grenoble, 24 au 27 Mai 2005, pp.337-352, 2005.

Fangseu Badjio E., Poulet F.: *Dimension Reduction for Visual Data Mining*, in proc. of **ASMDA'05**, the International Symposium on Applied Stochastic Models and Data Analysis, J. Janssen and P. Lenca (Eds), Brest, France, pp.266- 275, 2005.

Fangseu Badjio E., Poulet F.: *Feature Selection: CBR Retrieval Improvement for Knowledge Management*, on CD proc. of **IMT Conference'04**, The International Management and Technology Conference, Orlando, Florida, 2004.

Fangseu Badjio E., Poulet F.: *A decision support system for data miners*, on CD proc. of **IEEE-AISTA'04**, The International Conference on Advances in Intelligent Systems - Theory and Applications, Luxembourg-Kirchberg, Luxembourg, 2004.

Fangseu Badjio E., Poulet F.: *Qualité de prédiction des performances des algorithmes de classification de données*, **SFC'04**, Meetings of the French-speaking Classification Society, Bordeaux, pp.189-192, 2004.

Fangseu Badjio E., Poulet F.: *Data Mining Algorithm Prediction*, in proc. of **IFIP-AIAI'04**, The Symposium on Professional Practice in AI, Toulouse, France, Aug. 2004, pp.383-392, 2004.

Fangseu Badjio E., Poulet F. : *Guidage des utilisateurs en fouille visuelle de données*, in proc. of **EGC'04** Workshop on Visualization and Knowledge Discovery, Clermont-Ferrand, pp.13-18, 2004.

Chapitre 5 : Support à la sélection du meilleur algorithme pour la FVD

5.1 Introduction

Nous présentons une nouvelle technique de support automatique au choix du meilleur algorithme de classification supervisée de données pour la FVD qui corrige des erreurs observées dans les techniques existantes, utilisées dans le domaine de la classification supervisée, notamment la propagation de l'erreur de prédiction comme nous le verrons. Ces travaux que nous décrivons (choix d'un algorithme automatique pour la FVD) concernent les deux premières variantes du modèle de Ankerst et visent à une amélioration de la qualité des outils de FVD. Il s'agit dans une certaine mesure d'assister l'activité de conception du modèle des données. Les études en qualité des logiciels de FVD présentées dans les chapitres 3 et 4 montrent qu'il est nécessaire d'orienter, de conseiller, d'informer les utilisateurs de ces environnements. Partant de ce constat purement théorique, nous allons procéder à sa mise en pratique dans des outils de FVD. Concrètement, nous allons concevoir, développer et présenter un système d'aide à la décision pour le processus de FVD. L'aboutissement de l'activité de FVD est la création d'un modèle des données à traiter. Il s'agit d'une activité de conception à part entière et par conséquence en plus des besoins en système d'aide à la décision pour une meilleure qualité des systèmes de FVD, on peut trouver des fondements de cette nécessité d'assistance aux utilisateurs dans d'autres disciplines, par exemple en conception.

Face à l'augmentation de la quantité de données disponible dans le monde, la visualisation a été adoptée en fouille de données comme méthode d'exploration, de confirmation d'hypothèses ou enfin de présentation de données/résultats. La visualisation a ensuite servi comme technique de support à la découverte de connaissances dans les données. Le principe étant le suivant : à partir d'un ensemble de données, une

représentation graphique est produite. L'utilisateur s'appuie sur cette représentation et sur ses capacités en reconnaissance de formes pour paramétrer l'algorithme de découverte de connaissances, concevoir le modèle des données et procéder à des estimations.

La FVD consiste donc en l'utilisation de la visualisation comme canal de communication pour la fouille de données. Pour assister les utilisateurs dans un tel processus, il est nécessaire de prendre en compte les informations relatives à leurs profils ou à leurs compétences, à la tâche et au contexte du travail. Dans la plupart de domaines nécessitant une activité de conception (l'art, la programmation en informatique, l'architecture, l'industrie, etc…), des stratégies ont été développées pour une assistance à la conception (brainstorming pour la créativité collective, travail par association d'idées, etc. [Hatcheut et al., 2005]). Dans l'industrie en général, l'activité de conception a été rationalisée par la mise au point du langage de projet qui permet la gestion de toutes les phases de conception d'un nouveau produit. La nécessité d'assistance aux utilisateurs des outils de FVD ne relève donc pas seulement de la qualité des logiciels, des interfaces ou de l'ergonomie du logiciel.

Avant d'introduire le mécanisme de support à la décision du « fouilleur de données » et afin d'étayer notre propos, nous allons nous intéresser aux analogies qui peuvent exister entre l'activité de conception en général et l'activité de FVD. L'activité de FVD rappelons le aboutit à la conception d'un modèle des données.

5.1.1 Analogies entre conception en général et conception du modèle des données

La conception selon l'AFNOR est une activité créatrice, qui partant de besoins exprimés et des connaissances existantes aboutit à la définition d'un produit satisfaisant ces besoins et industriellement réalisables. Dans un sens beaucoup plus général, selon le grand dictionnaire terminologique de la langue française, la conception est une façon de voir ou de comprendre. En fouille de données en général, le modèle des données est une façon de voir ou de comprendre les données. La conception est caractérisée par un état initial flou, le concepteur a du mal à appréhender les besoins réels des utilisateurs. En cet autre point, il existe une analogie avec la conception du modèle des données. Le point de départ de la FVD est flou, l'analyste des données ne sait pas exactement ce qu'il cherche dans l'ensemble volumineux des données. Puis, il découvre de nouvelles corrélations dans ces données. Il s'agit d'informations inconnues avant et potentiellement utiles. Des opérations de transformation servent de support à cet effet. A chaque opération de transformation est attachée un point de vue qui conduit l'analyste des données à ne conserver que les éléments qui sont en relation avec le point de vue. En effet, le modèle des données à concevoir est complété et affiné au fur et à mesure. En ce sens, la FVD rejoint aussi la définition de la conception selon l'AFNOR.

Les différentes pièces du modèle des données construit interagissent. Les décisions de conception ne sont pas indépendantes les unes des autres, ce qui fait référence à la conception en architecture. Les différentes phases de ce processus de conception sont marquées par l'élaboration d'un dossier et une étape de validation. Ce dossier sert aux étapes suivantes. En FVD (3^{e} variante du modèle de Ankerst par exemple), il n'existe pas

un ordre préétabli pour le choix des variables à utiliser pour les coupes servant à la création d'un modèle des données. Il n'y a donc pas de chemin pré-établi vers la solution.

Une des difficultés relative à l'activité de conception et qu'on retrouve en fouille visuelle des données réside dans le fait qu'il n'existe pas une seule « bonne » solution. Tout comme dans les autres activités de conception, l'évaluation de la solution en FVD (modèle de données) est une tâche difficile car il s'avère irréalisable de générer toutes les solutions possibles. Par contre, il existe des spécificités de la conception en fouille visuelle de données. L'activité de conception dite « moderne » est collective. En FVD, le modèle des données est conçu la plupart de temps par un seul analyste des données.

A ce niveau, on pourrait se poser la question de savoir pourquoi assister l'activité de conception en FVD ? La suite de cette section apportera des éléments de réponse à cette question.

5.1.2 Nécessité d'assistance en conception du modèle des données

Les progrès scientifiques et techniques ont pour principal objectif de libérer les hommes de tâches répétitives et pénibles. A notre connaissance, toute activité de conception est pourvue d'outils d'assistance. Visioconférence, « chat », Decision Systems, Negociation Support Systems, AutoCAD, ArchiPlan en sont des exemples. Dans le domaine plus spécifique de l'informatique, des recommandations ergonomiques pour la conception et le développement des interfaces [Bastien et Scapin, 1993], [Nielsen, 1993a], ont été développées ainsi que des méthodes formelles pour une assistance à la conception de logiciels.

A ces raisons nous pouvons ajouter le fait que l'assistance à la conception du modèle des données pourrait augmenter la productivité des utilisateurs. En effet, l'outil d'assistance à l'utilisateur aurait pris en considération ses préférences et ses habitudes de travail d'où la nécessité d'une modélisation de l'utilisateur. D'autres points positifs seraient la réduction des erreurs susceptibles d'être commises par les utilisateurs, l'augmentation de l'acceptabilité de l'outil de FVD, un gain de temps d'exécution, la facilité de navigation, la cohérence visuelle et l'indication de la position courante des traitements.

Cependant, l'assistance en FVD est un processus difficile à mettre en œuvre. En effet, la FVD possède plusieurs dimensions (perception ou interaction pour ne citer que celles la). Pour une assistance aux utilisateurs, il est nécessaire de se plonger dans chacun de ces domaines afin de relever ce qui pourrait être très utile à cet effet. Ceci dit, des différentes dimensions citées ci-dessus émergent des domaines de recherche pourvus de résultats éloquents et susceptibles d'être adaptés et réutilisés dans notre contexte. L'analyse de la situation du travail en général et plus particulièrement la modélisation de l'utilisateur que nous avons décrit dans les chapitres 3 et 4 en sont des exemples. Il ressort de ces différentes analyses et évaluations qu'il est nécessaire d'aider les utilisateurs à opérer les différents choix décrits par le modèle de tâches. Notamment, le choix de la meilleure méthode d'analyse de données à exécuter pour un problème donné. La section suivante présente les travaux réalisés afin de pouvoir mettre en œuvre de façon concrète ce système d'aide dans un outil de FVD.

5.2 Assistance à la conception du modèle de données : retrouver les meilleurs algorithmes de classification supervisée

La fouille de données nécessite la mise en oeuvre, explicite ou non, de méthodes statistiques classiques (graphiques, sondages, composantes principales, correspondances multiples, classification hiérarchique, nuées dynamiques, discriminante, k plus proches voisins, segmentation, régression linéaire, logistique) ou moins classiques (arbres de classification et de régression, modèles graphiques d'indépendance conditionnelle) ou d'intelligence artificielle (perceptron multicouche, réseau auto associatif et bayésien, apprentissage et règles d'induction, reconnaissance de formes).

Afin de faciliter le choix de la meilleure méthode d'analyse de données, l'idéal pour un système d'ECD aurait été d'implémenter une de ces techniques, pouvant être utilisée pour la résolution de tout type de problème. Malheureusement, tout le monde s'accorde sur le fait qu'il n'existe pas de méthode d'analyse de données qui surpasse toutes les autres pour la résolution de différents types de problèmes, un algorithme peut être performant pour un problème donné et non performant pour un autre. Pour le traitement d'un problème soumis en entrée de l'environnement de fouille, se pose donc le problème de choix des (de l') algorithme(s) le(s) plus approprié(s) à cet effet. La responsabilité de ce choix dans la plupart des systèmes de fouille a été laissée aux soins du spécialiste des méthodes d'analyse (statisticien) qui est l'utilisateur final de l'outil. Les outils de FVD peuvent aussi être utilisés par des utilisateurs spécialistes du domaine des données. Les avantages d'une telle approche sont : l'utilisation de l'expertise du domaine des données tout au long du processus de fouille, la compréhensibilité et la confiance dans le modèle de données construit sont accrues car l'utilisateur a participé à sa construction, l'utilisation des capacités humaines en reconnaissance de formes.

Cette approche a cependant des inconvénients. La sélection de l'algorithme d'analyse de données par exemple n'est pas toujours triviale pour le spécialiste des méthodes d'analyse, encore moins pour le spécialiste des données. Il est donc nécessaire d'aider l'utilisateur afin qu'il fasse les meilleurs choix et qu'il conçoive le meilleur modèle des données. Dans cet ordre d'idées, nous avons rapproché le problème du choix d'algorithme d'analyse de données d'un problème de décision avec la possibilité de faire des prédictions.

Le paragraphe suivant situe plus explicitement le cadre de ce travail.

5.2.1 Cadre de l'étude : la fouille visuelle de données

Notre domaine d'étude rappelons-le est la FVD. Le modèle de tâche (figure 1.21, chapitre 1) en FVD spécifie qu'après la sélection des données à traiter, l'utilisateur procède au choix de la méthode de visualisation à appliquer aux données et/ou au choix de la méthode d'analyse de ces données qui peut être automatique ou interactive.

5.2.2 Etat de l'art : synthèse des travaux en prédiction de performances d'algorithmes de classification

Le choix des (de l') algorithme(s) le(s) plus approprié(s) pour la résolution d'un problème donné dans un environnement de fouille de données a fait l'objet de plusieurs travaux. Dans un premier temps, ces travaux visaient la sélection d'un unique algorithme pour la classification des données. Nous pouvons citer par exemple, des travaux basés sur l'exploitation des connaissances des experts en ce qui concerne l'applicabilité des algorithmes [Brodley, 1995]. L'inconvénient de cette approche est qu'il est impératif de disposer d'experts en chacune des méthodes d'analyse de données du système tout au long de son cycle de vie. Le rôle des experts est de produire des connaissances nouvelles pour la mise à jour de la base des règles. Il en est de même pour les approches telles que celles de :

- ([Engels, 1996], [Engels et Theusinger, 1998]) qui emploient la décomposition tâche méthode de [Chandrasekaran et al., 1992] pour mettre en oeuvre un module conseil aux utilisateurs tout au long du processus de fouille de données, ce qui permet de les guider à un haut niveau et d'améliorer par étapes le processus. A cet effet, des plans finis sont compilés dans des manuscrits pour l'exécution, permettant à l'utilisateur de construire le meilleur plan en utilisant un nombre limité d'opérations.
- [Kerber et al., 1998] qui ont documenté le processus de FD en employant des liens actifs pour programmer de façon visuelle les processus de FD et pour la rationalisation des choix principaux de conception. Ils rassemblent ces descriptions dans un dossier, cette approche facilite la réutilisation des processus de FD. Le résultat obtenu est un système de gestion de la connaissance pour des processus de FD.

Une autre approche consiste en l'utilisation de méta-règles issues des études expérimentales comme support à la prédiction des algorithmes appropriés pour la résolution d'un problème donné [Brazdil et Soares, 2000]. En effet, ces auteurs étudient le classement de différents algorithmes d'induction. Ce classement est basé sur leur performance pendant leur exécution sur des ensembles de données répertoriés. [Petrak, 2000] présente une analyse persuasive de l'efficacité relative à l'emploi de sous échantillons de l'ensemble des données soumis à l'analyse pour prévoir quel algorithme apportera la plus faible erreur sur l'ensemble de données total. Pour StatLog [Michie et al, 1994], l'étude consiste à savoir quels algorithmes d'induction pourraient être employés dans des circonstances particulières données. Les résultats de ce projet sont constitués pour un ensemble de données et d'algorithmes, des algorithmes jugés applicables ou non applicables suivant leur performance. Avec ces différentes approches, se pose le problème de traitement des cas nouveaux. Les règles ou méta-règles issues d'expérimentations sont produites à l'initialisation du système. On assiste au développement de techniques nouvelles d'analyse de données. L'inconvénient majeur des différentes solutions présentées dans ce paragraphe réside dans le fait que sans connaissance préalable des performances des algorithmes récemment développés, la prédiction du meilleur algorithme pour un ensemble de données est faite par approximation (adaptation) aux solutions de cas traités. Il s'en suit une propagation de l'erreur de prédiction.

Il ressort de ce tour d'horizon que deux approches ont été utilisées pour l'acquisition

des connaissances en vue de la prédiction des performances des algorithmes automatiques d'analyse de données. Les sous sections suivantes présentent plus explicitement chacune de ces approches.

Une solution potentielle à notre problème aurait été de procéder au codage de l'expertise d'un ou plusieurs spécialistes en méthodes d'analyse de données dans un système. Ce codage nécessite un recueil de connaissances. Il est à noter que l'acquisition de connaissances d'experts possède des inconvénients. En effet, il s'agit d'un processus difficile à réaliser. Les experts utilisent la plupart de temps des activités mentales dont ils ne sont même pas conscients.

Aussi, ne disposant pas d'experts en toutes les méthodes de classification de données pour une acquisition de connaissances, nous nous sommes appuyés sur d'autres stratégies. En effet, il ressort des projets tels que Statlog et Metal que les expériences réalisées sur un ensemble d'algorithmes et un ensemble de bases de données peuvent servir à la prédiction des performances des algorithmes.

Toute la difficulté à présent consiste à définir le mécanisme adéquat pour le codage et la maintenance de cette connaissance issue des expérimentations. De prime abord, le raisonnement à partir de cas s'y prête bien. Cette approche a déjà été utilisée à cet effet dans le cadre du projet Metal. Le raisonnement à partir de cas (RàPC) est une approche de solution utilisée en résolution de problèmes. Pour un nouveau problème à traiter (nouveau cas), les résultats des expériences passées (cas passés) sont adaptés et contribuent à sa résolution [Aamodt et Plaza, 1994].

5.2. 3 Support par des expérimentations au choix de la méthode d'analyse : description du problème

L'idée ici est de permettre à l'analyste des données d'obtenir, en fonction de l'ensemble de données du problème qu'il aurait à résoudre, la liste des algorithmes disponibles dans l'environnement classés du plus performant au moins performant. L'ensemble d'expériences réalisées soit sur les données soit sur les algorithmes forme la base de cas. Pour effectuer le choix de l'algorithme le plus adéquat pour un problème en entrée de l'environnement, on doit retrouver des cas similaires dans la base de cas et les adapter au cas considéré.

Pour cela, nous recherchons parmi les ensembles de données déjà traités, les plus similaires à celui en entrée du système à l'aide d'un algorithme des k plus proches voisins travaillant sur des mesures de comparaison des ensembles de données et un seuil de similarité. L'algorithme le plus performant ayant servi à l'exécution sur les données les plus similaires aux données du problème à résoudre est exécuté. La formalisation de notre problème est la suivante :

Etant donné :

- un ensemble *A* d'algorithmes candidats pour une tâche de classification supervisée dans un environnement de FVD,
- un ensemble *D* d'ensembles de données dont les performances (précision, temps

d'apprentissage, temps de test) sur chaque algorithme de l'ensemble d'algorithmes *A* sont connues,

- un nouvel ensemble de données *d* relatif au problème d'un utilisateur,

Il s'agit de procéder à un apprentissage par analogie :

- sélectionner dans l'ensemble *D* un sous-ensemble *S* d'ensembles de données tel que chaque élément de S soit similaire à *d*,
- retrouver les informations concernant les performances (précision, temps d'apprentissage, temps de test) des algorithmes de *A* sur les ensembles de données de *S*,
- prédire la performance des algorithmes de *A* sur *d* en fonction des performances des algorithmes de *A* sur les données de *S*.

5.2.3.1 Les ensembles de données à traiter

Les attributs d'un ensemble de données peuvent avoir une valeur nominale (attribut textuel), ou quantitative (valeur numérique), etc. [Card et Mackinlay, 1997].

Le tableau 5.1 donne une vue de la structure des ensembles de données.

Tableau 5.1 Structure des ensembles de données

	Variable 1	Variable 2	…	Variable n
Observation 1			…	
.	…	…	…	…
Observation q			…	

L'analyse des variables contenues dans les ensembles de données permet de calculer la similarité entre les ensembles de données à traiter.

5.2.3.2 Analyse des variables

L'objectif de l'analyse d'une variable [Jambu, 1999] est de pouvoir identifier en un seul coup d'œil les éléments essentiels de la répartition des individus selon cette variable. Originellement, faute de pouvoir visualiser directement la distribution des variables, on cherchait des identificateurs caractéristiques de répartition (moyenne, médiane, écart type), indicateurs qui permettaient le mieux de comparer les valeurs de cette variable dans le temps. Les éléments d'analyse des variables différent suivant leur type.

5.2.3.3 Analyse des variables quantitatives

Une variable quantitative est étudiée suivant plusieurs critères d'analyse : analyse de la tendance centrale, analyse de la dispersion autour de la tendance centrale, analyse de la forme. Les indicateurs de la tendance centrale sont : nombre unique, valeur type représentant l'ordre de grandeur de l'ensemble des indicateurs de la variable. Nous pouvons aussi citer : la médiane, la moyenne arithmétique, la moyenne généralisée :

quadratique, harmonique, géométrique, le mode, les quartiles et les déciles.

Les indicateurs de dispersion ont pour objectif de mesurer l'écart ou la dispersion de la variable autour de la valeur centrale. Comme exemples, nous pouvons citer : l'amplitude, l'intervalle interquartile, l'écart type, la variance, le coefficient de variation. En ce qui concerne les indicateurs de forme, on a : le coefficient d'asymétrie et le cœfficient d'aplatissement.

Les indicateurs de concentration sont : le rapport d'inter décile, l'indice de gini, l'indicateur de Theil et l'indicateur d'Atkinson.

5.2.3.4 Analyse des variables qualitatives

Pour l'analyse d'une variable qualitative, la modélisation statistique est un moyen simple permettant d'étudier ce qui équivaut à la tendance centrale pour les valeurs quantitatives et de classer la fréquence des modalités. En effet, les modalités les plus fréquentes sont aussi les plus probables et définissent la tendance centrale.

La mesure de diversité est définie par l'entropie de Shannon. Cette quantité de diversité peut être comparée à l'entropie maximale qui se manifeste quand toutes les possibilités sont égales.

Après avoir présenté des éléments d'analyse des variables qualitatives et quantitatives, nous allons voir les différents facteurs rentrant dans la comparaison des ensembles de données.

5.2.3.5 Eléments de décision pour la comparaison des ensembles de données

Des mesures statistiques permettent de définir la similarité entre ensembles de données. Ces ensembles de données ont des attributs qualitatifs et quantitatifs.

Tableau 5.2 Mesures de description des ensembles de données

Tableau 5.2 Mesures de description des ensembles de données		[Michie et al, 1994], [Seewald, 2002], [Kalousis, 2002]
n	nombre d'enregistrements	
p	nombre d'attributs	
k	nombre de classes	
bin	nombre d'attributs binaires	
nom	nombre d'attributs nominaux	
SDratio	ratio de l'écart type	
cancor1 [Köpf et Iglezakis, 2002]	premier coefficient de corrélation canonique	
fract1 [Köpf et Iglezakis, 2002]	première valeur propre	
Skewness [Michie et al, 1994]	coefficient d'asymétrie	
Kurtosis [Michie et al, 1994]	coefficient d'aplatissement	
H_A [Kalousis, 2002]	entropie d'attribut	
[Kalousis, 2002]	entropie moyenne des attributs	
H_X [Kalousis, 2002]	entropie de classes	
MCx [Kalousis, 2002]	entropie mutuelle des classes et d'attributs	
[Kalousis, 2002]	entropie mutuelle moyenne des classes et d'attributs	
EnAtr [Kalousis, 2002]	nombre équivalent d'attributs	

La comparaison des ensembles de données est possible grâce aux mesures du tableau 5.2. La définition et l'utilisation de ces différentes mesures ont été tirées des travaux tels que [Michie et al, 1994], [Seewald, 2002], [Kalousis, 2002] et [Köpf et Iglezakis, 2002].

En ce qui concerne l'évaluation des performances des algorithmes, les critères utilisés sont : le taux de précision obtenu par les modèles fournis par l'algorithme et le temps d'exécution (en phase d'apprentissage et en phase de test). Ces critères sont mémorisés dans l'historique des expériences issues de l'exécution des algorithmes.

5.2.3.6 Evaluation de la ressemblance entre ensembles de données

La similarité qui permet de mesurer le degré d'appariement entre deux ensembles de données consiste en la recherche de correspondances entre les descripteurs ou au calcul du degré d'appariement des descripteurs.

La mesure de la distance (euclidienne par exemple) entre les critères de

comparaison de deux ensembles de données X et Y permet d'évaluer le degré de ressemblance de la description de X par rapport à celle de Y.

Une normalisation permet d'obtenir des valeurs comprises entre [0,1] comme suit :

Pour $y = (y_0,\ldots,y_n)$ on a l'expression normalisée $Y = (Y_0,\ldots,Y_n)$ avec

La mesure de similarité prend sa valeur dans l'intervalle continu [0,1], la valeur 1 signifiant la parfaite inclusion de la première description dans la seconde. L'algorithme des k plus proches voisins présenté ci-dessus utilise cette notion de distance.

5.2.3.7 Algorithme des k plus proches voisins (kppv)

On donne :

X, un ensemble de données,

K, le nombre des plus proches voisins de cet ensemble de données à retrouver,

x l'ensemble de données à traiter.

L'idée est de chercher $x_1 \ldots x_k$ les K plus proches voisins de x dans X et de retourner

L'algorithme des k plus proches voisins est souvent utilisé dans les méthodes d'apprentissage supervisé pour le raisonnement à partir de cas. La phase d'apprentissage consiste à stocker les exemples de cas résolus. Le classement de nouveaux cas s'opère en calculant la distance entre les critères de description des données du cas à traiter et ceux des exemples de la mémoire d'apprentissage.

Cet algorithme a été utilisé par [Brazdil et Soares, 2000], [Brazdil et al, 2003], [Köpf et Iglezakis, 2002] et [Kalousis et Theoharis, 1999] dans le cadre de la prédiction des performances des algorithmes de classification supervisée de données, en vue du guidage des spécialistes des méthodes d'analyse de données.

L'idée de cette méthode en prédiction des performances d'algorithmes est la prise de décisions basée sur la recherche de un ou plusieurs cas similaires déjà résolus. En effet, l'algorithme cherche les k plus proches voisins du nouveau cas et prédit la réponse la plus fréquente de ces k plus proches voisins. La méthode utilise à cet effet deux paramètres : le nombre k et la fonction de similarité pour comparer le nouveau cas aux cas déjà classés.

5.2.4 Le raisonnement à partir de cas (RàPC)

Le RàPC est une méthode de résolution de nouveaux problèmes en adaptant les solutions de cas déjà traités. Un cas peut être défini comme une pièce de connaissance représentant une expérience. Pour un problème à résoudre, on recherche un cas déjà traité similaire au problème à traiter.

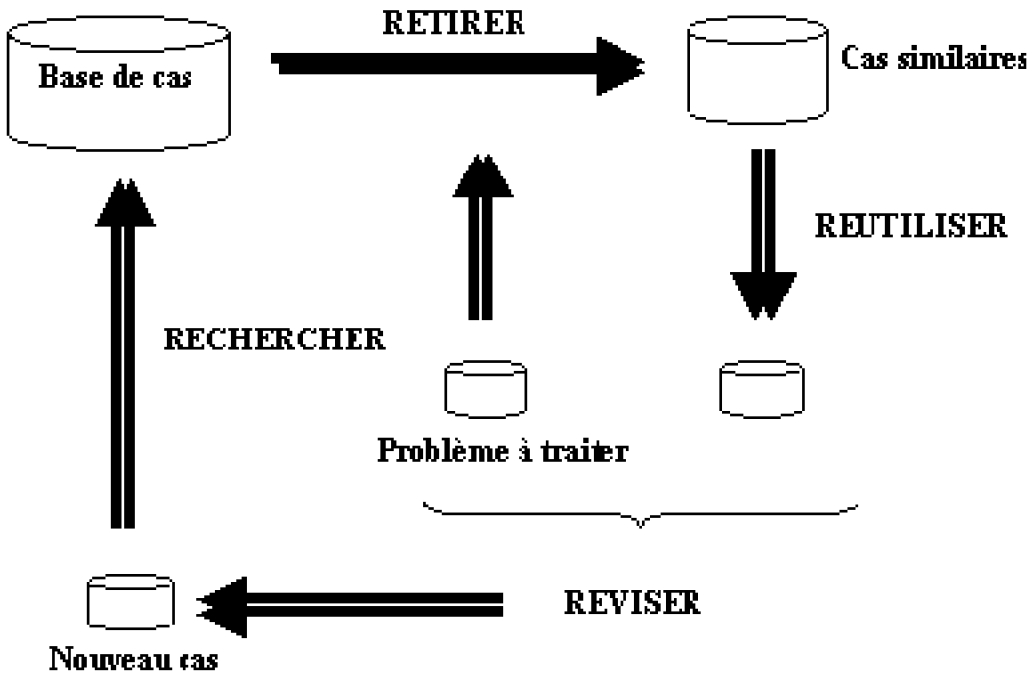


Figure 5.1 Principe du raisonnement à partir de cas

Dans le fonctionnement général d'un système à base de cas il existe une phase préalable d'initialisation de la base de cas. Pour un problème à résoudre, on assiste à la phase de recherche de cas analogues au cas à traiter, la phase d'adaptation des solutions trouvées à la résolution du nouveau cas, la phase de réutilisation de la solution adaptée et enfin la phase de maintenance de la base de cas.

5.2.4.1 Processus du raisonnement à partir de cas

Le raisonnement à base de cas est donc une approche de résolution de problèmes qui utilise des résultats basés sur des expériences passées pour résoudre des nouveaux problèmes.

Par application à la sélection de la méthode d'analyse de données en FVD, l'ensemble d'expériences réalisées soit sur les données soit sur les algorithmes dont les critères sont présentés dans le tableau 5.2 forme la base de cas. Pour effectuer le choix de l'algorithme le plus approprié pour un problème en entrée de l'environnement, l'idée ici est de retrouver des cas similaires dans la base de cas et les adapter au cas considéré. A cet effet, il est nécessaire que la base de cas soit construite soigneusement (représentation des cas, indexation, spécification intelligente des similarités entre les cas, facilité d'adaptation des cas et d'enrichissement de la base) et qu'il existe un moteur de recherche rapide et intelligent.

Notre formalisation du principe de cette approche telle qu'utilisée en prédiction d'algorithmes de classification supervisée de données est la suivante :

- chaque ensemble de données (D_j) de la base de cas est pourvu d'un ensemble de valeurs $V_j = a_{1j}, a_{2j}, \ldots, a_{nj}$,
- le problème à résoudre par l'utilisateur relatif à l'ensemble de données *X* a un ensemble $p = b_1, b_2, \ldots, b_n$ de critères,
- la meilleure solution *s* est un ensemble de données de *D* respectant la condition suivante : SIM(p, s) = $argmax(SIM(X, D_j))$.
- Pour aboutir à notre contribution, nous avons procédé en deux étapes. Dans un premier temps, nous avons implémenté et exécuté l'une des meilleures approches proposées pour la prédiction des performances des algorithmes d'analyse automatique de données [Brazdil et al., 2003]. Ensuite, nous avons recherché des moyens pour l'optimisation de cette solution. Enfin, l'un de ces moyens a été modélisé et implémenté. Ce moyen corrige les différents inconvénients des approches existantes décrites dans le paragraphe ci-dessous.

5.2.4.2 Inconvénients du raisonnement à partir de cas

Pertinence de la base de cas et apprentissage par adaptation : la base de cas traitée est pourvu d'un nombre limité de cas à l'initialisation du système par exemple. Dans les systèmes de prédiction à base de cas, quelque soit les ensembles de données D_j de *D* traités, il existe toujours un ensemble de données s répondant à la condition $s = argmax(SIM(p,D_j))$. La solution est injectée dans la base de cas, pourtant, elle est erronée pour la simple raison que la similarité entre le cas à traiter et son plus proche voisin dans la base de cas n'est pas forte. Le nouveau problème est ainsi résolu par adaptation aux connaissances acquises quelque soit la similarité de ce problème avec ces cas déjà traités. On constate que cette adaptation de cas traités à un nouveau problème entraîne une propagation de l'erreur de prédiction. La base sur laquelle s'appuient les travaux de prédiction contient alors des connaissances erronées. En effet, le choix de l'algorithme à exécuter est basé sur la performance du (des) problème(s) le(s) plus similaire(s) (maximum de similarité) au problème à résoudre. La similarité est une fonction qui prend ses valeurs dans l'intervalle [0, 1]. Des expérimentations nous ont permis de réaliser qu'il était possible d'obtenir comme taux de similarité maximale (entre le cas à traiter et les cas traités) 0.5 et que ce résultat aboutit à une proposition de solution à l'utilisateur pas tout à fait correcte. Plus précisément, le nombre de critères de comparaison des données étant élevé (une trentaine), seules les valeurs de similarité très proches de 1 sont réellement significatives d'après ces expérimentations. Pour remédier à cette situation, des expérimentations ont permis de fixer un seuil de similarité (seuil de rejet) en deçà duquel on ne considère plus que les problèmes sont similaires.

Evolution de l'environnement de fouille de données : on assiste de plus en plus au développement de techniques performantes et innovantes de fouille. Le raisonnement à partir de cas a cependant été utilisé pour la prédiction d'algorithmes dans des environnements de type statique avec une seule phase de collecte de données (connaissances). Il est donc nécessaire de prévoir l'ajout de nouvelles techniques dans un environnement de fouille et de prendre des dispositions en vue de prédictions. A notre

connaissance, il n'existe pas de module du système de RàPC capable de prendre la décision de lancer par exemple des processus (programmes d'analyse des données) en vue de l'acquisition et la gestion de connaissances parallèlement à l'exécution du système de RàPC. Pourtant, les systèmes de fouille de données peuvent être évolutif en ce sens que de nouveaux algorithmes ou de nouveaux ensembles de données doivent pouvoir être traités. Ce traitement implique la nécessité d'une phase d'acquisition de connaissances en ce qui concerne les critères de comparaison de l'ensemble de données ou du nouvel algorithme. Le RàPC repose donc sur le principe de la réutilisation des solutions et d'apprentissage à partir de ces cas traités. Ainsi pour un nouveau problème, la solution est obtenue par adaptation des cas de la base déjà traités au nouveau cas, des approximations successives sont opérées, étant donné qu'il n'existe pas à priori une entité pouvant être chargée de l'acquisition et de la gestion instantanée des connaissances expérimentales dans le modèle de base du RàPC. Pour pallier à cette situation nous avons opté pour un système multi agents (SMA) et nous avons pourvu l'un des agents du SMA d'une base de cas (connaissances) maintenable. Cet agent est chargé de l'acquisition des connaissances durant l'exécution du programme de support à la décision. En effet, un système à base de connaissances stocke des questions et éventuellement leurs réponses au fur et à mesure qu'il les découvre. Le comportement d'un système à base de connaissances peut être modifié significativement sans qu'il y ait besoin de le recompiler. Cette modification peut se faire de façon déclarative, c'est-à-dire en ajoutant un élément de connaissance qui a un sens indépendamment du programme. La particularité de notre système est que les connaissances seront interprétées non pas par des humains mais par la machine (un agent logiciel) qui s'en servira pour lancer l'exécution d'autres processus ou pour incrémenter sa base de connaissances.

L'approche ainsi proposée permet une intégration de nouvelles connaissances en ligne, une activation de la maintenance de cas traités suivant les valeurs obtenues par calcul de degrés d'appariement entre cas à traiter et cas résolus comme le montrent les modèles présentés dans les sections 5.2.5 et 5.2.6.

5.2.5Traitement de l' évolutivité de l'environnement de fouille

Nous présentons dans cette section le modèle élaboré pour la prise en compte de l'ajout de nouveaux algorithmes et de nouveaux ensembles de données dans l'environnement de fouille.

Ajout d'une méthode d'analyse de données

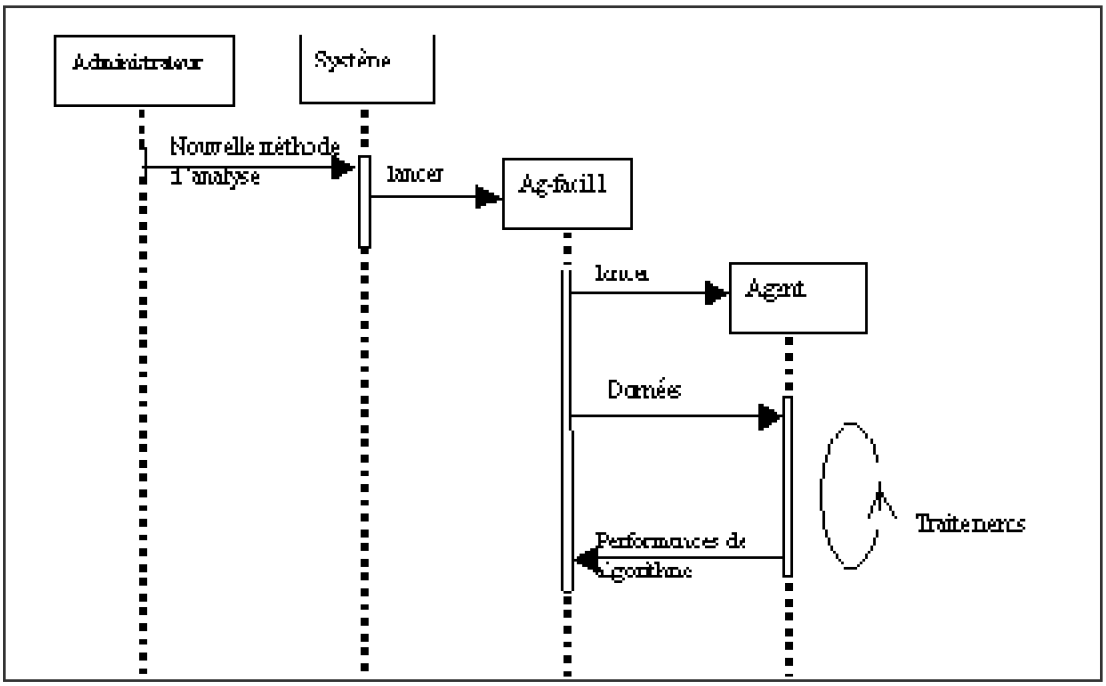


Figure 5.2 Modèle de l'ajout d'une méthode d'analyse de données à l'environnement

Trace de scénario de l'ajout d'une méthode d'analyse de données

Lorsque l'administrateur de l'environnement de fouille ajoute une méthode d'analyse de données à l'environnement, le système lance l'agent facilitateur des agents méthodes d'analyse (ag-facil1). Cet agent lance l'agent chargé de l'exécution de la nouvelle méthode d'analyse de données et lui fournit pour des traitements chaque ensemble de données déjà traité par les autres méthodes d'analyse de données de la plate forme. Les résultats sont sauvegardés dans la base de cas traités.

5.2.6 Ajout d'un nouvel ensemble de données

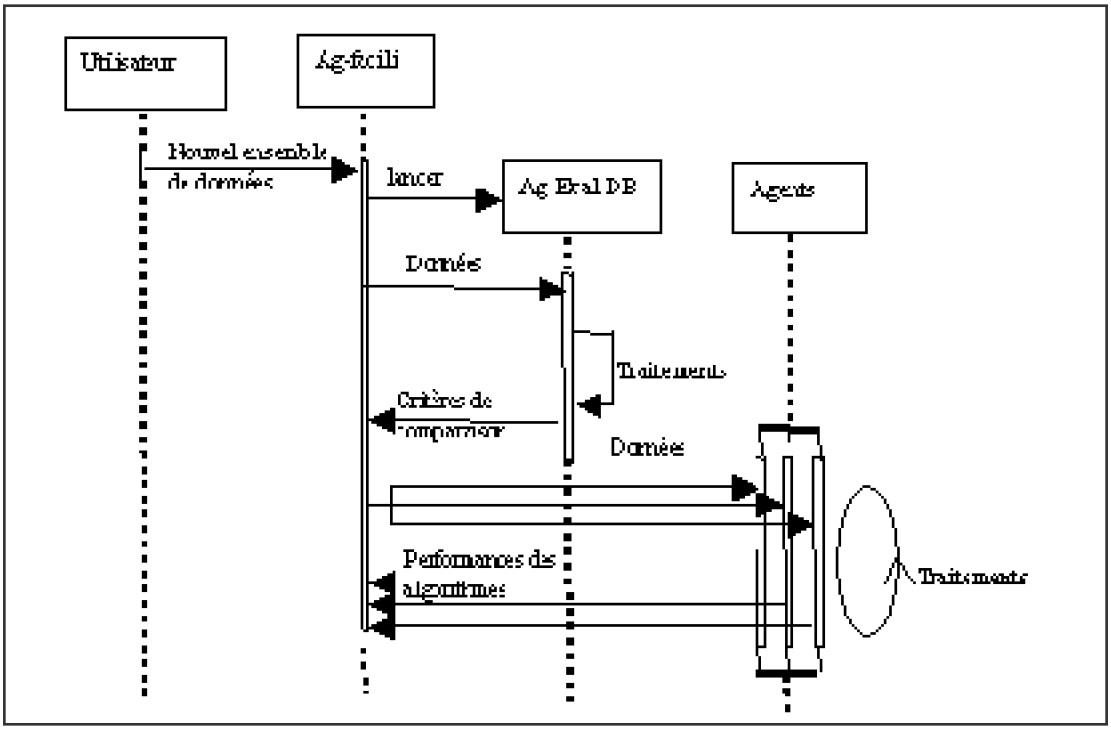


Figure 5.3 Modèle de l'ajout d'une méthode d'analyse de données à l'environnement

Trace de scénario de l'ajout d'un nouvel ensemble de données

Lorsqu'un utilisateur désire traiter un ensemble de données qui n'a jamais été traité sur la plateforme, le système lance l'agent facilitateur (ag-facili). Cet agent lance l'agent chargé du calcul des critères de comparaison des ensembles de données (Agent Eval DB). Si la ressemblance est trop faible, l'agent facilitateur lance aussi les agents chargés de l'exécution en parallèle des méthodes d'analyse de données de l'environnement de fouille et leur fournit pour traitement le nouvel ensemble de données. Les résultats sont sauvegardés dans la base de cas traités.

Avant de procéder à la modélisation complète de l'approche proposée, la section suivante présente une matérialisation concrète à travers une étude de cas des inconvénients du RàPC tel qu'utilisé en prédiction de performances d'algorithmes d'analyse de données.

5.3 Evaluation d'une approche usuelle des prédictions de performances : algorithme des k-ppv

5.3.1 Méthode d'évaluation : apprentissage suivi de vérification

Pour évaluer les capacités prédictives de l'approche des plus proches voisins, nous avons cherché à mesurer la pertinence de la liste classée par performance (précision, temps d'apprentissage, temps de test) décroissante d'algorithmes de classification délivrée par cet algorithme. Pour ce faire, nous avons un ensemble A constitué de 25 algorithmes de classification supervisée dans un ordre quelconque.

L'évaluation s'est déroulée en deux étapes. Pour la première étape, partant d'une base de cas non exhaustive (base de cas du début des traitements dans l'environnement de fouille de données), nous avons voulu prédire les performances des algorithmes sur deux nouveaux cas à traiter. Pour le second test, la base de cas est plus exhaustive mais le problème à résoudre n'est pas très semblable aux cas déjà traités.

5.3.1.1 Premier test

Pour ce test, nous avons considéré un ensemble D constitué de cinq ensembles de données [Michie et al, 1994], [Blake et Merz, 1998], [Metal, 2005] dont les valeurs des mesures de comparaison sont connues,

Le premier ensemble de données du problème à résoudre est : d_1 = *kldigits* et le second ensemble de données d_2 = *heart*.

Les performances des algorithmes de *A* sur *D* sont connues, plus précisément, elles sont issues des précédentes exécutions dans l'environnement.

Tout le monde s'accorde sur le fait que l'algorithme des kppv permet d'obtenir de meilleurs résultats pour *k=1*. Dans le cadre de ce test, nous nous sommes donc limités au plus proche voisin de l'ensemble de données à traiter.

Application de l'algorithme des plus proches voisins

Le plus proche voisin de d_1 dans *D* est l'ensemble de données *Digits*.

Le plus proche voisin de d_2 dans *D* est l'ensemble de données *Shuttle*.

Prédiction des performances des algorithmes de *A* sur les données similaires à d_1 et d_2

Pour la performance des algorithmes, nous avons choisi de travailler avec les cinq meilleurs algorithmes ayant servi à l'exécution de chaque ensemble de données (Digits et Shuttle), ensembles de données plus similaires à d_1 et d_2. Cette information est tirée de l'historique des exécutions passées.

Nous avons choisi de travailler avec un paramètre *k=5* pour mieux matérialiser la propagation de l'erreur de prédiction tout au long des traitements.

L'idée de la prédiction est la suivante : s'il existe un ensemble de données déjà traité similaire à l'ensemble de données du problème de l'utilisateur, l'exécution des algorithmes les plus appropriés au traitement de l'ensemble de données traité pour le problème de l'utilisateur aboutira aux meilleurs résultats. Les meilleurs algorithmes à utiliser pour Digits sont KNN, Quadisc, LVQ, Cascade, Alloc80 et les meilleurs algorithmes à utiliser pour Shuttle sont NewId, BayesTree, Cn2, Cal5 et CART.

Tableau 5.3 Prédiction des performances des algorithmes de A sur d1 et sur d2.

	1	2	3	4	5
Digits	KNN	Quadisc	LVQ	Cascade	Alloc80
Shuttle	NewId	BayesTree	Cn2	Cal5	CART

Performance effective de d_1 et d_2 et évaluation de la qualité des prédictions

Après avoir obtenu les prédictions de performances des algorithmes de *A* sur d_1 et sur d_2 nous avons cherché à savoir quelle était la pertinence de ces prédictions. Pour cela, nous avons calculé la performance effective des algorithmes de *A* sur d_1, par exécution de tous les algorithmes disponibles. Nous avons obtenu le tableau 5.4 ci-dessous. Les meilleurs algorithmes pour d_1 sont KNN, Alloc80, Quadisc, LVQ, Dipol92 et pour d_2 NaiveBayes, Discrim, Logdisc, Alloc80, Quadisc.

Tableau 5.4 Performance effective des algorithmes de A sur d1 et sur d2.

	1	2	3	4	5
d_1	KNN	Alloc80	Quadisc	LVQ	Dipol92
d_2	NaiveBayes	Discrim	Logdiscr	Alloc80	Quadisc

Les données de d_1 et celles de son plus proche voisin (*SIM* (d_1, *digits*) = *0.9998*) sont très similaires. On retrouve dans la prédiction des performances des algorithmes de *A* sur d_1 (tableau 5.4, deuxième ligne) pratiquement les mêmes algorithmes de classification supervisée que ceux de la performance effective de *A* sur d_1 (tableau 5.4, troisième ligne) à l'exception d'un seul (l'algorithme Cascade est prédit en quatrième position des algorithmes les plus appropriés au traitement de d_1 alors que dans la performance effective de d_1, on obtient Dipol92).

En ce qui concerne les données de d_2, elles ne sont pas similaires à celles de shuttle (son plus proche voisin) (*SIM*(d_2, *shuttle) = 0.93)*. Aucun algorithme prédit par le plus proche voisin (tableau 5.6) n'apparaît dans la liste de performance effective de *A* sur d_2 (tableau 5.8).

5.3.1.2 Second test

L'ensemble *A* reste identique à celui du pmier test, l'ensemble *D* contient beaucoup plus d'ensembles de données (19 au total) [Michie et al, 1994], [Blake et Merz, 1998], [Metal, 2005], les valeurs des mesures de comparaison sont connues, on a :

L'ensemble de données du problème à résoudre : *d* = *heart*.

Les performances des algorithmes de *A* sur *D* sont connues.

Application de l'algorithme des plus proches voisins

Tout comme dans l'exemple précédent, afin d'observer l'erreur de prédiction, nous nous sommes limités aux cinq plus proches voisins de *d,* avec *k = 5*. Nous avons donc comme plus proches voisins de *d* (Heart) les ensembles de données TseTse, NewBelgian, SatImage, Credit et Australian.

Prédiction des performances des algorithmes de *A* sur *d*

Pour la performance des algorithmes, nous avons aussi choisi de travailler avec les cinq meilleurs algorithmes de chaque ensemble de données.

La première colonne du tableau 5.5 représente les noms des ensembles de donnés les plus proches de *d*, du plus similaire au moins similaire. Pour chacun de ces ensembles de données, les colonnes de 2 à 6 représentent par ordre décroissant de performances,

les algorithmes de *A* appropriés au traitement des ensembles de données de la colonne 1.

Tableau 5.5 Prédiction des performances des algorithmes de A sur d en fonction des cas similaires à d

	1	2	3	4	5
TseTse	Cn2	IndCART	NewId	CART	Smart -Ac2
NewBelgian	Smart	IndCART	NewId	C4.5	Ac2
SatImage	KNN	LVQ	Dipol92	RBF	Alloc80
Credit	C4.5	IndCART	Cal5	Smart	Castle
Australian	Cal5	Itrule	Discrim	Logdiscr	Dipol92

L'ensemble de données TseTse est le plus similaire à *d (SIM(TseTse, heart)* = 0.9370), les algorithmes les plus performants ayant servi au traitement de TseTse devraient être les plus performants pour *d*.

Performance effective de *d* et évaluation de la qualité des prédictions

Après avoir obtenu les prédictions de performances des algorithmes de *A* sur *d*, nous avons cherché à savoir quelle était la pertinence de cette prédiction. Pour cela, nous avons recherché la performance effective des algorithmes de *A* sur *d*. Les meilleurs algorithmes pour le traitement de l'ensemble de données *d* sont : NaivesBayes, Discrim, Logdiscr, Alloc80, Quadisc.

Le meilleur algorithme obtenu par exécution de tous les algorithmes de *A* sur *d* est donc NaiveBayes. Cet algorithme n'apparaît pas dans la liste des 5 meilleurs algorithmes prédits pour TseTse ni par les autres voisins de *d*. Discrim et Logdisc, respectivement deuxième et troisième algorithmes plus performants pour le traitement de *d* sont retrouvés en troisième et quatrième position du classement des performances des algorithmes du cinquième plus proche voisin de *d* (Australian). Alloc80 qui occupe la quatrième position du tableau 5.11 est le cinquième algorithme plus performant du troisième plus voisin de *d* et Quadisc n'apparaît pas dans les prédictions. L'explication relative à ces résultats est la suivante : malgré le fait que les ensembles de données TseTse, NewBelgian, SatImage, Credit et Australian soient les plus proches voisins de *d*, leur similarité avec *d* est faible.

5.3.1.3 Conclusion

Lorsque la similarité entre l'ensemble de données du problème à résoudre et son plus proche voisin est faible, la prédiction des performances des algorithmes n'est pas très fiable. Pour les besoins de maintenance de la base de cas, cette prédiction erronée est sauvegardée dans la base de cas, ce qui entraîne une propagation de l'erreur de prédiction lors des futures prédictions.

Pour éviter ce problème, nous avons fixé à partir de simulations un **seuil de similarité** en deçà duquel nous ne considérons plus deux ensembles de données comme similaires.

Tableau 5.6 Algorithmes de prédiction de performance

1ppv	Nouvelle approche
Pour chaque ensemble de donnéesD_j de l'ensemble d'apprentissage **faire** calculer $SIM(X, D_j)$; fin pour rechercher $max\ (SIM(X, D_i))$	**Pour** chaque ensemble de donnéesD_j de l'ensemble d'apprentissage **faire** calculer $SIM(X, D_j)$; fin pour rechercher $max\ (SIM(X, D_i))$ si $max\ (SIM(X, D_i))$>= Seuil alors utiliser les mesures de performance des algorithmes de A sur D_j pour la prédiction; sinon exécuter le système multi-agent afin de retrouver les performances des algorithmes de *A* sur X. fin si

En effet, nous avons effectué une méta-analyse en exécutant une trentaine d'algorithmes (*A*) de classification supervisée de données sur une centaine d'ensembles de données (*D*) connus (UCI), l'idée étant de procéder à un apprentissage suivi d'une vérification. Nous avons ensuite calculé les similarités entre les différentes paires possibles de ces ensembles de données. Nous avons retiré tour à tour les données (critères de comparaison et mesures de performances des algorithmes) d'un ensemble de données de la base de cas, puis nous avons prédits les performances des algorithmes de *A* sur cet ensemble de données en fonction des performances des algorithmes sur son plus proche voisin. Pour chaque couple (ensemble de données, plus proche voisin), nous avons évalué la proximité entre les résultats obtenus par prédiction des performances des différents algorithmes en fonction du taux de similarité entre les deux ensembles de données. Les résultats de ce procédé nous ont permis de fixer ce seuil de similarité.

L'application de cette approche permet par exemple pour les cas présentés dans la première série d'expérimentations de restreindre la liste des plus proches voisins d'un problème à traiter, réduisant ainsi le nombre de mauvaises prédictions. Par exemple pour le premier test, on aura par application de la nouvelle approche : le plus proche voisin de d_1 est :Digits. La similarité entre d_1 et Digits est au-dessus du seuil de similarité fixé. Les résultats de prédictions resteront les mêmes que ceux du tableau 5.7. Le plus proche voisin de d_2 est : aucun ensemble de données. Toutes les similarités entre d_2 et les ensembles de données de *D* sont inférieures au seuil de similarité fixé. En ce qui concerne d_2, puisqu'on a trouvé qu'aucun cas traité par l'environnement n'est similaire à d_2, l'exécution des algorithmes de *A* sur d_2 en parallèle permet d'acquérir des connaissances réelles quant à la performance des algorithmes de *A* sur d_2.

Pour le second test, l'approche que nous proposons, suite à la recherche des plus proches voisins de *d* renverra un message suivant lequel, aucun cas traité par la base des cas n'est proche de *d*. Le processus utilisé pour le traitement de d_2 servira à retrouver les performances des algorithmes de *A* sur *d*.

5.3.1.4 Troisième test

Nous avons : un ensemble *A* constitué de 25 algorithmes de classification supervisée.

$$A = \begin{Bmatrix} Ac2, Alloc80, Backprop, Bayes, BayesTree, C4.5, CART, Cal5, Cascade, \\ Castle, Cn2, Default, Dipol92, Discrim, ITrule, IndCART, KNN, Kohonen, \\ LVQ, LogDisc, RBF, SMART, Quadisc, NewId, NaiveBayes \end{Bmatrix}$$

Un ensemble *D* de 19 ensembles de données [Michie et al, 1994], [Blake et Merz, 1998] et [Metal, 2005], les valeurs des mesures de comparaison sont connues.

$$D = \begin{Bmatrix} Australian, BELGIAN, BT, Credit, Chromosome, Cut, Diabetes, Digits, DNA, Faults, \\ German, Head, kDigit, NewRELGIAN, Satimage, Segment, Shuttle, TsetTse, Vehicle \end{Bmatrix}$$

L'ensemble de données du problème à résoudre : d_3 = *kldigit*.

Les performances des algorithmes de *A* sur *D* sont connues.

Les cas le plus similaires à d_3 par application de la nouvelle approche quelle que soit la valeur de k sont Digits et Cut.

Les cas les plus similaires à d_3 par application de l'algorithme des kppv avec *k=5* sont : Digits, Cut, BT, Shuttle, Chromosome.

5.3.2 Synthèse

L'approche que nous proposons restreint la marge d'erreurs de prédictions en limitant la liste des plus proches voisins d'un ensemble de données (d_3 par exemple) à ses plus proches voisins les plus significatifs (dont la similarité est supérieure ou égale au seuil fixé).

Dans cet ordre d'idées, avec notre approche, les algorithmes prédits comme plus performants pour le traitement de d_3 sont : KNN, Quadisc, CN2, LVQ, Alloc80. Les plus performants réellement étant : KNN, Alloc80, Quadisc, LVQ, Dipol92.

Avec les plus proches voisins (k = 5), les cinq algorithmes prédits comme les plus performants pour le traitement de d_3 sont : NewId, KNN, Quadisc, BayesTree, CN2.

5.4 Solution proposée

Comme support à la décision des analystes de données, nous présentons un algorithme permettant de prédire les performances d'algorithmes de classification supervisée en fonction des données en entrée du système. A cet effet, un système de raisonnement à partir de cas est utilisé. La base de cas est constituée de l'ensemble des expériences déjà réalisées (applications des algorithmes de classification supervisée sur des ensembles de données et caractéristiques des résultats obtenus). Pour un nouvel ensemble de données à traiter, la première étape est alors de rechercher dans la base de cas, l'ensemble de données le plus similaire à celui en entrée. Pour cette recherche nous utilisons un algorithme des k plus proches voisins. La similarité est calculée sur un ensemble de mesures de comparaisons des ensembles de données. Contrairement aux approches existantes, et afin de ne pas considérer comme semblables deux ensembles de données qui ne le sont pas suffisamment, nous utilisons un seuil en deçà duquel les données en

entrée ne seront plus considérées comme semblables au plus proche voisin trouvé (on parle de mécanisme de rejet de distance [Dubuisson, 1990]). Deux cas se présentent alors, soit les données en entrée sont suffisamment semblables au plus proche voisin trouvé et alors on obtient le classement des algorithmes de classification suivant leur performance sur ce plus proche voisin, soit les données en entrée sont trop différentes du plus proche voisin et alors on exécute l'ensemble des algorithmes disponibles sur les données en entrée. Dans ce dernier cas, afin de limiter le temps d'attente de l'utilisateur, on peut prévoir d'exécuter les différents algorithmes en parallèle puisque ces exécutions sont totalement indépendantes les unes des autres.

5.4.1 Modélisation du système multi-agent

Le SMA proposé est constitué d'un agent facilitateur, d'un agent qui se charge de l'évaluation des critères de comparaison des ensembles de données. A chaque méthode d'analyse de données de l'environnement est affecté un agent qui est chargé de son exécution. Un second agent facilitateur se charge de la gestion de ces différents agents.

On rappelle que le besoin ici est de rendre l'utilisateur (spécialiste du domaine des données) autonome, c'est-à-dire capable à partir d'un ensemble de données, de choisir une méthode d'analyse de données, d'aboutir lui-même à des modèles de données et d'interpréter ces modèles. Il s'avère nécessaire de lui proposer de l'aide. Une approche simple d'aide au choix des meilleurs algorithmes de classification de données à exécuter pour une tâche donnée serait d'exécuter tous les algorithmes de l'environnement sur l'ensemble de données du problème à traiter. Mais, l'exécution d'une de ces méthodes peut s'avérer être longue. Il est impératif d'éviter un temps d'attente important, car cela pourrait susciter l'ennui et contribuer au désintéressement des utilisateurs.

L'approche proposée est fondée sur un système multi agents (SMA) et consiste à adapter les connaissances déjà acquises au choix des méthodes d'analyse de données si le seuil de rejet n'est pas atteint. Si le seuil de rejet est atteint, on procède à une acquisition de nouvelles connaissances de la machine en situation d'apprentissage pour l'aide. Contrairement aux systèmes usuels de raisonnement à partir de cas déjà traités, l'acquisition de nouvelles connaissances dont il est question est un processus indépendant de la situation d'apprentissage pour l'aide. Nous reviendrons plus explicitement dans la suite de cet article sur cette étape de prise de décision (section expérimentale). La notion d'agent et de SMA fait l'objet du paragraphe suivant.

En effet, nous proposons l'intégration de la fouille visuelle de données dans un système multi agents (SMA) en vue non seulement de contribuer à l'autonomie de l'utilisateur mais aussi d'améliorer la qualité des solutions et des temps d'exécution. Partant de la définition d'un agent logiciel de [Ferber, 1995], nous pouvons définir un agent comme une entité autonome, c'est-à-dire capable d'agir sur elle-même et sur son environnement en vue de réaliser ses objectifs. L'agent dispose d'une représentation partielle de cet environnement. Dans un environnement multi agents, l'agent peut communiquer avec les autres agents, son comportement est la conséquence de ses observations, de ses connaissances et des interactions avec d'autres agents.

L'inclusion du système de fouille de données dans un SMA consiste à définir une

société d'agents et les interactions possibles entre eux. Le premier avantage d'une telle approche repose sur cette définition des interactions entre agents qui permet d'opérer des traitements en parallèle. Le second avantage fait référence à l'autonomie des agents. Comme l'indique la définition d'un agent, il s'agit d'une entité capable d'agir sur elle-même. Il se pose un besoin d'acquisition de connaissances tout au long des traitements par le logiciel de fouille et sans recompilation du programme. Un agent pourra sans aide extérieure le faire. Les prises de différentes décisions peuvent aussi être déléguées à des agents. En résumé, la répartition de calculs et la coopération de machines distantes, propriétés des SMA sont bénéfiques pour ce faire. Par application d'une approche basée sur un SMA, un gain de temps pourra être opéré par rapport à l'exécution de toutes les méthodes d'analyse de données (décrite ci-dessus) et un gain de qualité sera aussi opéré par rapport aux méthodes existantes basées sur des approximations successives (adaptation de cas traités). L'aspect adaptatif et l'autonomie des SMA permettent au fur et à mesure des traitements d'acquérir des connaissances (en situation d'apprentissage pour l'aide au choix) relatives à la performance des algorithmes sur des problèmes à résoudre et de mettre à jour la base de connaissances. Le dernier avantage de l'approche proposée est la possibilité de mise à jour de la base de connaissances en l'absence d'experts en méthodes d'analyse de données.

Le système que nous décrivons dispose de trois mécanismes d'apprentissage : l'apprentissage en raisonnant par rapport aux cas déjà traités, l'apprentissage en s'adaptant pour résoudre par exemple des cas peu similaires aux problèmes déjà traités et enfin l'apprentissage en mémorisant qui permet de sauvegarder les solutions des cas déjà traités. Grâce à cette combinaison de stratégies, il nous est possible de traiter le problème de la propagation de l'erreur de prédiction comme nous le montrerons dans la partie réservée aux expérimentations qui fera suite à la modélisation du système.

5.4.1.1 Trace de scénario de la session d'analyse de données

Un utilisateur se connecte, l'agent facilitateur (Ag-Facil) est créé et lancé. L'agent facilitateur se met en attente des données du problème à résoudre. Dès réception des données, l'agent facilitateur lance l'exécution de l'agent chargé de l'évaluation des critères de comparaison des données (Ag-Eval-BD). Les informations concernant les méta-donnés calculés par l'agent Ag-Eval-BD sont retournées à l'agent facilitateur qui effectue la fusion de ces métas donnés avec les connaissances disponibles dans la base de connaissances. Le principe de la fusion est le suivant : s'il existe un cas traité dans la base de connaissances dont les critères de comparaison de données sont similaires aux critères de comparaison des données du problèmes à résoudre, les informations relatives à la performance des algorithmes de la plate forme sur ce cas similaire sont transmises à l'utilisateur. Si tel n'est pas le cas, l'agent facilitateur sollicite l'agent chargé de l'exécution en parallèle sur le réseau pour l'exécution de chaque algorithme de la plate forme sur le nouvel ensemble de données. L'exécution en parallèle des agents méthodes d'analyse permet un gain en temps de traitement.

Les résultats des performances des algorithmes sont transmis à l'agent facilitateur qui effectue une mise à jour de la base de connaissances et propose les différentes alternatives (solutions) susceptibles d'être choisies à l'utilisateur puis lui demande s'il

souhaite définir des poids à accorder aux différents critères de comparaison des algorithmes (compréhensibilité des résultats obtenus, vitesse d'exécution, temps d'apprentissage, temps de test, etc…). Après la réponse de l'utilisateur, l'agent facilitateur effectue une fusion et transmet les résultats finaux à l'utilisateur. Le résultat est une liste contenant l'ensemble des algorithmes disponibles sur la plate forme triée par ordre décroissant de performance. L'idée ici est de pouvoir exécuter le premier algorithme de la liste, au cas ou l'exécution du premier algorithme n'arrive pas à son terme, l'agent facilitateur lance l'exécution du second algorithme et ainsi de suite.

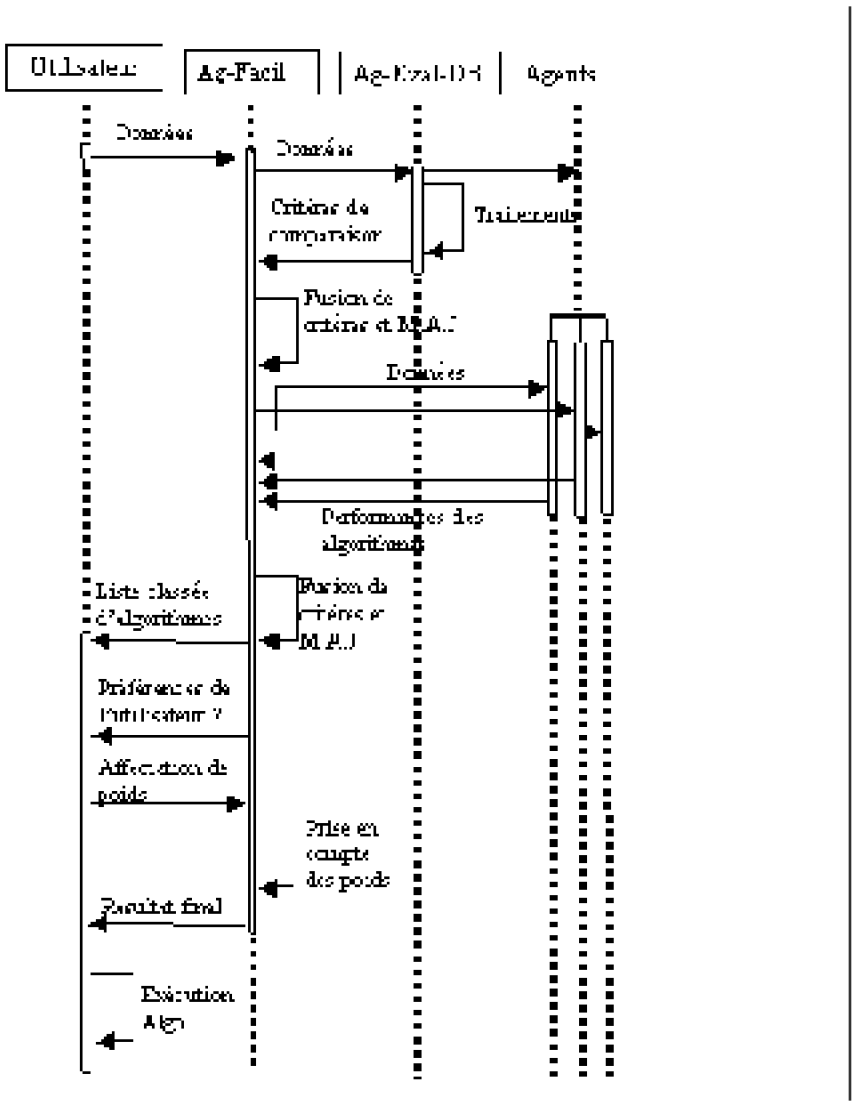


Figure 5.4 Modèle de la session d'analyse de données

5.4.1.2 Prise de décision (Agent Facilitateur)

Afin de matérialiser concrètement la prise de décision par l'agent facilitateur, nous reprenons la formulation d'un problème multicritère de décision faite par [Vansnick, 1990]. Le problème de sélection des (de l') algorithme(s) de classification de données à exécuter

peut être défini comme le modèle « *D*, *C*, *U* » :

- *D* est l'ensemble des méthodes de classification susceptible d'être choisies,
- *C* est l'ensemble des critères suivant lesquels les méthodes de classification pourront être choisies,
- *U* est l'ensemble des évaluations de l'impact des actions (solutions possibles) selon chacun des attributs ou critères.

Le paragraphe suivant décrit les critères sur lesquels sont fondés les décisions ainsi que la méthode utilisée pour leur évaluation.

5.4.1.3 Critères d'évaluation pour la prise de décision (C)

L'ensemble d'événements élémentaires ou aléas (critères) noté *C* porte tout d'abord sur les caractéristiques des données à traiter et ensuite sur les mesures de qualité des algorithmes, mesures obtenues par exécution des algorithmes de la plate forme sur des ensembles de données. En ce qui concerne les caractéristiques des données, nous avons :

- du point de vue descriptif : le nombre d'attributs, le nombre d'attributs symboliques, le nombre d'enregistrements, le nombre de classes, l'écart type, …
- pour les attributs quantitatifs, des mesures telles que la corrélation, les coefficients de skewness et de Kurtosis…),
- pour les attributs qualitatifs, des mesures telles que l'entropie, l'entropie relative…

La compréhensibilité des modèles fournis par l'algorithme, la vitesse d'exécution, le temps d'exécution, l'exactitude, la sensibilité du coût constituent les critères retenus quant à la mémorisation des expériences issues de l'exécution des algorithmes.

Ces différentes mesures serviront à la définition de la troisième composante du système de décision présenté ci-dessous.

5.4.1.4 Evaluation de l'impact des critères

L'évaluation de l'impact des critères fait suite à plusieurs étapes. Premièrement, pour l'initialisation de la base de connaissances, les phases suivantes sont exécutées : répertorier des ensembles de données, calculer les valeurs des critères de comparaison des données sur ces ensembles, effectuer des classifications sur les ensembles de données par les algorithmes disponibles sur la plate forme de fouille, évaluer les résultats puis garder une trace des résultats obtenus.

Deuxièmement pour un nouveau problème posé, il s'agit de retrouver le(s) problème(s) résolus dont les solutions peuvent être réutilisées pour le traitement du nouveau cas, puis, de proposer les algorithmes les plus performants ordonnés par performances décroissantes aux utilisateurs.

Typiquement, on représente un ensemble de données par un vecteur de critères du tableau 5.2 (nombre d'individus, moment d'ordre 3, 4, entropie, etc.). Après calculs, on

associe à chaque critère sa valeur. Les critères de comparaison sont tous à valeur numérique. Nous pouvons donc utiliser une variante de la métrique de Minkowski pour évaluer la ressemblance entre les ensembles de données. Chaque élément de U pour un cas à traiter sera obtenu par la variante choisie de la formule (1) dont les paramètres sont explicités dans le paragraphe suivant.

$$\varphi(x, y) = \left(\sum_i w_i \left| x_i - y_i \right|^p\right)^{1/p} \qquad (1)$$

x_i et y_i représentent respectivement les valeurs du ième attribut décrivant les critères x et y. w_i représente le poids attribué à cet attribut. Pour p = 1, on parle de distance de Manhattan, pour p = 2, on parle de distance euclidienne et pour p = □, on parle de distance de Chebychev.

Pour un attribut donné du vecteur de comparaison de critères, la distance est calculée à partir des différences $| x_i - y_i |$. Lorsque les attributs décrivant les exemples ont des domaines de valeurs de tailles différentes, il y a un risque de fausser les résultats. Pour p = 1 par exemple, les attributs ayant une grande dispersion de valeurs (grand écart $| x_i - y_i |$) sont implicitement favorisés, ce qui contribue à augmenter exagérément la distance [Bisson, 2000]. Il est donc nécessaire de procéder à une normalisation. Avec la normalisation, la formule que nous utilisons, spécifiée ci-dessous pondère équitablement à la fois les grands écarts et les petits écarts.

$$dist(x, y) = \sum_i w_i \frac{\left| x_i - y_i \right|}{\max(x_i) - \min(x_i)} \qquad (2)$$

Comme l'indique la section 5.2 consacrée à la synthèse des travaux réalisés dans le domaine de cette étude, d'autres approches méthodologiques peuvent être utilisées. Le paragraphe suivant présente leurs principales limites.

5.4.2 Etude de cas

Le problème à résoudre : choix d'un algorithme de l'ensemble *D* (description faite ci-dessous) pour le traitement de l'ensemble de données dont la description est faite dans le tableau 5.15.

A chaque étape des expérimentations, une comparaison des résultats obtenus par l'approche proposée avec ceux d'un algorithme réalisé dans le cadre du projet Metal est opérée. Deux bases de connaissances sont utilisées et mises à jour avec les résultats intermédiaires de chacune des approches.

L'ensemble *D* pour notre exemple illustratif est constitué de 12 algorithmes, implémentations de *WEKA* [Witten et Eibe, 2005] : *NB* [John et Langley, 1995), *Lwl* [Atkerson et al., 1997], *Ibk* [Aha et Kiber, 1991], *DTab* [Kohavi, 1995], *Smo* [Platt, 1998], *KStar* [Cleary et Trigg, 1995], *1R* [Holte, 1993], *Jrip* [Cohen, 1995], *PART* [Eibe et Witten, 1998), *Log* [le Cessie, 1992], *J48* [Quinlan, 1993], *AdaboostM1* [Freund et Schapire, 1996].

Le tableau5.15 présente une description des ensembles de données (issus du

répertoire de UCI [Blake et Merz, 1998]) qui ont servi à la phase initiale d'acquisition de connaissances. Comme éléments de description on a le protocole de test utilisé et quelques critères de comparaison des ensembles de données utilisés pour le calcul des similarités entre ensembles de données (nombre d'individus, de classes, d'attributs numériques et d'attributs catégoriques).

Tableau 5.7 Caractéristiques des ensembles de données utilisés pour l'évaluation

	Individus	Classes	Dim-num	Dim-cat	Ens-test
titanic	2201	2	0	3	10-fold
tic-tac-toe	958	2	0	9	10-fold
nursery	12960	5	0	8	10-fold
mushrooms	8124	2	0	22	10-fold
parity5_5	1024	2	0	10	10-fold

Le protocole de test utilisé pour le traitement de l'ensemble de données du problème à résoudre est le suivant : l'ensemble de données en entrée est *c_class_flares*, il comporte 1389 attributs, 8 classes, 10 attributs tous catégoriques.

A lapremière étape des traitements, l'initialisation des bases de connaissances a été faite, les mêmes informations y sont contenues. Pour la seconde étape, il s'agit deretrouver un des ensembles de données (pour cette présentation, nous nous limitons au plus proche voisin) répertoriés dans les bases de connaissances, significativement semblable à l'ensemble de données du problème en entrée, la distance entre les critères de comparaison de *c_class_flares* et ceux de tous les ensembles de données (tableau 1) est calculée, le résultat obtenu est le suivant :

– **Approche du projet Metal**, le plus proche voisin est *parity5_5*, *dist (c_class_flares,* parity5_5*) = 0.0122*.

– **Nouvelle approche proposée**, *dist (c_class_flares,* parity5_5*) > seuil_fixé* (0.005), *exécuter le système multi-agent pour retrouver les performances effectives.*

La troisième étape quant à elle consiste en des propositions de choix de l'algorithme de l'ensemble *D* à exécuter par les deux approches.

– **Approche du projet Metal** : exécuter l'algorithme le plus efficace ayant servi à l'exécution de *parity5_5* pour cet ensemble de données. Nous nous sommes limités pour cette présentation aux quatre meilleurs algorithmes. Le classement est le suivant : *PART* avec un taux de précision de 90.14%, *J48* (85.94%), *Ibk* (50.88%) et *AdaboostM1* (49.80%).

– Classement avec **la nouvelle approche proposée** : *1R* (84.30%), *Lwl* (84.30%), *Jrip* (84.30%), *AbaboostM1* (84.30%).

Les deux bases de connaissances sont mises à jours avec ces différents résultats. Afin d'évaluer la qualité des prédictions obtenues, nous allons en fonction des connaissances en notre disposition (bases de connaissances) retrouver l'algorithme de *D* le plus efficace pour le nouvel ensemble de données *m_class_flares* avec 1389 individus, 6 classes et 10 attributs catégoriques.

L'ensemble de données des bases de connaissances significativement semblable à l'ensemble de données du problème à résoudre est *c_class_flares* avec comme distance (dist (*m_class_flares*, *c_class_flares*) = 0.0034).

– Le classement avec **l'approche du projet Metal** est : ***PART*** avec pour précision 90.14%, ***J48*** (85.94%), ***Ibk*** (50.88%), ***AdaboostM1*** (49.80%).

– Le classement avec **la nouvelle approche proposée** est : ***1R*** (84.30%), ***Lwl*** (84.30%), ***Jrip*** (84.30%), ***AbaboostM1*** (84.30%).

– **classement réel** : ***1R*** (95.10%)**, Lwl** (95.10%)**, *AdaboostM1*** (95.10%)**, *DTab*** (95.10%)

De nombreuses autres expérimentations ont été opérées, il en ressort que lorsque la distance entre l'ensemble de données du problème à résoudre et ses plus proches voisins est grande (par exemple *parity5_5* et *m_class_flares*), la prédiction des performances des algorithmes est très souvent erronée par application des approches existantes. Concrètement, la distance entre *parity5_5* et *c_class_flares* est élevée, la prédiction est erronée. Le résultat obtenu est le suivant : *PART*, *J48*, *Ibk*, AdaboostM1 au lieu de *1R*, *Lwl*, *Jrip*, *AdaboostM1*. Pour les besoins de maintenance de la base de connaissances, cette prédiction erronée est sauvegardée dans cette base. Ce qui entraîne une propagation de l'erreur de prédiction aux futures prédictions. Pour revenir à l'illustration de notre approche présentée dans les paragraphes précédents, la prédiction des performances des 12 algorithmes traités sur l'ensemble de données *c_class_flares* (*PART*, *J48*, *Ibk*, *AdaboostM1*) par application d'une approche existante est sauvegardée dans la base de connaissances. Pour le traitement du nouvel ensemble de données, le résultat de la prédiction de performance est *PART*, *J48*, *Ibk*, *AdaboostM1* au lieu de *1R*, *Lwl*, *AdaboostM1* et *DTab*. Au fur et à mesure des traitements, on assiste à une propagation de cette erreur. Pour qu'une telle erreur ne se produise pas, nous avons introduit un seuil de rejet au-delà duquel on ne considère plus le plus proche voisin comme voisin. Le cas échéant, le SMA qui a été introduit permet d'acquérir les connaissances relatives aux performances effectives de l'algorithme. Ainsi le classement obtenu par notre approche sur *c_class_flares* est le classement réel des algorithmes traités. Le résultat de la prédiction des performances des algorithmes sur *m_class_flares* par notre approche est beaucoup plus fiable.

5.5 Conclusion

En fouille ou analyse de données, différentes méthodes ou stratégies peuvent être utilisées pour effectuer une tâche. La meilleure solution dépend bien entendu du problème à traiter d'où la nécessité d'une méthode permettant de guider l'utilisateur afin de lui permettre d'atteindre cette meilleure solution sachant que la prédiction de performance des algorithmes entraîne un gain en temps car l'utilisateur n'exécute pas tous les algorithmes d'analyse de données avant de choisir le plus approprié pour sa tâche.

Pour les besoins de cette étude visant un meilleur guidage des utilisateurs des environnements de FVD, nous avons développé deux systèmes. Le premier système

implémente un des meilleurs résultats obtenus dans le domaine de la prédiction des performances des algorithmes [Brazdil et al., 2003]. Ensuite nous avons procédé à des améliorations en tenant compte des enseignements venant de la première expérimentation pour le développement du second système.

Les méthodes proposées pour le choix de l'algorithme le plus performant pour un problème donné s'appuient sur des critères de comparaison des données et sur des critères de qualité des algorithmes. Pour un problème soumis en entrée de l'environnement de fouille, il s'agit de retrouver par analogie l'algorithme le plus performant ayant servi à la résolution d'un problème similaire au problème à résoudre déjà traité et dont le résultat est stocké dans une base de connaissances. Force a été pour nous de constater que ces travaux s'arrêtaient juste à la proposition soit d'un algorithme, soit d'une composition de processus probables de classification. Ces systèmes utilisent soit le raisonnement à base de cas, soit des ontologies à cet effet. Pour le raisonnement à base de cas, le système s'appuie sur l'expérience (cas déjà résolus). Ces cas résolus vont guider la compréhension des nouvelles situations. Un système de ce type effectue une recherche des cas similaires au problème à résoudre dans la base de connaissances. Même si un tel cas n'existe pas, une classification selon les cas similaires est effectuée et cette classification entraîne une perte d'information. En général, deux cas de figure se présentent : soit les cas nouveaux (non expérimentés) sont traités avec perte d'informations ou ces cas ne sont pas du tout traités. En effet, ces algorithmes de prédiction n'utilisent que des fonctions de calcul de similarités mais pas des fonctions pour l'adaptation.

Nous avons intégré la fouille graphique de données dans un SMA qui permet de par son autonomie l'évolutivité de l'environnement et des connaissances et de par son parallélisme un gain de temps de traitement. Contrairement aux approches existantes qui ne prévoient pas le traitement de nouveaux algorithmes et de nouveaux ensembles de données, l'approche proposée traite ce problème.

La principale limite de ce nouveau système concerne le temps d'exécution. En effet, la prédiction des performances pour les cas « nouveaux » ne permet pas un gain de temps, l'ensemble des algorithmes est exécuté sur le nouvel ensemble de données ou bien le nouvel algorithme est exécuté sur l'ensemble des ensembles de données. Cette limite constitue le principal atout de la méthode. En effet, ces différentes exécutions évitent une propagation de l'erreur de prédiction observée dans les systèmes existants, garantissant ainsi la qualité de connaissances sauvegardées.

Le paramétrage des algorithmes de classification supervisée constitue aussi un problème en guidage des utilisateurs. Comme perspectives au support au choix de meilleurs algorithmes de classification, on pourrait appliquer la méthode décrite dans ce chapitre, basée sur des critères de comparaison des ensembles de données à la définition de meilleurs paramètres compte tenu du paramétrage des cas déjà traités.

Chapitre 6 : Support au prétraitement des données en

FVD

6.1 Introduction

Nous nous intéressons au problème de prétraitement de grands ensembles de données pour la classification supervisée. Il ressort de l'état de l'art des méthodes de visualisation de données et du diagnostic des systèmes de fouille visuelle de données pour la classification supervisée [Fangseu Badjio et Poulet, 2005b] qu'il existe une limite quant à la quantité de données susceptible d'être représentée en une seule fois sur un écran. Pourtant, les progrès scientifiques et techniques permettent aux organisations de stocker des masses de plus en plus importantes de données et d'informations. Il arrive que l'ensemble de données à traiter avec les outils de FVD dépasse la limite tolérée par ces outils, il s'avère alors impossible ou pénible de procéder aux tâches interactives de fouille. En général, les données assez volumineuses comportent des informations bruitées, non significatives, redondantes, etc. Notre but est de réduire les informations contenues dans les ensembles de données volumineux aux informations les plus significatives.

Un ensemble de données est constitué d'attributs et d'observations. Réduire l'information contenue dans l'ensemble de données peut consister à agréger le nombre d'observations et extraire un sous-ensemble d'attributs pertinents. Avec un nombre élevé d'attributs et d'observations, la FVD nécessite une plus grande charge de travail de la part de l'utilisateur. La méthode décrite dans ce chapitre est très importante dans la mesure où elle permet de réduire la charge cognitive des utilisateurs et il s'agit d'un outil d'aide à la décision nécessaire à plus d'une catégorie d'utilisateurs potentiels d'environnements de FVD de données comme nous le verrons. De plus, elle permet de procéder à la construction interactive du modèle des données.Dans le domaine de l'extraction de connaissances dans les données, il existe des techniques expérimentalement validées pour l'amélioration des résultats des outils d'analyse de données en vue du traitement de grands ensembles de données. Deux approches sont utilisées dans ces techniques : une approche orientée données et une approche orientée algorithme. L'approche orientée données repose sur la discrétisation, la réduction du nombre d'observations ou la sélection des attributs pertinents de l'ensemble de données à traiter. Ce type d'approche permet ainsi de modifier l'ensemble de données initial par la sélection d'attributs (SA) et/ou la réduction d'observations.

L'approche orientée algorithme permet de concevoir des algorithmes rapides via l'optimisation de codes, la distribution des traitements, le parallélisme et la réduction de l'espace de recherche durant la construction du modèle de données.

Nous allons nous intéresser aux approches orientées données pour la sélection d'attributs les plus significatifs de l'ensemble de données à traiter et aux approches orientées algorithme qui prônent la réduction de l'espace de recherche durant cette sélection d'attributs. Un problème majeur se pose alors quant au choix d'une des

méthodes connues d'avance pour la sélection d'attributs par exemple, sachant qu'il n'existe pas de méthode qui soit meilleure que toutes les autres dans tous les cas de figure. Une solution qui constitue notre contribution dans ce travail serait d'utiliser une combinaison de techniques ou de stratégies (méthodes de sélection d'attributs). Pour ce faire, nous nous appuyons sur la théorie du consensus dont nous expliciterons le principe dans l'état de l'art dédié à ce sujet. L'utilisation de cette combinaison de stratégies ou d'expertises pour la sélection d'attributs peut être justifiée par l'un des faits suivants :

- il n'est pas possible de déterminer à priori quelle méthode de sélection de sous-ensemble d'attributs est meilleure que toutes les autres (en tenant compte des différences entre le temps d'exécution et la complexité (il s'agit ici de tolérer un temps d'exécution élevé pour un modèle qui nécessite également moins d'attributs)),
- un sous-ensemble optimal d'attributs n'est pas nécessairement unique,
- la décision d'un comité d'experts est généralement meilleure que la décision d'un seul expert.

Comme nous le verrons, l'algorithme de SA proposé qui combine des décisions de plusieurs experts reçoit en entrée des sous-ensembles d'attributs issus de plusieurs expertises et produit comme résultat un sous-ensemble unique d'attributs.

Les résultats obtenus après expérimentations permettent de conclure que l'approche proposée réduit de façon significative l'ensemble de données à traiter sans perte de qualité pour l'algorithme de classification utilisée et permet de les traiter de façon interactive.

Cette contribution commence par un état de l'art et la problématique du sujet abordé, puis, la technique utilisée pour la sélection d'attributs dans des ensembles de données est explicitée, ainsi que des problèmes relatifs à ce traitement. Ensuite, la théorie du consensus, l'algorithme de sélection d'attributs et la méthode d'agrégation des individus contenus dans les ensembles volumineux de données sont présentés. Enfin, nous procédons à des expérimentations avant la conclusion et les perspectives de ces travaux.

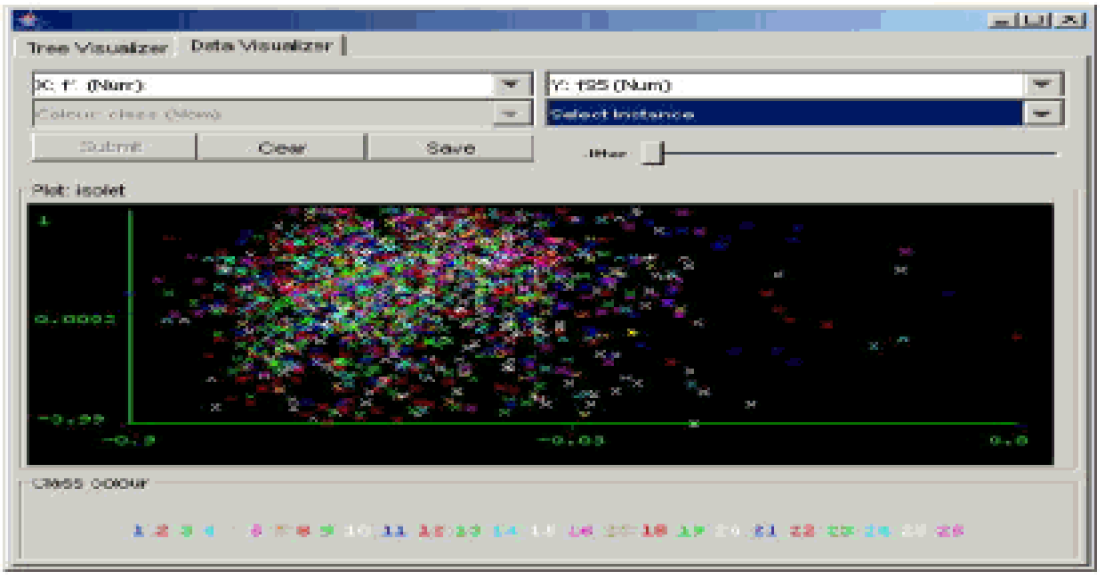


Figure 6.1 Représentation de l'ensemble de données Isolet (618 attributs, 1560 individus, 26 classes) sous forme de matrice en 2D [Chambers et al, 1983]

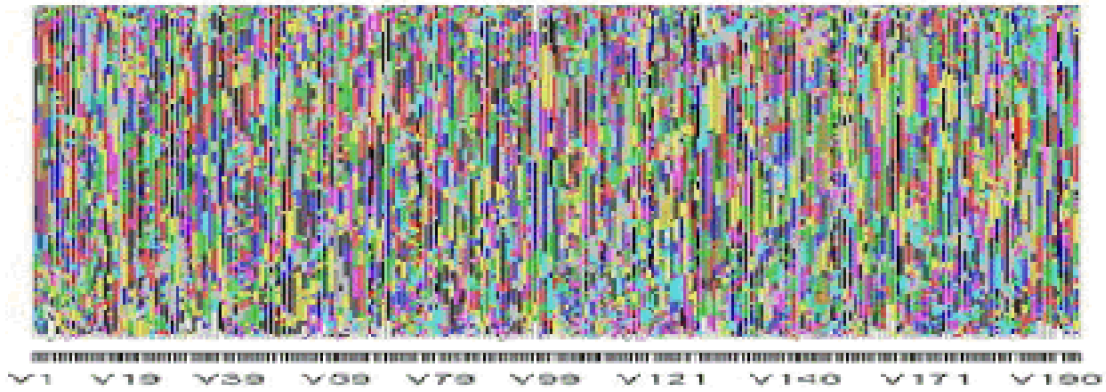


Figure 6.2 Représentation graphique d'un sous ensemble de 200 attributs de l'ensemble de données Isolet avec les coordonnées parallèles [Inselberg, 1985]

6.2 Etat de l'art et problématique

Il nous semble important de revenir sur l'état de l'art des méthodes de visualisation de données présenté au chapitre 1 qui fait état de plusieurs techniques de visualisation de données multidimensionnelles utilisables en FVD parmi lesquelles on distingue les techniques orientées pixels, les matrices 2 ou 3D, les coordonnées parallèles, etc. Dans la plupart de ces techniques de visualisation [Keim, 1996], le nombre de données susceptibles d'être représentées en même temps à l'écran est limité. Dans un premier temps, nous essayons de résoudre le problème suivant : comment sélectionner des attributs d'un ensemble de données pourvu de plusieurs attributs et rejeter les autres ? Le sous-ensemble d'attributs qui sera ainsi sélectionné permettra d'obtenir une représentation visuelle beaucoup plus adéquate à la tâche de FVD par rapport à

l'ensemble de données entier.

Dans le cadre de nos travaux, nous utilisons la FVD pour construire de façon interactive des arbres de décision à partir d'une représentation matricielle des données. L'arbre de décision est construit par un utilisateur qui utilise ses capacités humaines en perception et ses connaissances du domaine des données. Pour que la représentation matricielle soit utilisable, le nombre d'attributs et d'observations des données doit être réduit. La sélection d'attributs dans un ensemble de données fait l'objet de la section suivante.

6.2.1 Sélection d'attributs significatifs

La sélection d'attributs permet de choisir un sous-ensemble de variables suffisant pour décrire un ensemble de données. C'est un processus permettant d'identifier et de retirer autant que possible les informations redondantes et non utiles de l'ensemble de données. Des techniques performantes [John et al., 1994], [Kira et Rendell, 1992], etc. de sélection de sous-ensembles d'attributs ont été développées afin de faire face à trois types de problèmes posés par les méthodes d'analyse de données :

- la réduction du coût et la complexité des algorithmes d'apprentissage,
- l'amélioration de la précision des modèles de données obtenus par un processus d'apprentissage,
- l'amélioration de l'intelligibilité de ces modèles de données.

Conformément à l'état de l'art relatif à ce sujet, la sélection d'attributs dans un ensemble de données comprend une phase de génération de sous-ensembles d'attributs, une phase d'évaluation des attributs générés avec une fonction d'évaluation et un critère d'arrêt. La fonction d'évaluation de sous-ensembles d'attributs peut être un algorithme d'induction ou une mesure statistique. Cette fonction permet de distinguer deux types d'approches : des approches de type enveloppe [John et al., 1994] et des approches de type filtre [Kira et Rendell, 1992]. Les méthodes existantes de sélection d'attributs peuvent être adaptées pour une utilisation en FVD, mais comme nous le montrons dans la section 6.2.2.1, cette adaptation nécessite de résoudre quelques problèmes au préalable.

6.2.1.1 Procédure de génération des sous-ensembles d'attributs

Le processus de génération de sous-ensembles d'attributs permet d'explorer un espace constitué de différentes combinaisons d'attributs disponibles dans l'ensemble de données. Si on considère N comme étant le nombre d'attributs de l'ensemble de données initial, afin d'obtenir une meilleure précision de l'algorithme de sélection d'attributs, il importe d'opérer une recherche exhaustive dans une combinaison de 2^N sous-ensembles d'attributs.

Lorsque N est grand, l'exploration exhaustive de sous-ensembles constitués de différentes combinaisons d'attributs est fastidieuse voire impossible. Des heuristiques et stratégies ont été définies pour une optimisation de cette étape. La recherche exhaustive peut alors par exemple être remplacée par une recherche de type heuristique ou une recherche de type aléatoire. Concrètement, les méthodes utilisées durant le processus de

génération de sous-ensembles d'attributs peuvent être de type branch and bound, forward selection, backward elimination. Il existe aussi diverses améliorations de ces heuristiques, nous pouvons citer par exemple des méthodes de type séquentiel : on parle alors de sequential forward selection (SFS) ou de sequential backward selection (SBS), etc.

Les approches de génération aléatoires utilisent des probabilités et sélectionnent de manière aléatoire un sous-ensemble d'attributs de l'ensemble de données, des poids sont attribués à chaque variable sélectionnée.

La procédure de génération de sous-ensemble d'attributs aboutit à l'évaluation de ses attributs qui permet de mesurer leur pertinence par rapport au problème à résoudre.

6.2.1.2 Fonction d'évaluation

La fonction d'évaluation de sous-ensembles d'attributs peut être un algorithme d'induction ou une mesure statistique. Cette fonction permet de distinguer deux types d'approches :

- Des approches de type enveloppe [John et al., 1994] : elles utilisent des algorithmes d'induction comme critère d'évaluation des sous-ensembles d'attributs choisis. A chaque itération des approches enveloppes, la qualité du sous-ensemble d'attributs est évaluée avec un algorithme inductif.
- Des approches de type filtre [Kira et Rendell, 1992]; dans cette approche, l'adéquation d'un attribut est obtenue par application de mesures statistiques. Comme exemples de ces mesures statistiques, nous avons : le gain informationnel [Dumais et al., 1998], [Quinlan, 1993] et le coefficient de corrélation [Hall, 2000], etc.

Il importe de souligner que les approches de types enveloppes permettent d'obtenir de meilleurs résultats par rapport aux approches de type filtre. Mais elles sont plus lentes car elles appellent de façon répétitive l'algorithme d'induction [Kotsiantis et Pintelas, 2004].

Ayant présenté la procédure de génération de sous-ensemble d'attributs et la fonction d'évaluation de leur pertinence, le dernier facteur nécessaire à l'exécution de l'algorithme de SA est le critère d'arrêt.

6.2.1.3 Critère d'arrêt

A l'étape d'initialisation de l'algorithme de SA, il existe de nombreux paramètres à définir, parmi lesquels on retrouve les critères d'arrêt. Afin d'aboutir aux meilleurs résultats, il est nécessaire de choisir les meilleurs paramètres sachant que quelques critères d'arrêt des algorithmes de SA peuvent être trouvés dans cette liste :

- un nombre défini d'attributs a été sélectionné,
- un nombre défini d'itérations a été atteint,
- l'addition ou la suppression d'un attribut ne produit pas de meilleurs résultats,
- un sous-ensemble optimum d'attributs a été obtenu compte tenu du critère d'évaluation.

Après avoir présenté les approches utilisées par les algorithmes de sélection de

sous-ensembles d'attributs d'un ensemble de données, nous allons à présent voir quels problèmes sont susceptibles de se poser durant la réutilisation de ces procédures du domaine de l'analyse de données en général au domaine plus spécifique de la FVD.

6.2.2 Problèmes en sélection d'attributs significatifs pour la FVD

Rappelons brièvement qu'il existe une panoplie de méthodes de SA. A la lumière de ce qui précède, nous pouvons conclure qu'il existe plusieurs paramètres à fixer pour un algorithme de ce type (procédure de génération, fonction d'évaluation et critère d'arrêt). Comme nous l'avons mentionné antérieurement, il n'existe pas une méthode qui soit meilleure que toutes les autres dans tous les cas. De plus, lorsque le nombre d'attributs de l'ensemble de données à traiter est élevé, la charge cognitive des utilisateurs est grande, sachant qu'il y en aura qui ne pourront même pas réaliser leurs tâches de FVD dans ce contexte. En effet, un environnement de FVD peut être utilisé par des spécialistes du domaine des données et des spécialistes des méthodes d'analyse de données. Il est important d'observer que les différents utilisateurs peuvent être intéressés suivant les cas par les approches filtres et/ou les approches enveloppes. Dans tous les différents cas de figure, un outil d'aide à la sélection d'un sous-ensemble pertinent d'attributs devrait fournir des résultats assez précis. Mais comment retrouver et paramétrer l'algorithme qui suivant le problème à résoudre renverra les meilleurs sous-ensembles d'attributs ? Ceci tout en sachant que :

la visualisation de plus de quelques dizaines d'attributs rend souvent inutilisable la fouille visuelle de données,

un sous-ensemble optimal d'attributs n'est pas nécessairement unique,

il n'est pas possible de déterminer à priori quelle méthode de sélection de sous-ensemble d'attributs est meilleure que toutes les autres,

la décision d'un comité d'experts est généralement meilleure que la décision d'un seul expert.

Nous avons défini un nouvel algorithme de sélection de sélection d'attributs qui comme nous le verrons combine des décisions pondérées de plusieurs experts (des algorithmes de sélection de sous-ensembles d'attributs). Plus précisément, étant donné deux ou plusieurs méthodes de sélection de sous-ensembles pertinents d'attributs dans un ensemble de données, la question est de savoir comment l'on peut utiliser ces différentes méthodes pour fournir un résultat efficace. Afin de répondre à cette question, nous nous sommes appuyés sur la théorie du consensus qui peut être définie comme un procédé de prise de décision qui utilise entièrement les ressources d'un groupe. Le but est de combiner plusieurs distributions de probabilités en une seule probabilité dans l'optique de résumer des estimations de plusieurs experts. La théorie du consensus trouve l'une de ses justifications dans le fait qu'une décision prise par un groupe d'experts est meilleure en terme d'erreur quadratique moyenne que la décision d'un seul expert. Une telle démarche possède de nombreux avantages. En effet, statistiquement parlant, la consultation de plusieurs expertises lors de la résolution d'un problème est une façon subjective d'accroître la taille de l'échantillon dans une expérience, un ensemble d'experts permet d'obtenir plus d'information qu'un seul expert [Clemen et Winkler, 1999].

L'algorithme proposé « Consensus Theory Based Feature Selection » (CTBFS) reçoit en entrée des sous-ensembles d'attributs issus de chaque expertise. Une procédure intégrée permet de définir de façon visuelle et interactive des poids à affecter aux décisions de chaque expert. CTBFS retourne en sortie un sous-ensemble d'attributs représentant une agrégation des différents sous-ensembles d'attributs reçus en entrée.

Des représentations graphiques de l'ensemble de données constituées uniquement des attributs sélectionnés sont utilisées pour la définition interactive de poids à affecter aux différents experts qui interviennent dans la sélection d'attributs. Il s'agit ici d'un problème d'optimisation de l'affectation de poids aux experts. Dans un problème d'optimisation, il y a un espace des solutions et une fonction d'évaluation afin d'accéder à la qualité de la solution.

Les sections suivantes présentent la théorie du consensus, l'algorithme de sélection d'attributs basé sur cette théorie ainsi que le processus d'assignation visuelle de poids aux experts.

6.3 Théorie du consensus : état de l'art

La théorie du consensus consiste à rechercher un accord parmi des solutions proposées par un groupe d'experts. Cette théorie a été largement utilisée en classification, statistique et en sciences sociales [Barthélemy et al., 1984], [Barthélemy et Janowitz, 1991], [Day et McMorris, 2003] et [Domenach et Leclerc, 2004]. Selon [Clemen et Winkler, 1999], la consultation de plusieurs experts constitue une version subjective d'augmentation de la taille de l'échantillon dans une expérience. Ces experts peuvent en effet fournir plus d'information qu'un seul expert.

Les méthodes basées sur le consensus (combinaison ou agrégation) peuvent être classées en deux catégories : les approches mathématiques et les approches comportementales.

Les approches comportementales tentent de générer un agrément entre les experts par une interaction entre eux [Clemen et Winkler, 1999].

Dans les approches mathématiques [Chen et al., 2005], les opinions individuelles d'experts sont exprimées sous forme de distributions de probabilité subjectives d'un événement incertain et sont combinés par diverses méthodes mathématiques pour former une distribution de probabilité agrégée. Il existe plusieurs modèles de combinaison mathématiques pour la définition d'un consensus [Winkler, 1968], [French, 1985], [Genest et Zidek, 1986] et [Cook, 1991]. Les approches utilisées dans ces combinaisons peuvent être axiomatiques ou bayésiennes. Les approches utilisées de façon usuelles sont basées sur des axiomes, comme le « linear opinion pool (Lin-OP)» ou le « logarithmic opinion pool (Log-OP)».

Le Lin-OP est la somme linéaire des probabilités à posteriori de chaque solution experte. La fonction de décision utilisée à cet effet est la suivante :

$$p(s) = \sum_{i=1}^{n} w_i p_i(s) \qquad (1)$$

Le Log-OP est la moyenne pondérée géométrique des distributions de probabilités individuelles. La fonction de décision dans ce cas est :

$$p(H_j) = \kappa \prod_{i=1}^{N} p_i(s_j)^{w_i} \qquad (3)$$

Dans la fonction de décision présentée ci-dessus, le facteur poids détermine l'influence de chaque expert sur la décision commune. Il existe deux types d'opérateurs d'affectation de poids : les opérateurs contextuels et les opérateurs non contextuels. Nous proposons l'utilisation d'une fonction dépendante du contexte de la décision à prendre pour l'affectation de poids aux différentes expertises. Cette méthode basée sur des représentations graphiques utilise comme nous le verrons les capacités usuelles humaines en perception.

6.4 Algorithme de sélection d'attributs basé sur la théorie du consensus (CTBFS)

Le domaine considéré est constitué d'une valeur limite du nombre d'attributs susceptibles d'être correctement visualisés et traités da façon interactive (C_{cmd}), un ensemble *M* d'experts (algorithmes de sélection d'attributs) $E = \{E_1, \ldots, E_M\}$, chaque expert E_i dispose d'un sous-ensemble de *L* experts (qui représentent les différents critères ou paramètres importants des algorithmes de sélection d'attributs) $E_i = \{e_1, \ldots, e_L\}$. L'utilisation de ce sous-ensemble d'experts (E_i) peut être justifié par le fait que dans un algorithme de sélection d'attributs significatifs, il n'y a aucun critère qui permet d'obtenir de meilleurs résultats que tous les autres. Chaque critère possède des attributs de qualité spécifiques. Il est nécessaire de prendre en considération tous les différents attributs de qualité.

Nous avons aussi un sous-ensemble d'attributs $DS = \{D_1, \ldots, D_L\}$, où $D_i = \{d_1, \ldots, d_K\}$ et *K* est variable. Les sous-ensembles d'attributs sont disponibles selon les paires expert/attributs (e_j, D_j), où $e_j \in E_i$ et $D_j \in DS$.

Chaque attribut sélectionné par un sous expert e_j a une fréquence *freq = 1/nb* d'apparition dans la décision finale, où *nb* est le nombre d'attributs sélectionnés par le sous expert.

Nous définissons un critère de préférence d'un attribut (règle de consensus) comme étant le produit des fréquences d'apparition de l'attribut dans les sous-ensembles d'attributs des experts. Nous utilisons la Log-OP pour le calcul de la préférence d'un attribut *d*.

$$pref(X = d) = \prod_{i=1}^{M} P(X = d \mid D_i = b_i)^{w_i} \qquad (4)$$

où : $P(X = d \mid D_i = b_i)$ est la probabilité à posteriori que l'attribut testé appartienne au sous-ensemble d'attributs à sélectionner lorsque la décision du $m^{ième}$ expert est b_i, w_i est le poids assigné à l'expert.

A cette étape, il est important de revenir sur l'affectation de poids aux différents experts intervenant dans la procédure de sélection des sous-ensembles d'attributs.

6.5 Affectation visuelle de poids pour la prise de décision collective

Une décision collective constitue une décision raisonnable que tous les membres d'un groupe peuvent accepter. Les fonctions de combinaison ou d'agrégation d'opinion nécessitent un facteur poids (voir les formules 1 et 2). Le poids détermine l'influence de chaque expert sur la décision commune. Les poids affectés aux différents experts peuvent être égaux ou on peut rechercher une combinaison optimale de facteurs poids. Toute la difficulté relative à un tel processus est de trouver la stratégie d'optimisation de ces facteurs, sachant que notre objectif est de retrouver des poids qui permettent de réduire de façon significative le nombre d'attributs et si possible d'avoir une meilleure précision en fouille visuelle de données (FVD).

Les poids affectés aux experts doivent à cet effet être proportionnels à leurs décisions. L'idée ici est de donner des poids élevés aux meilleurs experts (en terme de représentation graphique de leur sélection d'attributs).

Nous pensons que la meilleure façon de juger de la qualité de l'expertise proposée par les différentes méthodes de sélection d'attributs serait de procéder à des représentations graphiques de l'ensemble de données à traiter avec uniquement les attributs les plus significatifs choisis par chaque expert. L'idée tout au long de ce processus rappelons-le est de donner un poids faible ou alors de ne pas tenir compte de la décision d'un expert qui aurait choisi un très grand nombre d'attributs (d'où une impossibilité de représenter graphiquement l'ensemble de données).

La méthode d'affectation de poids que nous proposons a pour fondements théoriques un principe de la théorie de Gestalt (une vue d'ensemble est meilleure que la somme des parties) et des propriétés pré-attentives de la vision humaine. En ce qui concerne le principe de Gestalt, en visualisant l'ensemble d'éléments intervenant dans une décision, un processus cognitif se met en place.

Dans notre contexte, l'application du principe de Gestalt en ce qui concerne la visualisation de l'ensemble d'éléments rentrant dans le processus de décision se résume en une représentation graphique multi vue. Chaque vue représente le point de vue de chaque expert, c'est-à-dire la représentation graphique de l'ensemble de données pourvu uniquement des attributs sélectionnés par l'expert, comme l'indique la figure 6.1.

En effet, la technique utilisée pour l'affectation visuelle de poids aux experts intervenant dans le processus de décision collective est une représentation graphique à vues multiples des coordonnées parallèles [Inselberg, 1985]. Chaque vue représente l'ensemble de données à traiter réduit par un des experts. Les coordonnées parallèles permettent de représenter en 2D des données multidimensionnelles sans perte d'information.

Six experts de type filtre ont servi à la sélection des attributs visualisés dans la figure 1. L'expert 1 représente le critère de sélection consistance, l'expert 2 représente l'entropie de Shannon, l'expert 3 quant à lui utilise la distance comme fonction d'évaluation. La

fonction d'évaluation pour l'expert 4 est le gain d'information, le coefficient de Gini pour l'expert 5 et le coefficient de Cramer pour l'expert 6.

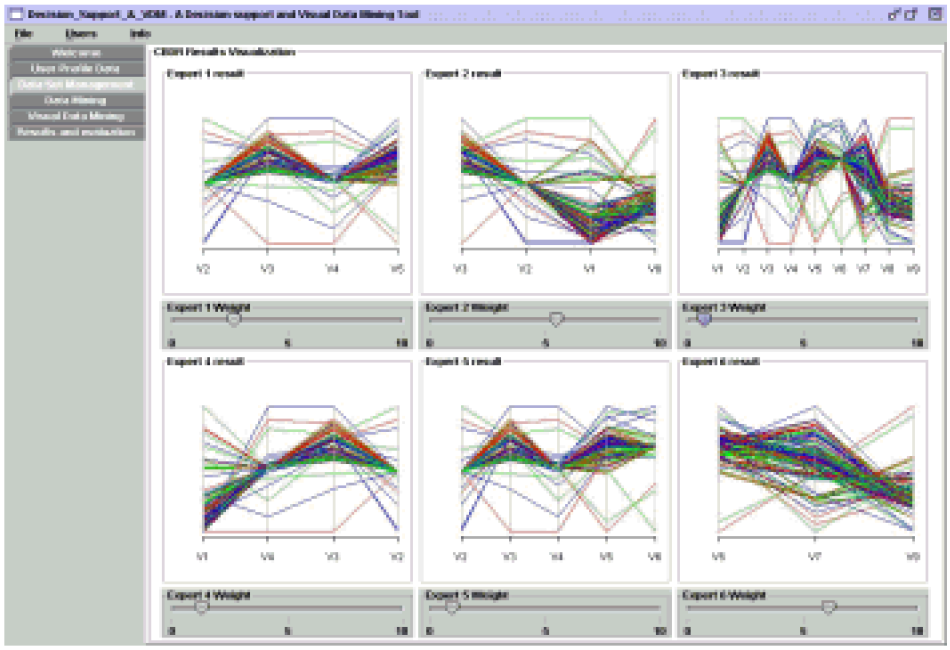


Figure 6.3 Outil d'affectation visuelle de poids aux experts intervenant dans CTBFS.

Il est à noter que les outils usuels d'affectation de poids sont des « boîtes noires ». L'avantage principal de l'approche ainsi proposée tient du fait que l'utilisateur est impliqué et participe dans le processus de prise de décision. Il existe un ensemble de propriétés visuelles qui sont traitées de manière pré attentive très rapidement, avec précision et sans effort particulier. Ce qui permet aux utilisateurs d'affecter des poids convenables aux différents experts.

De plus, les techniques de visualisation permettent d'améliorer la résolution de problèmes. La visualisation permet de découvrir plus aisément des motifs dans les données, de réduire l'espace de recherche d'information par rapport aux méthodes automatiques, de procéder à des opérations perceptuelles d'inférence et d'augmenter la mémoire et les ressources de traitement de l'utilisateur [Dull et Tegarden, 1999], [Card et al., 1999] et [Tegarden, 1999].

6.6 Réduction du nombre d'observations

Certains ensembles de données disposent d'un grand nombre d'attributs et/ou d'observations. Nos investigations en ce qui concerne la réduction des observations dans un ensemble de données consistent à agréger l'information contenue dans cet ensemble de données.

En effet, au lieu de traiter l'ensemble de données avec un grand nombre d'individus, l'idée est d'utiliser juste un échantillon *S* des individus de cet ensemble de données.

Considérons une collection d'observations $\{D_1, \ldots, D_n\}$, à représenter

graphiquement, qui nécessite aussi l'application des procédures de FVD. L'agrégation *S* de cette collection d'observations est une partition $\{S_1, \ldots, S_k\}$ de $\{D_1, \ldots, D_n\}$ et tout S_i est un cluster. Le clustering divise les observations d'un ensemble de données en groupes pour des besoins d'agrégation ou pour une amélioration de la compréhension de ces données. Le clustering qui a été utilisé en compression de données permet de retrouver très efficacement les plus proches voisins d'un point. Pour le clustering, l'ensemble de données initial est séparé en observations de même classe class (ID_i). Ensuite, pour chaque ensemble ID_i, nous appliquons l'algorithme K-means [MacQueen, 1967] afin de retrouver les clusters ou groupes d'éléments disponibles dans ID_i. Typiquement, un algorithme de clustering permet de partitionner *N* entrées $x_1, x_2, , x_N$ en *k* clusters. Les objets regroupés dans chaque groupe résultant sont similaires entre eux et différents des objets des autres groupes. Les algorithmes de clustering essayent de trouver une partition *k* qui maximise une fonction d'objectif en ce qui concerne la mesure de similarité. Par exemple, une fonction d'objectif peut trouver le cluster qui maximise la somme des similarités des objets de la même partition (cluster).

Cette approche a déjà été utilisée dans le cadre d'un prétraitement de grands ensembles de données avec des algorithmes de type « support vector machine (SVM)» par [Do et Poulet, 2005], [Poulet, 2004]. Les auteurs ont testé et validé cette approche sur de grands ensembles de données.

6.7 Expérimentations

Pour les besoins d'expérimentation de la technique proposée qui a été développée sous Windows avec Java et le langage R, nous utilisons un pentium IV, 1.7 GHz. Les ensembles de données que nous utilisons proviennent de l'UCI [Blake et Merz, 1998] et du Kent Ridge Bio-medical Data Set Repository [Jinyan et Huiqing, 2002]. Cette étude de cas dispose de deux tests. Pour les besoins de ces expérimentations, les poids affectés aux différents experts ont pour valeur 1.

L'hypothèse selon laquelle le processus de décision collective que nous avons défini (CTBFS) reflète efficacement les différences dans les jugements des experts sera à vérifier durant ces tests.

6.7.1 Sélection d'attributs

Deux types de recherche sont utilisées durant la phase de génération des sous ensembles d'attributs par les experts : la recherche aléatoire et la recherche séquentielle. Etant donné l'importance des données à traiter, ces heuristiques ont été adoptées afin de réduire le temps nécessaire à l'exploration des différents sous ensembles d'attributs.

6.7.1.1 Premier test

Le domaine considéré dans le cadre de cette première expérimentation est constitué d'un ensemble *M* constitué de 3 experts de type filtre et de 3 experts de type enveloppe *E = {consistence, entropie de Shannon, distance, (LDA, QDA, Kppv)* [Ripley, 1996]*}*, le nombre d'attributs susceptibles d'être traités convenablement est $C_{cmd} = 20$.

Les résultats de l'algorithme proposé (CTBFS) sont comparés à ceux de Las Vegas Filter [Liu et Setiono, 1996], un algorithme de sélection d'attributs de type filtre et StepClass du package KlaR (langage de programmation R), un algorithme de sélection d'attributs de type enveloppe. A cet effet, nous évaluons les performances des ensembles de données pourvus des attributs sélectionnés par ces trois méthodes (LVF, StepClass et CTBFS) avec l'algorithme des k plus proches voisins kppv (implémentation de WEKA [Witten et Eibe, 2005]). Nous avons fixé le paramètre K de l'algorithme des kppv à 1.

Les ensembles de données à traiter dans le cadre de cette première expérimentation sont pourvus de nombreux attributs (colonne 2 du tableau 6.1) et il serait impossible de les visualiser en une seule fois à l'écran quelque soit la méthode de représentation graphique choisie.

Les résultats exposés dans le tableau 6.1 permettent d'observer que l'algorithme CTBFS que nous proposons permet de réduire considérablement le nombre d'attributs des ensembles de données comme le montre les résultats de la colonne 3 du tableau 6.1. La colonne 5 de ce tableau quant à elle fait observer que la précision de l'algorithme de kppv est améliorée pour 4 ensembles de données sur 7. Pour les trois autres ensembles de données, on assiste certes à une perte de précision avec un écart maximal de 16.97% avec un minimum de précision de 68.87% mais l'ensemble de données final peut être visualisé et traité de manière interactive, ce qui n'est pas le cas des ensembles de données initiaux comme nous l'avons souligné.

Tableau 6.1 Comparaison du nombre d'attributs et de la précision obtenus avec l'algorithme des kppv avant et après la sélection d'attributs par l'algorithme CTBFS

Nom	NbAt_Initial	NbAt_CTBFS	Précision_initiale	Précision_CTBFS
Lung-Cancer	57	**4**	37.5%	**75%**
Promoter	59	**9**	**85.84%**	68.87%
Sonar	60	**8**	**86.54%**	71.15%
Arrhythmia	280	**4**	53.44%	**59.96%**
Isolet	618	**14**	**85.57%**	70.24%
ColonTumor	2000	**19**	77.42%	**79.03%**
CentralNervSyst	7129	**20**	56.67%	**60%**

Tableau 6.2 Comparaison du nombre d'attributs et de la précision obtenus avec l'algorithme des kppv avant et après la sélection d'attributs par les algorithmes CTBFS, LVF et Stepclass.

Nom	NbAttr CTBFS	NbAttr LVF	NbAttr Stepclass	CTBFS précision	LVF précision	Stepclass précision
Lung-Cancer	**4**	17	**4**	**75%**	62.5%	71.87%
Promoter	**9**	16	59	68.87%	80.19%	**85.85%**
Sonar	**8**	18	**4**	***71.15%***	82.21%	***71.63%***
Arrhythmia	**4**	109	**4**	**59.96%**	54.65%	**60.84%**
Isolet	**14**	268	**8**	***70.24%***	83%	***57.98%***
ColonTumor	**19**	918	**5**	**79.03%**	77.42%	**79.03%**
CentralNervSyst	**20**	3431	**8**	***60%***	***58.33%***	**71.67%**

On observe sur la colonne 3 du tableau 6.2 que la méthode LVF permet de sélectionner un nombre très important d'attributs, qu'il serait impossible de visualiser (par exemple pour les ensembles de données Arrhythmia, Isolet, ColonTumor et CentralNervSyst). Par rapport à la méthode proposée, la précision obtenue pour ces ensembles de données est équivalente voire supérieure par exemple pour l'ensemble de données Isolet, sachant que l'algorithme CTBFS renvoie au maximum 20 attributs. En ce qui concerne l'algorithme Stepclass, l'ensemble de données Promoter possède aussi un nombre important d'attributs.

En terme de précision, en dehors de l'ensemble de données Promoter pour lequel CTBFS a une précision inférieure à celle de Stepclass et de LVF, la précision obtenue pour les autres ensembles de données avec l'algorithme proposé est au moins égale suivant les cas à celle de LVF ou à celle de Stepclass mais avec un nombre d'attributs qui convient à la fouille visuelle de données.

6.7.1.2 Deuxième test

Dans ce second test, on va s'intéresser au comportement de l'algorithme CTBFS sur des ensembles de données de taille moyenne. Le domaine considéré reste le même. Les résultats obtenus par CTBFS sont aussi comparés à ceux de LVF et Stepclass. Dans cette expérimentation, nous évaluons les performances des ensembles de données pourvus des attributs sélectionnés par LVF, StepClass et CTBFS avec l'algorithme C4.5, implémentation de WEKA.

Tableau 6.3 Comparaison du nombre d'attributs et de la précision obtenus avec l'algorithme C4.5 avant et après la sélection d'attributs par les algorithmes CTBFS, LVF et Stepclass.

	NbAt Initial	NbAt CTBFS	NbAt LVF	NbAt STEP	Précis CTBFS	Précis LVF	Précis STEP
arrhythmia	280	4	**109**	4	**66.15%**	**66.15%**	63.72%
bupa	6	5	2	4	**68.99%**	52.17%	60%
credit_a	15	5	3	4	**86.53%**	73.06%%	74.90%
crx	15	5	3	5	**77.97%**	63.33%	73.48%
glass	9	3	2	2	**61.22%**	47.66%	**62.62%**
hepatitis	19	6	4	16	**80%**	79.35%	**80.65%**
ionosphere	34	4	8	2	**88.89%**	83.76%	79.77%
isolet	618	14	**268**	8	**66.58%**	**73.83%**	58.63%
lung_cancer	57	4	17	4	**71.88%**	62.5%	65.63%
monks	6	4	3	2	**89.52%**	74.19%	72.58%
promoter	59	9	16	**59**	74.53%	68.87%	**79.25%**
sonar	60	8	18	4	**71.15%**	64.90%	65.87%
Voting	16	7	3	8	**94.94%**	88.51%	**96.32%**

Le premier objectif du prétraitement des données pour la FVD rappelons-le est la réduction du nombre d'attributs, autrement, il est impossible de traiter l'ensemble de données. Ensuite, on s'intéresse à la variation de la précision dans les ensembles de

données résultant de ce prétraitement. La colonne 3 du tableau 6.3 montre que ce premier objectif est atteint pour les différents ensembles de données testés. Il est à noter que l'observation des résultats obtenus avec l'algorithme LVF relève un nombre beaucoup plus important d'attributs sélectionnés pour les ensembles de données Arrythmia et Isolet.

En ce qui concerne la précision, on observe un gain avec l'approche proposée sur plusieurs ensembles de données traités (bupa, credit_a, crx, ionosphere, lung_cancer, monks et sonar). Une égalité de précision apparaît entre CTBFS et LVF/Stepclass pour les ensembles de données arrythmia/hepatitis. Etant donné le nombre d'attributs sélectionnés par LVF pour Isolet et Stepclass pour Promoter, nous pouvons conclure que CTBFS permet d'obtenir de meilleurs résultats, le traitement interactif pouvant s'opérer dans les deux cas de figure avec cette méthode.

6.8 Conclusion

Nous avons présenté un algorithme basé sur la théorie du consensus et l'affectation visuelle de poids pour la sélection d'attributs significatifs en FVD. En effet, lorsque le nombre d'attributs et/ou le nombre d'observations d'un ensemble de données est important, il s'avère impossible ou alors pénible de représenter graphiquement l'ensemble de données et d'observer des corrélations dans cet ensemble de données.

La technique présentée permet de définir un nombre maximum d'attributs à sélectionner dans l'ensemble de données à traiter, nombre rendant possible la visualisation de ces données. La première nécessité pour nous est de pouvoir représenter visuellement l'ensemble de données à traiter. Les expérimentations effectuées à cet effet ont été concluantes. Ensuite, nous nous sommes intéressés à la précision des algorithmes C4.5 et kppv sur les ensembles de données à traiter pourvus uniquement des attributs relevés par application de la théorie du consensus. Force a été pour nous de constater que pour plusieurs de ces ensembles de données le taux de précision était amélioré par rapport au taux de précision initial (pour les kppv) et par rapport à LVF et Stepclass pour C4.5. Cette comparaison a été concluante comme l'indique les résultats obtenus en section 5. A la suite de la sélection des attributs, l'utilisation des algorithmes de clustering nous permet de réduire le nombre d'individus des ensembles de données de 50 à 75% avec un maximum de 200 clusters par application de l'algorithme K-Means.

Comme perspectives à ces travaux, nous comptons étendre l'application de la théorie du consensus au choix de la meilleure méthode de visualisation de données pour un ensemble de données à traiter.

Conclusion et perspectives

L'objectif visé par ce travail est une amélioration de la qualité des outils de FVD. Les progrès scientifiques et techniques permettent de libérer les hommes des tâches répétitives et pénibles. Pour faire face à l'augmentation de la masse de données disponible à travers le monde et au désir de découvrir des connaissances enfouies dans ces données, des techniques d'ECD ont vu le jour. Initialement, le souci majeur pour ces techniques a été leur fonctionnalité ou qualité technique. Cependant, la qualité technique à elle seule ne permet pas de déterminer la qualité effective d'un logiciel. Il s'avère important de s'assurer aussi de leur convivialité et de la satisfaction des utilisateurs qui s'en servent, autrement, le temps passé à développer des outils techniquement efficaces serait vain. Nous pensons que la qualité interne (technique) des outils de FVD pour la classification supervisée est une condition nécessaire mais pas suffisante pour assurer leur qualité effective qui implique leur acceptabilité par leurs utilisateurs finaux. En effet, pour mesurer la qualité d'un outil de FVD pour la classification supervisée, on procède par exemple à la validation croisée. On obtient de ce processus le taux de précision du modèle construit qui en réalité illustre la qualité interne du logiciel.

Afin de permettre une étude qualitative un peu plus complète des outils de FVD, nous avons allié qualité technique, qualité et satisfaction d'utilisation, utilisabilité et utilité. Pour ce faire, nous avons trouvé les fondements de notre approche dans les disciplines telles que l'ergonomie des logiciels, le génie logiciel, les interfaces homme machine, etc.

L'ergonomie des logiciels à travers l'analyse des utilisateurs et de la tâche de FVD nous a permis de mieux cerner les utilisateurs ainsi que leurs besoins et la tâche de FVD.

Ceci nous a conduit à la définition du modèle utilisateur. L'idée du modèle utilisateur est d'utiliser les informations relatives au profil et aux préférences des utilisateurs afin de pouvoir les guider tout au long du processus de découverte de connaissances dans les données. Comme montré dans le chapitre 3, le modèle utilisateur n'a pas encore été intégré dans un environnement de FVD.

Après avoir cerné les caractéristiques des utilisateurs, leurs besoins ainsi que la tâche de FVD, notre objectif a été de développer une méthode d'évaluation de ces outils qui puisse servir aux spécialistes (développeurs, analystes de données, etc.) et aux utilisateurs finaux et qui puisse permettre une analyse assez fine des outils de ce type. Nos travaux ont donné naissance à deux méthodes d'analyse et d'évaluation qualitative de ces outils. La première méthode est dédiée à l'inspection experte et la seconde méthode peut servir au diagnostic par tout type d'utilisateur des techniques de FVD. La méthode d'inspection experte est une adaptation des guides de style généraux d'analyse et d'évaluation des interfaces graphiques au domaine spécifique de la FVD. La méthode de diagnostic utilisateur mixe des aspects tels que la qualité technique, le modèle de présentation de l'interface, la qualité des visualisations ou représentations graphiques, la qualité d'utilisation, la qualité des scénarios et permet d'accéder aux points de vue subjectif et/ou objectif des utilisateurs à travers le thème utilisateur. Ces différents thèmes de la méthode d'analyse ou d'évaluation permettent d'opérer un réel diagnostic pour les outils existants et constituent des mises en garde pour le développement de nouveaux outils.

Des études de cas menées avec la méthode de diagnostic proposée nous ont permis de noter des problèmes de qualité qui ne relèvent pas de l'estimation de l'erreur de prédiction. Ce qui confirme notre hypothèse de départ : l'estimation de l'erreur de prédiction est une condition nécessaire, mais pas suffisante pour l'analyse qualitative en FVD.

L'analyse de la situation de travail en FVD ainsi que des études de cas portant sur le diagnostic des systèmes existants de FVD nous ont aussi permis de constater que le processus de FVD nécessitait de nombreux choix. Par exemple le choix de la méthode d'analyse de données à exécuter ou le choix de la méthode de visualisation de données nécessaire à l'exploration de données ou à la confirmation d'hypothèse sur ces données. Nous proposons une technique d'aide au choix de la meilleure méthode d'analyse de données pour la classification supervisée de donnés au chapitre 5.

Toujours dans l'optique de guider les utilisateurs, améliorant ainsi la qualité des outils de FVD, au chapitre 6, nous présentons une nouvelle approche pour le traitement des ensembles de données de très grande taille en FVD. Les limites de l'approche visuelle concernant le nombre d'individus et le nombre de dimensions sont connues de tous. Pour pouvoir traiter des ensembles de données de grande taille, une solution possible est d'effectuer un prétraitement de l'ensemble de données avant d'appliquer l'algorithme interactif de fouille visuelle. La réduction du nombre d'individus est effectuée par l'application d'un algorithme de clustering.

La réduction du nombre de dimensions se fait par la combinaison des résultats d'algorithmes de sélection d'attributs par application de la théorie du consensus (avec une

affectation visuelle des poids).

Nous évaluons les performances de nos nouvelles approches sur des ensembles de données de l'UCI [Blake et Merz, 1998] et du Kent Ridge Bio Medical Dataset Repository [Jinyan et Huiqing, 2002].

Les travaux d'analyse, d'évaluation ou de diagnostic des outils de FVD s'apparentent à ceux de [Grinstein et al, 1997] qui évalue les matrices de scatter plot 2D, 3D, les coordonnées parallèles, etc. du point de vue technique (représentation de données de grande dimension, accès aux données, utilisation de couleur, etc.). Cependant, l'évaluation du point de vue utilisateur n'a pas été traitée dans ces travaux. Très récemment, sont apparus des travaux visant une étude qualitative des outils de FVD. Pour [Fangseu Badjio et Poulet, 2004a], il s'agissait de promouvoir un ensemble de recommandations ergonomiques pour le développement d'outils de FVD de bonne qualité. [Marghescu et al, 2004] ont proposé une méthode d'évaluation de la qualité de visualisation, de la qualité d'interaction et de la qualité d'information d'un environnement de FVD. L'évolution de nos travaux décrits dans [Fangseu Badjio et Poulet, 2004a] a conduit à l'analyse conjointe de l'utilisabilité, de l'utilité et l'acceptabilité des environnements de FVD [Fangseu Badjio et Poulet, 2005a], [Fangseu Badjio et Poulet, 2005b]. Par rapport aux travaux de [Marghescu et al, 2004], dans nos travaux, nous analysons plus finement les outils de FVD. En plus de la qualité de l'interaction, de l'information et de la visualisation étudiés par l'approche qu'ils ont proposée, nous nous intéressons au modèle de présentation de l'interface utilisateur, à la qualité technique de l'outil (système d'exploitation (interopérabilité), l'accès et le traitement des données, l'adaptabilité de la tâche de FVD) à l'aisance de l'utilisateur.

En perspectives à nos travaux, du point de vue analyse ou évaluation qualitative, la technique de diagnostic basée sur des métriques de qualité (chapitre 4) que nous avons proposée se présente à l'heure actuelle sous forme d'un questionnaire. Nous envisageons une automatisation de ce questionnaire avec possibilité de pondération et de support à la décision quant au choix d'un logiciel de FVD. Nous comptons aussi faire valider expérimentalement la méthode d'inspection experte proposée au chapitre 3.

Du point de vue guidage des utilisateurs, nous envisageons premièrement un déploiement du modèle utilisateur proposé au chapitre 3 dans un outil de FVD et une évaluation de sa valeur ajoutée.

Deuxièmement, en ce qui concerne la technique d'aide au choix de la meilleure méthode d'analyse de données pour la classification supervisée proposée au chapitre 5, l'aide aux utilisateurs est sous forme textuelle. Nous envisageons l'emploi d'une méthode visuelle à l'instar des cartes ou des réseaux pour guider les utilisateurs. La carte ou le réseau à développer devra donner une idée de l'impact des choix réalisables sur la suite du processus de FVD. Nous comptons aussi utiliser les résultats obtenus dans ce chapitre pour un meilleur paramétrage des algorithmes de classification supervisée.

Enfin, en ce qui concerne le prétraitement de grands ensembles de données, nous comptons étendre l'application de la théorie du consensus au choix de la meilleure méthode de visualisation de données pour un ensemble de données à traiter.

Références

[Aamodt et Plaza, 1994] Aamodt A., Plaza E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. Artificial Intelligence Communications. IOS Press, vol. 7(1), pp. 39-59, 1994.

[AFNOR, 2003] AFNOR : Ergonomie de l'informatique. Aspects logiciels, matériels et environnementaux, Recueil Normes Informatique, ISBN 2-12-236211-1, 2003.

[Aha et Kibler, 1991]Aha D., Kibler D.: Instance-based learning algorithms, Machine Learning, vol.6, pp.37-66, 1991.

[Ankerst et al, 1999] Ankerst M., Elsen C., Ester M., Kriegel H.-P.: Visual classification: An interactive approach to decision tree construction. In Proc. of the Fifth International Conference on Knowledge Discovery and Data Mining, pp.392–396, 1999.

[Ankerst et Keim, 1996] Ankerst M., Keim D.A., Kriege H-P.: 'Circle Segments': A Technique for Visually Exploring Large Multidimensional Data Sets. In Proc. Of Visualization'96, Hot Topic Session, 1996.

[Ankerst, 2000] Ankerst M.: Visual Data Mining. PhD Thesis, Ludwig Maximilians University of Munich, 2000.

[AS Evaluation, 2005] http://www.lirmm.fr/InfoViz/ASEval/index.php, accédé le 21 septembre 2005.

[Asimov, 1985] Asimov D.: The grand tour: A tool for viewing multidimensional data.

SIAM Journal on Scientific and Statistical Computing, vol. 6(1), pp.128-143, January 1985.

[Atkerson et al., 1997] Atkeson C., Moore A., Schaal S.: Locally weighted learning, Artificial Intelligence Review, vol.11, pp.11-73, 1997.

[Bach, 2004] Bach C. : Elaboration et validation de Critères Ergonomiques pour les Interactions Homme-Environnements Virtuels, Thèse de doctorat, Université de METZ, 2004.[Balbo, 1994] Balbo S. : Evaluation ergonomique des interfaces utilisateur : un pas vers l'automatisation, Thèse préparée au sein du Laboratoire de Génie Informatique - IMAG - Genoble1, 1994.

[Barthélemy et al, 1984] Barthélemy J.-P., Leclerc B., Monjardet B., 1984 : Quelques aspects du consensus en classification, in Dara analysis and informatics (eds. Diday et al.), Amsterdam: Elsevier, pp.307-315, 1984.

[Barthélemy et Janowitz, 1991] Barthélemy J.-P., Janowitz M.F.: A formal theory of consensus, Siam. J. Discr. Math., vol. 4, pp.305-322, 1991.

[Barthet, 1988] Barthet M.F. : Logiciels interactifs et ergonomie, modèles et méthodes de conception, Dunod Informatique, 219 pages, 1988.

[Basili et al, 1994] Basili V., Caldiera G., Rombach D.: The Goal Question Metric Approach. Encyclopedia of Software Engineering, Wiley 1994.

[Bastien et Scapin, 1993] Bastien J.M.C., Scapin D.L. : Critères ergonomiques pour l'évaluation d'interfaces utilisateurs. Rapport technique INRIA n° 156, Juin 1993, INRIA : Le Chesnay, 1993.

[Bastien et Scapin, 1993b] Bastien J.M., Scapin D.L.: Preliminary Findings on the Effectiveness of Ergonomic Criteria for the Evaluation of Human-Computer Interfaces, in Proceedings of INTERCHI'93. pp.187-188, 1993.

[Bertin, 1967] Bertin J. : La Sémiologie graphique Paris, Mouton, 1967.

[Bertin, 1977] Bertin J. : La Graphique et le traitement Graphique de l'information. Flammarion, 1977.

[Bertin, 1981] Bertin J.: Graphics And Graphic Information-Processing. Berlin. Walter de Gruyter, 1981.

[Bisson, 2000] Bisson G. : La similarité : une notion symbolique/numérique. Apprentissage symbolique-numérique (tome 2). Eds Moulet, Brito. Editions CEPADUES. pp.169-201, 2000.

[Blake et Merz, 1998] Blake C., Merz C.: UCI Repository of machine learning databases, [www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, University of California, Department of Information and Computer Science, 1998.

[Boehm et Basili, 2001] Boehm B. et Basili V.: Software Defect Reduction Top 10 List, IEEE Computer, Vol.. 34, No. 1, January 2001.

[Boëhm, 1978] Boëhm B.: Characteristics of software quality, Vol 1 of TRW series on software technology, North-Holland, Amsterdam, Netherlands, 1978.

[Brazdil et al., 2003] Brazdil P., Soares C., Costa J.: Ranking Learning Algorithms Machine Learning: Using IBL and Meta-Learning on Accuracy and Time Results. Machine Learning, vol. 50(3), pp.251-277, 2003.

[Brazdil et Soares, 2000] Brazdil P., Soares C.: A Comparison of Ranking Methods for Classification Algorithm Selection, Machine Learning: ECML 2000, 11th European Conference on Machine Learning, R. López de Mántaras and E. Plaza (Eds.), LNAI 1810, Springer Verlag, pp.63-74, 2000.

[Brodley, 1995] Brodley C.: Recursive Automatic Bias Selection for Classifier Construction. Machine Learning, vol. 20(1-2), pp.63-94, 1995.

[Brown, 1988] Brown C.M.L.: Human-Computer Interface Design Guidelines, Xerox Corporation, 1988.

[Brunk et al., 1997] Brunk C., Kelly J. Kohavi R.: MineSet : an integrated system for data mining, International Conference on Knowledge Discovery and Data Mining (KDD'97), AAAI Press, pp 135-138, 1997.

[Calvary, 2002] Calvary G. : Ingénierie de l'interaction homme-machine : rétrospective et perspectives, Interaction homme-machine et recherche d'information, Traité des Sciences et Techniques de l'Information, Lavoisier, Hermès, pp.19-63, 2002.

[Card et al, 1999] Card S., Mackinlay, J., Scneiderman B.: Readings in Information Visualization: Using Vision to Think, Morgan Kaufman, 1999.

[Card et Mackinlay, 1997] Card S. K., Mackinlay J.:The structure of the information visualization design space. In Proceedings of the IEEE Symposium on Information Visualization 1997 (InfoVis 1997), pp. 92-99, 1997.

[Chambers et al, 1983] Chambers J., Cleveland W., Kleiner B., Tukey P.: Graphical Methods for Data Analysis, Wadsworth, 1983.

[Chandrasekaran et al., 1992] Chandrasekaran B., Johnson T.R., Smith, J.W.: Task-Structure Analysis for Knowledge Modeling. CACM 35, vol. 9, pp.124-137, 1992.

[Chen et al., 2005] Chen Y., Chu C.-H., Mullen T., Pennock D. M.: Information Markets vs. Opinion Pools: An Empirical Comparison , ACM Conference on Electronic Commerce (EC 05), Vancouver, British Columbia, Canada, June 5-8, pp.58-67, 2005.

[Chen et Czerwinski, 2000] Chen C., Czerwinski M.: Empirical evaluation of Information Visualizations: an introduction", International Journal of Human-Computer Studies, Vol. 53, pp.631-635, 2000.

[Chernoff, 1973] Chernoff H.: The use of faces to represent points in k-dimensional space graphically. Journal of the American Statistical Association, vol. 68, pp.361-368, 1973.

[Chi, 2000] Chi Ed H.: A Taxonomy of Visualization Techniques using the Data State Reference Model. In Proceedings of the Symposium on Information Visualization (InfoVis '00), IEEE Press, 2000. Salt Lake City, Utah, pp.69-75, 2000.

[Cleary et Trigg, 1995] Cleary J.G., Trigg L.E.: K*: An Instance- based Learner Using an Entropic Distance Measure, Proceedings of the 12th International Conference on Machine learning, pp.108-114, 1995.

[Clemen et Winkler, 1999] Clemen R. T., Winkler, R. L.: Combining probability distributions from experts in risk analysis. Risk Analysis, vol.19(2), pp.187–203, 1999.

[Cohen, 1995] Cohen W.W.: Fast Effective Rule Induction in Proceedings of the Twelfth International Conference on Machine Learning, pp.115-123, 1995.

[Collier et al., 1999] Collier K., Carey B., Sautter D., Marjaniemi C.: A Methodology for Evaluating and Selecting Data Mining Software. In proc of the 32nd Hawaii International Conference on System Sciences, 1999.

[Constantine, 2001] Constantine L.L.: Design studies 1-3. http://foruse.com/Resources.htm#Articles, 2001, accédé en mars 2005.

[Cook, 1991] Cook R. M.: Experts in Uncertainty: Opinion and Subjective Probability in Science. Oxford University Press, New York, 1991.

[Costabile, 2001] Costabile M.F.: Usability in software life cycle In Handbook of of Software Engineering and Knowledge Engineering, SK Chang Ed., World Scientific, Vol , World Scientific Publ. Company, pp.179-192, 2001.

[Coutaz et al., 1993] Coutaz J., Salber D., Balbo S.: Towards Automatic Evaluation of Multimodal User Interfaces, Amodeus Project document : SM/WP32, 1993.

[Coutaz, 1990] Coutaz J. : Interfaces homme-ordinateur : Conception et Réalisation. Bordas, Paris, 1990.

[Cox et al, 1997] Cox K.C., Eick S.G., Wills G.J., Brachman R.J.: Visual Data Mining: Recognizing Telephone Calling Fraud, Fraud, Data Mining and Knowledge Discovery Vol. 1, pp.225-231, 1997.

[Crampes, 1995] Crampes M. : Composition Multimédia dans un Contexte Narratif. Thèse de doctorat de l'Université de Montpellier II - Sciences et Techniques du Languedoc, 1995.

[Day et McMorris, 2003] Day W.H.E., McMorris F.R.: Axiomatic Consensus Theory in Group Choice and Biomathematics, SIAM, Philadelphia, 2003.

[Detweiler et Omanson, 1996] Detweiler M.C., Omanson R.C.: Ameritech Web Page User Interface Standards and DesignGuidelines, Ameritech Corp., Chicago, 1996. Accessible at http://www.ameritech.com/corporate/testtown/library/standard/web_guidelines/index.html

[Dix et al., 1998] Dix A., Finlay J., Abowd G., Beale R.: Human-Computer Interaction, Second Edition, Prentice Hall, 1998.

[Do et Poulet, 2005] Do T-N., Poulet, F.: Mining Very Large Datasets with SVM and Visualization ", in proc. of ICEIS'05, 7th Int. Conf. on Entreprise Information Systems, Miami, USA, 2005, Vol. 2, pp.127-141, 2005.

[Domenach et Leclerc, 2004] Domenach F., Leclerc B.: " Consensus of classification systems, with Adams' results revisited ". In D. Banks, L. House, F.R. McMorris, P. Arabie, and W. Gaul, editors, Classification, Clustering and Data Mining Applications, Springer, Berlin, pp.417-428, 2004.

[Domingos et Hulten, 2001] Domingos P., Hulten G.: Catching Up with the Data: Research Issues in Mining Data Streams. Proceedings of the Workshop on Data Mining and Knowledge Discovery, ACM SIGMOD/PODS'01, California, USA, 2001.

[Dubuisson, 1990] Dubuisson B. : Diagnostic et reconnaissance des formes, Hermès, 1990.

[Dull et Tegarden, 1999] Dull R.B., Tegarden D.P.: A comparison of three visual representations of complex multidimensional accounting information. Journal of Information Systems. Vol. 13, No. 2 (Fall), pp. 117-131, 1999.

[Dumas, 1999] Dumas J., Redish J.: A Practical Guide to Usability Testing. Intellect Books, Portland, OR, 1999 (revised edition).

[Eibe et Witten, 1998] Eibe F., Witten I. H.: Generating Accurate Rule Sets Without Global Optimization. In Shavlik, J. ed., Machine Learning: Proceedings of the Fifteenth International Conference, San Francisco, CA, Morgan Kaufmann Publishers, 1998.

[Engels et Theusinger, 1998] Engels R., Theusinger C.: Using a Data Metric for Offering Preprocessing Advice in Data-mining Applications. In Proceedings of the Thirteenth European Conference on Artificial Intelligence, pp.430-434 1998.

[Engels, 1996] Engels R.: Planning Tasks for Knowledge Discovery in Databases; Performing Task-Oriented User-Guidance. KDD 1996pp.170-175, 1996.

[Fangseu Badjio et Poulet, 2004a] Fangseu Badjio E., Poulet F. : Guidage des utilisateurs en fouille visuelle de données, in proc. of EGC'04 Workshop on Visualization and Knowledge Discovery, Clermont-Ferrand, pp.13-18, 2004.

[Fangseu Badjio et Poulet, 2005a] Fangseu Badjio E., Poulet F. : Définition des spécificités de la fouille visuelle des données pour une évaluation de l'interaction homme machine, in proc. of 3e Atelier Visualisation et Extraction de Connaissances, EGC'05, Paris, 2005, pp.7-14, 2005.

[Fangseu Badjio et Poulet, 2005b] Fangseu Badjio E., Poulet F.: Dimension Reduction for Visual Data Mining, in proc. of ASMDA'05, the International Symposium on Applied Stochastic Models and Data Analysis, J. Janssen and P. Lenca (Eds), Brest, France, May 2005, pp.266- 275, 2005.

[Fangseu Badjio et Poulet, 2005c] Fangseu Badjio E., Poulet F.: Visual data mining tools: quality metrics definition and application, in proc. of ICEIS'05, the 6th International Conference on Enterprise Information Systems, Miami, Florida, USA, May 2005, pp.98-103, 2005.

[Fangseu Badjio et Poulet, 2005d] Fangseu Badjio E., Poulet F.: Ergonomic Criteria for Visual Data Mining, International Symposium of Visual Data Mining (VDM) of IEEE 9th International Conference on Information Visualization (IV@VDM'05), Poster, London, UK, Jul 2005 (accepted).

[Fangseu Badjio et Poulet, 2005e] Fangseu Badjio E., Poulet F.: Towards usable visual data mining environments, to appear in proc. of HCII'05, the 11th International Conference on Human-Computer Interaction, Las Vegas, Nevada, USA, Jul 2005.

[Fangseu Badjio et Poulet, 2005f] Fangseu Badjio E., Poulet F.: User Guidance: From Theory to Practice, the Case of Visual Data Mining, to appear in proc. of IEEE-ICTAI'05, the 17th IEEE International Conference on Tools with Artificial Intelligence, Hong Kong, China, Nov 2005.

[Fangseu Badjio, 2005a] Fangseu Badjio E.: Quality evaluation of visual data mining tools, in proc. of AC'05, the IADIS International Conference Applied Computing 2005 - 22-25 February 2005, pp.133-138, 2005.

[Farenc, 1997] Farenc C. : ERGOVAL : une méthode de structuration des règles ergonomiques permettant l'évaluation automatique d'interfaces graphique, Thèse de Doctorat de l'Université Toulouse 1, janvier 1997.

[Fayyad et al., 1996] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P.: editors Advances

in Knowledge Discovery and Data Mining. AAAI Press / MIT Press, Menlo Park, CA, 1996.

[Fayyad et Uthurusamy, 2002] Fayyad U., Uthurusamy R.: Evolving Data Mining into Solutions for Insights. Communication of the ACM, 45 (8), pp.28-31, 2002.

[Ferber, 1995] Ferber J. : Les systèmes multi-agents. Vers une intelligence collective. InterEditions, Paris, 1995.

[Fernandes, 1995] Fernandes T.: Global interface design: A guide to designing international user interfaces. Boston, MA: AP Professional, 1995.

[Finyan et Huiqing, 2002] Jinyan L., Huiqing L.: Kent Ridge Bio-medical Data Set Repository. http://sdmc.lit.org.sg/GEDatasets, 2002, accédé le 2 octobre 2005..

[Flanagan, 1954] Flanagan J. C.: The critical incident technique. Psychological Bulletin, vol. 51(4), pp.327-359, 1954.

[French, 1985] French S.: Group consensus probability distributions: a critical survey. Bayesian Statistics, vol. 2, pp.83–202, 1985.

[Freund et Schapire, 1996] Freund Y., Schapire R.E.: Experiments with a new boosting algorithm, in Proceedings of the International Conference on Machine Learning, Morgan Kaufmann, San Francisco, pp 148-156, 1996.

[Furnas, 1986] Furnas G.W.: Generalized fisheye views. In Human Factors in Computing Systems CHI'86 Conference Proceedings, Boston, MA, pp.16-23, 1986.

[Galitz, 1996] Galitz, W. O.: The essential guide to user interface design: An introduction to GUI design principles and techniques. New York: Wiley, 1996.

[Genest et Zidek, 1986] Genest C., Zidek J. V.: Combining probability distributions: A critique and an annotated bibliography. Statistical Science, 1(1):114–148, 1986.

[Ghezzi et al,. 1991] Ghezzi G., Jazayeri M., Mandrioli D.: Fundamentals of software engineering. Prentice-Hall, New Jersey, USA, 1991.

[Google, 2005] GOOGLE: http://www.google.angel-cage.de/html/newsstatistics0704.html, 2005, accédé en mars 2005.

[Gould et Lewis, 1985] Gould J.D., Lewis C. H., "Designing for Usability - Key Principles and What Designers Think," Communications of the ACM, 28, pp. 300-311, 1985.

[Grinstein et al, 1997] Grinstein G.G., Hoffman P., Laskowski S.J, Pickett R.M.: Benchmark Development for the Evaluation of Visualization for Data Mining. In Proceedings of the Workshop: Issues in the Integration of Data Mining and Data Visualization, Newport Beach, California, 1997.

[Hall, 2000] Hall M.: Correlation-based feature selection for discrete and numeric class machine learning. Proceedings of the Seventeenth International Conference on Machine Learning, pp. 359-366, 2000.

[Han et Cercone, 2001] Han J., Cercone N.: "Interactive Construction of Decision Trees" in proc. of PAKDD'2001, LNAI 2035, pp.575-580, 2001.

[Han et Kamber, 2001] Han J., Kamber M.: Data Mining: Concepts and Techniques. Morgan Kaufman, 2001.

[Harinarayan et al., 1996] Harinarayan V., Rajaraman A., Ullman J.: Implementing Data

Cubes Efficiently, Proc. ACM SIGMOD Conf, Montreal, pp.205-216, 1996.

[Hatcheut et al., 2005] Hatchuet A., Masson P. L. & Weil B. (2005 à paraître), Activité de conception, organisation de l'entreprise et innovation, G. Minguet and C. Thuderoz, Eds.

[Healey, 1996] Healey C.G.: Choosing Effective Colours for Data Visualization. In Proceedings of the 7th conference on Visualization '96, pp.263-270, 1996.

[Hoc, 1991] Hoc J-M.: Book Review: ``Handbook of Human-Computer Interaction, '' edited by M. Helander. International Journal of Man-Machine Studies 35(6): 930-931,1991.

[Holte, 1993] Holte R.C.: Very simple classification rules perform well on most commonly used datasets. Machine Learning, Vol. 11, pp.63-91, 1993.

[Howard, 1987] Howard S., Murray D.: A taxonomy of evaluation techniques for HCI, in Proceedings of INTERACT'87, H.J.Bullinger and Shackel (Editors), Elsevier Science Publishers B.V. (North-Holland), IFIP, pp. 453-459, 1987.

[Hû et al., 2001] Hû O., Trigano P., Crozat S. : Une aide à l'évaluation des logiciels Multimédias de formation, publié dans la revue STE - Sciences et Techniques Educatives (ed Hermès), numéro spécial 'Communication Homme/Machine et Apprentissage', Volume 8 n°3, pp.239-274, 2001.

[Huber, 1985] Huber P.J.: Projection Pursuit, The Annals of Statistics, vol. 13 (2) pp. 435-474, 1985.

[IEEE 610.12, 1990] ANSI/IEEE 610.12:1990: Glossary of software engineering terminology, 1990.

[IMS-LIP, 2003] IMS Global Learning Consortium (2003a) IMS Learning Design Best Practice, Version 1.0 Final Specification, 138 p.; IMS Learning Design XML Binding, Version 1.0 Final Specification, 82 p.; IMS Learning Design Information Model, Version 1.0 Final Specification, 87 p.

[Inselberg, 1985] Inselberg A.: The plane with parallel coordinates. The Visual Computer, vol. 1, pp.69-91, 1985.

[Inselberg, 1998] Inselberg A.: Visual Data Mining with Parallel Coordinates, Computational Statistics Vol. 13(1), pp.47-63, 1998.

[ISO 9241-11 1998] ISO (International Organization for Standardization): ISO 13407: Human-Centered Design Process for Interactive Systems, 1998.

[ISO, 1988] Information Processing Systems - Open Systems Interconnection - LOTOS - A Formal Description TechniqueBased on temporal Ordering of Observational Behaviour, 1988.

[ISO, 1992] ISO/WD 9241: Part 11 Ergonomic requirements for Office Work with Visual Displays Units, International Standard Organization, 1992.

[ISO, 1999] ISO: Draft International Standard (DIS) 9241:Requirements for Office Work with Visual Display Terminals, Genève, 1999.

[Jambu, 1999] Jambu M. : Méthodes de base de l'analyse des données, Eyrolles 1999.

[Jézéquel et Meyer, 1997] Jézéquel J-M., Meyer B.: Design by contract: The lessons of ariane. *Computer (IEEE)* , vol. 30(2), pp.129-130, 1997.

[Jinyan et Huiqing, 2002] Jinyan L., Huiqing L.: Kent Ridge Bio-medical Data Set Repository. http://sdmc.lit.org.sg/GEDatasets, 2002, accédé en octobre 2005.

[John et al., 1994] John G.H., Kohavi R., Pfleger K.: Irrelevant Features and the Subset Selection Problem, International Conference on Machine Learning, pp.121-129, 1994.

[John et Langley, 1995] John G.H., Langley P.: Estimating Continuous Distributions in Bayesian Classifiers. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Mateo, pp.338-345, 1995.

[Kalousis et Theoharis, 1999] Kalousis A., Theoharis T.: NOEMON: Design, implementation and performance results of an intelligent assistant for classifier selection. Intelligent Data Analysis, vol. 3 (5), pp.319-337, 1999.

[Kalousis, 2002] Kalousis A.: Algorithm Selection via Meta-Learning. PHD Thesis, Thesis Number:3337. University of Geneve, Department of Computer Science, 2002.

[Kaptelinin, 1995] Kaptelinin V., Kuutti K., Bannon L.: Activity Theory: Basic Concepts and Applications. In Blumenthal et al. (Eds.) Human-Computer Interaction. Lecture Notes in Computer Science. Springer, pp.189-201, 1995.

[Kaptelinin, 1999] Kaptelinin V., Nardi, B.A., Macaulay C.: The Activity Checklist: A Tool For Representing the "Space" of Context, Interactions, pp.27-39, 1999.

[Karat, 1988] Karat, J.: Software evaluation methodologies. In Helander, M. (Ed.). Handbook of human-computer interaction. Amsterdam: Elsevier Science B. V., pp.891-903, 1988.

[Keim et Kriegel, 1994] Keim D.A., Kriegel H.-P.: VisDB: Database Exploration using Multidimensional Visualization, Computer Graphics & Applications Journal, pp.40-49, 1994.

[Keim, 1996] Keim D. A.: Pixel-oriented Visualization Techniques for Exploring Very Large Databases. Journal of Computational and Graphical Statistics, vol. 5(1), pp. 58-77. 1996.

[Kerber et al., 1998] Kerber R., Beck H., Anand T., Smart B.: Actives Templates: Comprehensive Support for Knowledge discovery Process. Intl Conf. on Knowledge Discovery and Data mining, pp.244-248, 1998.

[King et al, 1998] King M.A., Elder IV J.F., Gomolka B., Schmidt E., Summers M., Toop K.: Evaluation of Fourteen Desktop Data Mining Tools. From the 1998 IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA, pp.12-14, 1998.

[Kira et Rendell, 1992] Kira K., Rendell L.A.: A practical approach to feature selection. In Proc. of the Tenth Int'l Conf. on Machine Learning, pp.500-512, 1992.

[Kodratoff, 1996] Kodratoff Y. : Extraction de connaissances à partir de données : un nouveau sujet pour la recherche scientifique. Actes du XIVème Congré INFORSID, Bordeaux, pp.3-22, 1996.

[Köfp et Iglezakis, 2002] Köfp C., Iglezakis I.: Combination of Task Description Strategies and Case Base Properties for Meta-Learning, Proc of the 2nd Intl. Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning (IDDM), pp.65-76, 2002.

[Kohavi, 1995] Kohavi R.: The Power of Decision Tables. In N Lavrac and S Wrobel,

editors, Machine Learning: Proceedings of the Eighth European Conference on Machine Learning ECML95, Lecture Notes in Artificial Intelligence 914, Springer Verlag, pp.174-189, 1995.

[Kohavi, 2000] Kohavi R.: Data Mining and Visualization. Invited talk at the National Academy of Engineering US Frontiers of Engineers, Sept 2000. PDF and Compressed postscript. Available in book form ISBN: 0-309-07319-7, 2000.

[Köpf et Iglezakis, 2002] Köfp C., Iglezakis I.: Combination of Task Description Strategies and Case Base Properties for Meta-Learning, Proc of the 2nd Intl. Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning (IDDM), pages 65-76, 2002

[Kotsiantis et Pintelas, 2004] Kotsiantis S.B., Pintelas P.E.: Hybrid Feature Selection instead of Ensembles of Classifiers in Medical Decision Support, Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems, July 4-9, Perugia - Italy, pp. 269-276, 2004.

[le Cessie, 1992] le Cessie, S. et van Houwelingen J.C. : Ridge Estimators in Logistic Regression. Applied Statistics, Vol. 41(1), pp.191-201, 1992.

[Leont'ev, 1978] Leont'ev A.N.: Activity, Consciousness, Personality. Englewood Cliffs, NJ, Prentice Hall, 1978.

[Lewis, 1990] Lewis C., Polson P.G., Wharton C., Rieman J.: Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. In Chew, J .C., and Schneiderman J. Whiteside eds. CHI'90: Human Factors in Computing Systems. ACM: New York, pp.235-242, 1990.

[Liu et Setiono, 1996] Liu H., Setiono R.: A probabilistic approach to feature selection: a filter solution. In Proc, The 13th International Conference on Machine Learning, pp.319-327, 1996.

[Lohninger, 1994] Lohninger H.: "INSPECT, a program system to visualize and interpret chemical data.", Chemomet. Intell. Lab. Syst. 22 (http://qspr03.tuwien.ac.at/lo/), pp.147-153, 1994.

[Marghescu et al, 2004] Marghescu D., Rajanen M., Back B.: Evaluating the Quality of Use of Visual Data-Mining Tools, in Proc. of 11th European Conference on IT Evaluation, 11-12 November, 2004, Amsterdam, Netherlands, pp. 239-250, 2004.

[Mariage, 2005] Mariage C. : MetroWeb: logiciel de support à l'évaluation de la qualité ergonomique des sites web, Thèse de Doctorat en Sciences de Gestion, UCL, Louvain-la-Neuve, 2005.

[Mayhew, 1992] Mayhew D.J.: Principles and guidelines in software user interface design. Englewood Cliffs, NJ: Prentice Hall, 1992.

[McCall et al, 1977] McCall J.A., Richards P.K., Walters G.F.: Factors in software quality. Vols I-III, Rome Air Development Centre, Italy, 1977.

[Meinadier, 1991] Meinadier J.P. : L'interface utilisateur pour une informatique conviviale, Dunod, 222 p, 1991.

[Metal, 2005] METAL Project, http://www.metal-kdd.org/, accédé en janvier 2005.

[Michie et al, 1994] D. Michie, D.J. Spiegelhalter, C.C. Taylor, (eds.), Machine Learning, Neural and Statistical Classification Ellis Horwood, 1994.

[Monk et al, 1993] Monk A., Wright P., Haber J., Davenport L.: Improving your human-computer interface: A practical technique. Hemel Hempstead, UK: Prentice Hall, 1993.

[Morses et al., 2000] Morse E., Lewis M., Olsen K.A.: Evaluating visualization: using a taxonomic guide", International Journal of Human-Computer Studies, Vol. 53, pp.637-662, 2000.

[Murine et Carpenter, 1984] Murine G., Carpenter C.: Measuring software product quality. Quality progress, Vol 7(5), pp.16-20, 1984.

[Nielsen et Mollich, 1990] Nielson J., Mollich R.: Heuristic evaluation of user interfaces, CHI'90 ACM, New York, pp.249-256, 1990.

[Nielsen et Philipps, 1993b] Nielsen J., Philipps V.L., Estimating the Relative Usability of Two Interfaces: Heuristic, Formal, and Empirical Methods Compared in Proceedings of InterCHI'93, pp.214-221, 1993.

[Nielsen, 1993a] Nielsen J., Usability Engineering, Academic Press Inc., ISBN 0-12-518405-0, 1983.

[Nielsen, 1994] Nielsen J. 1994. Estimating the number of subjects needed for a thinking aloud test. International Journal of Human-Computer Studies, 41 (3), pp.385-397, 1994.

[Norman, 1986] Norman D.A., Draper S.W., User Centered System Design: New Perspectives on Human Factors Interaction, Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, 1986.

[Oudshoorn et al., 1996] Oudshoorn M.J., Widjaja H., Ellershaw S.K. : Aspects and Taxonomy of Program Visualization, World Scientific, Singapore, pp.3-26, 1996.

[Petrak, 2000] Petrak J., Fast Subsampling Performance Estimates for Classification Algorithm Selection In Keller J. and Giraud-Carrier C. Eds. ECML-2000 Workshop Notes on Meta Learning: Building Automatic Advice Strategies for Model Selection and Method Combination, Barcelona, Spain, pp.3–14, 2000.

[Picket et Grinstein, 1988] Pickett R.M., Grinstein G.: Iconographic displays for visualizing multidimensional data. In *Proc. of the 1988 IEEE International Conference on Systems, Man, and Cybernetics* , volume 1, pp.514-519, 1988.

[Pickett, 1970] Pickett R.M.: Visual analyses of texture in the detection and recognition of objects, in B. S. Lipkin and A. Rosenfeld, editors, Picture Processing and Psycho-Pictorics. Academic Press, New York, pp. 298-308, 1970.

[Platt, 1998] Platt J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf, C. Burges, and A. Smola, eds., MIT Press, 1998.

[Pollier, 1991] Pollier A., Evaluation d'une interface par des ergonomes : Diagnostics et Stratégies, Rapport Technique INRIA n°1391, Février 1991.

[Poulet, 2001] Poulet F. : CubeVis : Voir pour mieux comprendre, XXXIIIe Journées de Statistiques, Nantes, 2001.

[Poulet, 2002a] Poulet F., Full-View: A Visual Data-Mining Environment, in International Journal of Image and Graphics, Vol.2, N.1, pp.127-144, 2002.

[Poulet, 2002b] Poulet F.: Cooperation between automatic algorithms, interactive

algorithms and visualization tools for visual data mining. In Proc. of Visual Data Mining workshop, PKDD2002, pp. 67-79, 2002.

[Poulet, 2004] Poulet F.: SVM and Graphical Algorithms: A Cooperative Approach, in proc. of ICDM 2004, pp. 499-502, 2004.

[Preece, 1993] Preece J.: (ed) Guide to Usability: Human Factors in Computing, Addsion-Wesley, Wokingham, England, 1993.

[Quinlan, 1993] Quinlan R.: C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[Rao et Card, 1994] Rao R., Card S.K.: The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus +Context Visualization for Tabular Information, in Proc. of CHI'94, Boston, ACM Press, pp.318-322, 1994.

[Robertson et al, 1991] Robertson G.G., Mackinlay J.D., Card S.K.: Cone Trees: animated 3D visualizations of hierarchical information, in Proc. of the SIGCHI conference on Human factors in computing systems: Reaching through technology, pp.189-194, 1991.

[Saporta, 2005] Saporta G. : Data mining: une nouvelle façon de faire de la statisque. http://cedric.cnam.fr/~saporta/DM.pdf, 2005, accédé en mars 2005.

[Scapin et Bastien, 1997] Scapin D., Bastien C.H., Ergonomic criteria for evaluating the ergonomic quality of interactive systems, Behaviour & Information Technologie 16, pp.220-231, 1997.

[Schneiderman, 1992] Schneiderman B.: Designing the User Interface: Strategies for Effective Human-Computer Interaction. Reading, MA : Addison-Wesley, 1992.

[Schneiderman, 1996] Shneiderman B.: The Eyes Have It: A Task by Data Type Taxonomy For Information Visualizations, in Proc. of IEEE Symposium on Visual Languages, IEEE Service Center, Sep 3-6, pp.336-343, 1996.

[Seewald, 2002] Seewald A.K.: Meta-Learning for Stacked Classification (extended version). In Proceedings of the Second International Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning (IDDM-2002), University of Helsinki, Department of Computer Science, Report B-2002-3, 2002.

[Senach, 1990] Senach B., Evaluation ergonomique des Interfaces Homme-Machine : une revue de la littérature, Rapport de l'INRIA n°1180, Mars 1990.

[Seuren, 1996] Seuren M.: Design and Implementation of an Automatic Event Abstraction Tool. PhD thesis, Waterloo, Ontario, Canada, 1996.

[Smith et Mosier, 1986] Smith S.L., Mosier J.N.: Guidelines for designing user interface software. Massachusetts, USA: The MITRE corporation, 1986.

[Stasko et al., 2000] Stasko J., Catrambone R., Guzdial M., McDonald K.: An evaluation of spacefilling information visualizations for depicting hierarchical structures", International Journal of Human-Computer Studies, Vol. 53, pp.663-694, 2000.

[Suchman, 1987] Suchman, L. A. (1987). Plans and situated actions: The problem of human-machine communications. Cambridge, UK: Cambridge University Press.

[Sun, 1999] Sun: Java Look and Feel Design Guidelines, Microsystems, Inc., Palo Alto, accessible à http://java.sun.com/products/jlf/edl/dg/index.htm, 1999, accédé en

janvier 2005.

[Tegarden, 1999] Tegarden D. P.: Business information visualization. Communications of the AIS. Vol. 1, Article 4 (January), 1999.

[Tricot et al., 2003] Tricot A., Plégat-Soutjis F., Camps J.-F., Amiel A., Lutz G., Morcillo A. (2003). Utilité, utilisabilité, acceptabilité : interpréter les relations entre trois dimensions de l'évaluation des EIAH. Dans Desmoulins, C., Marquet, P., Bouhineau, D. (Dir.), "Environnements Informatiques pour l'Apprentissage Humain 2003". Strasbourg : ATIEF ; INRP. 391-402.
[OAI : oai:archive-edutice.ccsd.cnrs.fr:edutice-00000154_v1] - http://archive-edutice.ccsd.cnrs.fr/edutice-00000154.

[Tufte, 1990] Tufte E.R.: Envisioning Information. Graphic Press, 1990.

[Tufte, 1993] Tufte E.R.: The Visual Display Of Quantitative Information. Graphic Press, 1993.

[Vanderdonckt, 1994] Vanderdonckt J. : Guide ergonomique de la présentation des applications hautement interactives, Presses Universitaires, Namur, 1994.

[Vanderdonckt, 1998] Vanderdonckt J. : Conception ergonomique de pages WEB, Vesale, 1998.

[Vansnick, 1990] Vansnick J.C.: Measurement theory and decision aid, in Bana e Costa (ed.), Readings in Multiple Criteria Decision Aid, Springer-Verlog, Berlin, pp.81-100, 1990.

[Ware et al., 2001] Malcolm Ware M., Eibe F., Holmes G., Hall M., Witten I.H.: Interactive Machine Learning: Letting Users Build Classifiers. International Journal of Human-Computer Studies, Vol.55, No.3, pp.281-292, 2001.

[Wharton et al., 1994] Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994). The cognitive walkthrough method: A practitioner's guide. In Nielsen, J., & Mack, R. L. (Eds.), Usability inspection methods, New York, NY: John Wiley & Sons, pp.105-140, 1994.

[Whiteside et al., 1988] Whiteside J., Bennett J., Holtzblatt K.: Usability engineering: our experience and evolution. In M. Helander, Ed. Handbook of Human Computer Interaction, Amsterdam: Elsevier, pp.791-817, 1988.

[Winkler, 1968] Winkler R. L.: The consensus of subjective probability distributions. Management Science, vol.15(2), pp.61–75, 1968.

[Witten et Eibe, 2005] Witten I.H., Eibe F.: Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[Wolf, 1989] Wolf C.G.: The role of laboratory experiments in HCI: help, hindrance or Ho-hum? (Panel session). Proceedings of CHI+89 conference, Austin, TX, 30 April-4 May 1989. New York: ACM, pp. 265-268, 1989.

[Wolpert et Macready, 1997] D. H. Wolpert, W. G. Macready, No Free Lunch Theorems for Optimization, IEEE Transactions on Evolutionary Computation, vol.1, pp.67–82, 1997.

[Zighed et Rakotomalala, 2003] Zighed A.D., Rakotomalala R. : Extraction de connaissances à partir des données (ECD). Techniques de l'Ingénieur, H3 744, pp.1-26, 2003.

