V4Miner

Un environnement de programmation visuelle pour la fouille de données

Thanh-Nghi Do — Jean-Daniel Fekete

Equipe Aviz, INRIA Futurs/LRI
Bât. 490, Université Paris-Sud
F-91405 ORSAY Cedex, France
{Thanh-Nghi.Do, Jean-Daniel.Fekete}@lri.fr

RÉSUMÉ. *Nous présentons un nouvel environnement de programmation visuelle (V4Miner) pour l'extraction interactive de connaissances à partir de données. La plate-forme utilise simultanément un ensemble de méthodes de visualisation et d'apprentissage automatique permettant de découvrir de manière intuitive, interactive et rétroactive des connaissances. La tâche de fouille est exprimée sous la forme d'un flot graphique de données où une méthode est représentée par un composant graphique de JavaBeans, la communication entre les méthodes se base sur un bus de données. L'utilisateur peut créer et contrôler totalement le processus d'extraction de connaissances sans aucune programmation. Notre plate-forme de programmation visuelle permet d'une part de réduire la complexité d'une tâche de fouille de données et d'autre part d'améliorer la qualité et la compréhensibilité des résultats obtenus. Des premiers résultats sont présentés sur des ensembles de données du Kent Ridge Bio Medical Dataset Repository.*

ABSTRACT. *The visual data flow environment called V4Miner simultaneously uses a set of interactive techniques, visualization methods and machine learning algorithms to discover knowledge in an intuitive, interactive and retroactive way. A data mining task is easily expressed by a graphical visual data flow where a method is represented by a graphic JavaBeans component, the communication between methods is based on a data bus. The user can create and fully control the discovery process without any programming. The visual data flow makes it possible on the one hand to reduce the complexity of an analysis task and on the other hand to improve quality and users' comprehensibility of the obtained results.*

MOTS-CLÉS : *fouille interactive de données, environnement de programmation visuelle, approche anthropocentrée, ensemble des méthodes.*

KEYWORDS: *mining interactive, visual programming, human center, ensemble methods.*

1. Introduction

Le volume de données stockées double actuellement tous les 9 mois (Fayyad et al, 2004) et donc le besoin d'extraction de connaissances dans les grandes bases de données est de plus en plus important. La fouille de données (Fayyad et al, 1996) vise à traiter des ensembles de données pour identifier des connaissances nouvelles, valides, potentiellement utilisables et compréhensibles. Seul l'utilisateur peut déterminer la pertinence des résultats par rapport à ses attentes. Les outils de fouille de données doivent donc être interactifs et anthropocentrés.

(Fayyad et al, 2001) et (Keim, 2002) ont mentionné l'utilisation des méthodes graphiques interactives pour augmenter le rôle de l'utilisateur dans l'analyse exploratoire et la fouille de données. Les méthodes de visualisation peuvent être utilisées pour le pré-traitement de données (par exemple la sélection de données), en fouille de données (utilisation de la capacité humaine en reconnaissance des formes) et en post-traitement (par exemple pour voir les résultats pour améliorer la confiance et la compréhensibilité du modèle).

Dans cet article, nous proposons un environnement de programmation visuelle (V4Miner) permettant à l'utilisateur d'extraire de manière intuitive et interactive des connaissances à partir de données en utilisant simultanément un ensemble de méthodes de visualisation et d'apprentissage automatique. La rétroaction est prise en compte dans le processus de fouille de données parce que l'environnement permet à l'utilisateur de créer, contrôler et interagir avec le processus d'extraction de connaissances sans aucune programmation. Une tâche de fouille est exprimée par un flot visuel de données où une méthode est représentée par un composant en JavaBeans (Sun Microsystems Inc. 1994-2007), la communication entre les méthodes se base sur un bus de données. L'utilisateur peut facilement concevoir lui-même un flot de données et paramétrer visuellement les composants. Ensuite le système peut générer le programme correspondant ou l'utilisateur peut lancer la tâche d'analyse de données à partir de l'environnement graphique. Dans ces deux cas, l'utilisateur peut contrôler de manière interactive et intuitive le processus d'extraction de connaissances, il peut faire des retours en arrière à n'importe quelle étape ou modifier des paramètres en considérant des résultats intermédiaires pour améliorer la qualité des résultats. Nous avons fait l'expérimentation de fouille de données bio-médicales (Jinyan & Huiqing, 2002) et obtenu des résultats intéressants. Une tâche d'analyse de données bio-médicale est difficile parce qu'on traite souvent un grand nombre de dimensions (des milliers) et un petit nombre d'individus (des centaines). A l'aide de V4Miner, cette tâche est rapidement exprimée par un flot visuel de données qui utilise simultanément des méthodes de sélection de dimensions, de classification et de visualisation. On peut interactivement lancer de manière complète ou partielle le flot pour analyser ces données bio-médicales. Les résultats obtenus indiquent que notre plate-forme V4Miner d'une part réduit la complexité d'une tâche de fouille de données et d'autre part permet d'améliorer la qualité et la compréhensibilité des résultats obtenus.

Le paragraphe 2 présente brièvement l'état de l'art sur les plate-formes existantes d'expérimentation pour la fouille de données. Le paragraphe 3 présente ensuite la conception générale, les fonctionnalités et les outils dans l'environnement de programmation visuelle pour la fouille de données. Le paragraphe 4 présente les études de cas des tâches de fouille de données bio-médicales à l'aide de la plate-forme de programmation visuelle avant la conclusion et les travaux futurs dans le paragraphe 5.

2. Etat de l'art

On trouve quelques plate-formes d'expérimentation pour la fouille de données dans le répertoire des logiciels du site KDnuggets (http://www.kdnuggets.com/software/index.html).

Weka (Witten & Frank, 2000) est une bibliothèque gratuite qui implémente en Java des algorithmes d'apprentissage et une plate-forme de programmation visuelle. Il donne à l'utilisateur la possibilité d'ajouter facilement ses propres algorithmes, mais il n'y a pas beaucoup de méthodes de visualisation et de techniques interactives.

HIVE (Ross & Chalmers, 2003) est un environnement de programmation visuelle pour la réduction de dimensions. Il n'y a pas beaucoup de méthodes d'analyse exploratoire de données. Il est difficile de l'étendre parce que HIVE est très spécifique.

YALE (Mierswa et al., 2006) est un environnement gratuit libre d'expérimentation pour l'extraction de connaissances. Il intègre plusieurs méthodes de visualisation et d'apprentissage à l'environnement. YALE ré-utilise la bibliothèque WEKA. Il est souple et extensible. Cependant YALE n'est pas convivial pour la programmation visuelle.

TANAGRA (Rakotomalala, 2005) est une plate-forme gratuite d'expérimentation pour la fouille de données. Il implémente en Delphi des méthodes de fouilles de données issues du domaine de la statistique exploratoire, de l'analyse de données, de l'apprentissage automatique, mais il ne comporte pas beaucoup de méthodes de visualisation et de techniques interactives.

Orange (Demsar et al., 2004) est une plate-forme d'expérimentation pour la fouille de données. Il implémente en C++/Qt/Python des méthodes d'apprentissage automatique et de visualisation.

GeoVista Studio (Takatsuka & Gahegan, 2002) est un environnement de programmation visuelle pour l'analyse en géospatiale. Il est gratuit, libre, souple, extensible et implémente en Java des algorithmes d'apprentissage et des méthodes de visualisation et techniques interactives. GeoVista Studio n'est pas encore convivial pour l'utilisateur.

Notre environnement de programmation visuelle V4Miner permet à l'utilisateur d'extraire de manière intuitive et interactive des connaissances en utilisant simultanément un ensemble de méthodes de visualisation et d'apprentissage automatique. Il est libre, gratuit, souple, général et extensible en Java.

3. Environnement de programmation visuelle

La conception de l'environnement graphique interactif de programmation visuelle V4Miner décrit figure 1 vise à réduire la complexité et à impliquer plus significativement l'utilisateur dans le processus d'extraction de connaissances.

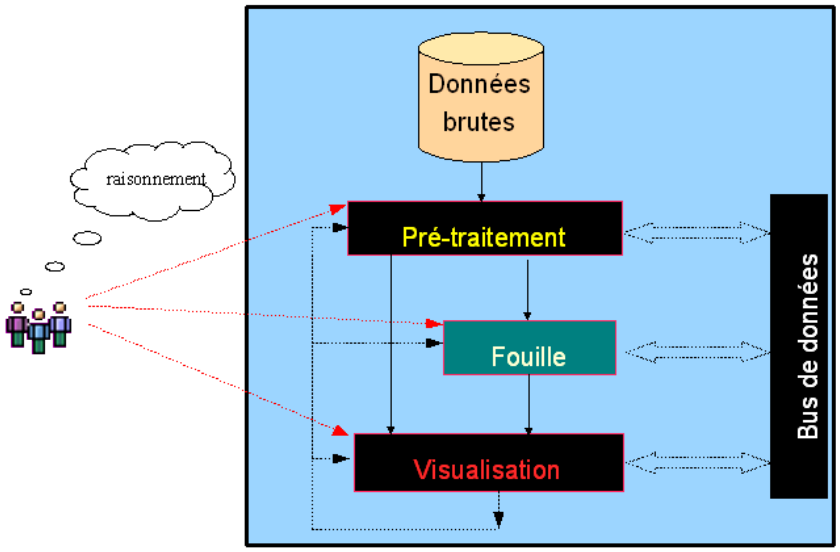

Figure 1. *Conception du V4Miner pour l'extraction de connaissances*

Nous proposons de rassembler un ensemble de méthodes existantes de visualisation et d'apprentissage automatique dans une plate-forme unique permettant de donner à l'utilisateur l'opportunité d'intervenir de différentes manières dans le processus de fouille.

L'environnement graphique interactif V4Miner est basé sur le concept de composant natif de JavaBeans standard et un Beanbuilder JBeanStudio (Takatsuka, 2002-2007). On peut réutiliser un ensemble de méthodes existantes de visualisation et d'apprentissage automatique à moindre coût. Il est souple, général et extensible.

Dans la plate-forme V4Miner, une méthode est encapsulée par un JavaBean et le bus de données est utilisé pour la communication entre les méthodes d'analyse. Aucune connaissance préalable de la programmation n'est requise. L'utilisateur effectue une tâche de fouille en créant lui-même un flot visuel de données. Il a la possibilité de choisir les méthodes à utiliser pour sa tâche et relier les entrées et sorties des méthodes. Tout fonctionne de manière graphique, intuitive et interactive (figure 2). L'utilisateur peut contrôler totalement le processus d'extraction de connaissances. Il peut lancer ensuite de manière complète ou partielle l'exécution des méthodes du processus. La rétroaction est prise en compte, c'est-à-dire qu'il est possible de retourner en arrière à n'importe quelle étape et que l'on peut paramétrer visuellement les méthodes en considérant des résultats intermédiaires pour améliorer la qualité des résultats. Le système peut générer un programme exécutable de Java à partir du flot visuel de données. La plate-forme V4Miner réduit donc la complexité et le coût d'une tâche de fouille de données. La compréhensibilité des résultats peut-être améliorée parce que l'utilisateur participe activement au processus d'extraction de connaissances et il peut utiliser un ensemble de méthodes de visualisation pour interpréter les résultats.

Nous avons intégré dans l'environnement V4Miner plusieurs méthodes de visualisation, des algorithmes d'apprentissage automatique et des méthodes statistiques. Nous avons ré-utilisé des outils existants de la bibliothèque Infovis Toolkit (Fekete, 2004) et la bibliothèque Weka (Witten & Frank, 2000).

L'utilisateur peut exprimer une tâche de fouille de données en utilisant simultanément :

- des méthodes de visualisation comme les coordonnées parallèles (Inselberg, 1985), les matrices de scatterplot 2D, 3D (Carr et al, 1983), le fisheye, l'excentric labeling (Fekete & Plaisant, 1999), la treemap (Johnson, 1992), l'icicle-tree, le node-link-tree, le node-link-graph ou la matrix-graph (Henry & Fekete, 2006),

- des méthodes de réduction et de sélection de dimensions : l'analyse en composantes principales PCA (Pearson, 1901), les méthodes de noyaux PCA (Schölkopf et al., 1998), l'analyse factorielle discriminante FDA (Fisher, 1936), les méthodes de noyaux FDA (Mika et al., 1999), l'analyse factorielle des correspondances AFC (Greenacre, 1984), le gain informationnel ou la mesure du Khi-2,

- des méthodes d'apprentissage automatique à l'aide de séparateurs à vaste marge SVM (Vapnik, 1995) : SMO (Platt, 1999), LS-SVM (Suykens & Vandewalle, 1999), LSVM (Mangasarian & Musicant, 2001), PSVM (Fung & Mangasarian, 2001), Newton-SVM (Mangasarian, 2001), SVM-1 (Fung & Mangasarian, 2002), Reduced SVM (Lee & Mangasarian, 2000) et d'autres méthodes de noyaux (Cristianini & Shawe-Taylor, 2000),

- des arbres de décision : C4.5 (Quinlan, 1993), CART (Breiman et al., 1984) et les forêts aléatoires RF (Breiman, 2001),

- des algorithmes de clustering : les k moyennes KM (McQueen, 1967), EM (Dempster, 1977) et OPTICS (Ankerst et al., 1999),

- des méthodes de régression : la régression logistique LR (Bliss, 1934) et l'arbre de régression logistique LMT (Landwehr et al., 2003),

- des méthodes d'ensemble : le Bagging (Breiman, 1996) et le Boosting (Freund & Schapire, 1996)

- d'autres méthodes : les plus proches voisins kNN (Fix & Hodges, 1952), le Naive Bayes (Good, 1965), Apriori (Agrawal et al., 1993), la visualisation de la marge (Frank & Hall, 2003), les courbes ROC ou les réseaux de neurones.

Bien entendu il serait aussi intéressant d'étudier les possibilités de mixage des différentes approches de visualisation et d'apprentissage en espérant améliorer ainsi le processus et ou les résultats. L'utilisateur peut bénéficier des avantages de chaque méthode pour améliorer les performances de l'utilisation d'une seule méthode. Par exemple, les méthodes automatiques peuvent traiter de grands ensembles de données avec de bons taux de précision et des méthodes de visualisation sont simples et compréhensibles. L'approche coopérative permet de bénéficier des avantages de chaque méthode pour obtenir des résultats compréhensibles et de qualité.

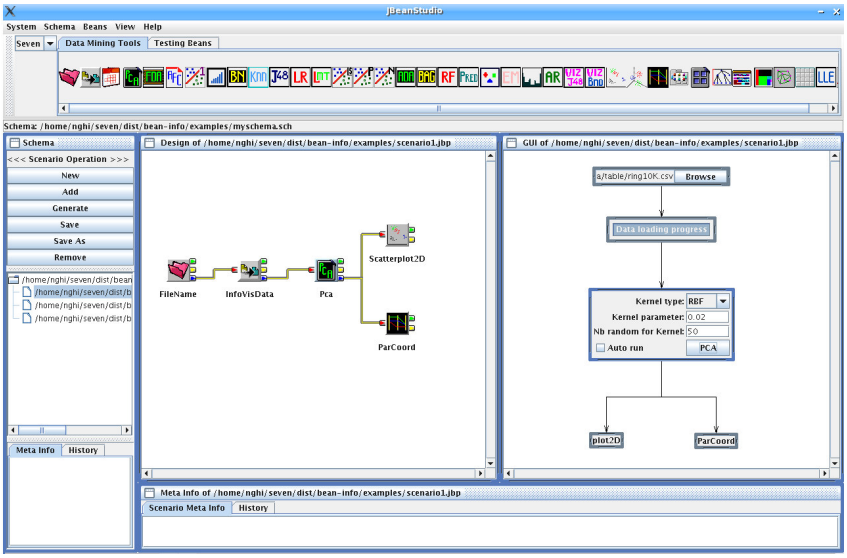

Figure 2. *Tâche de réduction de dimensions des données avec l'algorithme PCA et une fonction de noyau RBF à l'aide de l'environnement V4Miner*

Par exemple, nous considérons une tâche de réduction de dimensions à l'aide de l'environnement V4Miner. L'utilisateur crée facilement lui-même un flot visuel avec les composants du chargement de données FileName, InfoVisData, de l'algorithme d'analyse en composantes principales Pca et les visualisation des ensembles de données sur les projections avec Scatterplot2D et ParCoord comme sur l'exemple de la figure 2. L'utilisateur peut paramétrer les méthodes. S'il souhaite lancer de manière complète le processus il doit alors cocher "Auto run". Dans l'exemple sur la figure 3, l'utilisateur peut interagir avec les Scatterplot-2D pour l'analyse exploratoire des données RingNorm (Delve, 1996) avec 10000 individus en 20 dimensions et 2 classes, puis projette les données et finalement visualise l'ensemble de données avec le Scatterplot-2D. L'utilisateur peut retourner à l'étape de l'algorithme PCA pour modifier les paramètres permattant d'améliorer la qualité du résultat. On voit la visualisation avec le Scatterplot-2D sur les deux premiers axes obtenus par l'algorithme PCA et une fonction de noyau RBF.

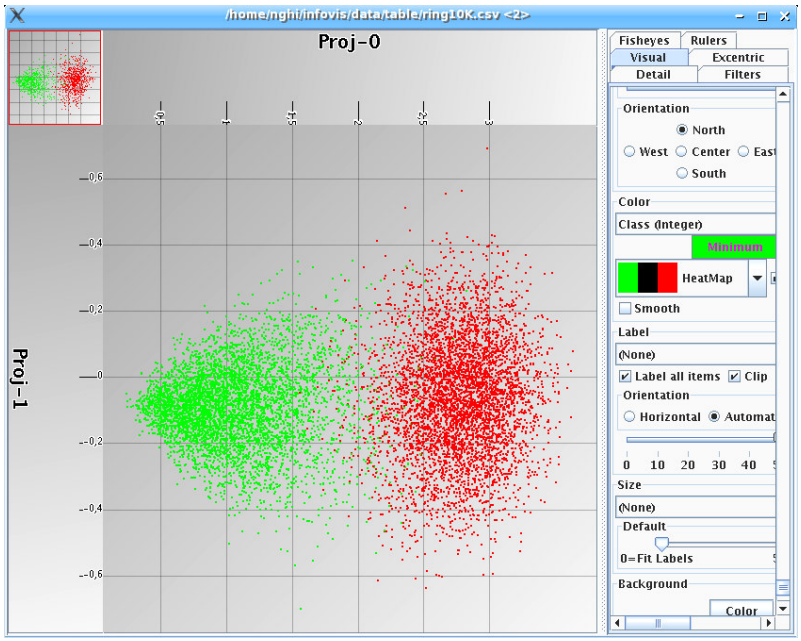

Figure 3. *Visualisation des deux premières projections des données RingNorm avec le scatterplot-2D*

4. Fouille de données bio-médicales à l'aide du V4Miner

Nous étudions la classification de données bio-médicales (Jinyan & Huiqing, 2002) décrites dans le tableau 1.

	Classes	Individus	Dimensions	Protocole de test
Colon Tumor	2	62	2 000	leave-1-out
Ovarian Cancer	2	253	15 154	10-fold
Lung Cancer	2	181	12 533	32 Trn – 149 Tst
Leukemia	2	72	7 129	38 Trn – 34 Tst

Tableau 1. *Description des ensembles de données bio-médicales*

Cette tâche est difficile parce qu'on traite souvent un grand nombre de dimensions (des milliers) et un petit nombre d'individus (des centaines). Donc, la première étape est de sélectionner un sous-ensemble de dimensions à l'aide de l'algorithme SVM-1 pour que les coordonnées parallèles, les scatterplot-2D et l'algorithme d'arbre de décision C4.5 puissent traiter efficacement les données dans le sous-ensemble de dimensions obtenu. L'utilisateur l'effectue facilement par l'intermédiaire du flot visuel à l'aide de l'environnement V4Miner comme représenté sur la figure 4.

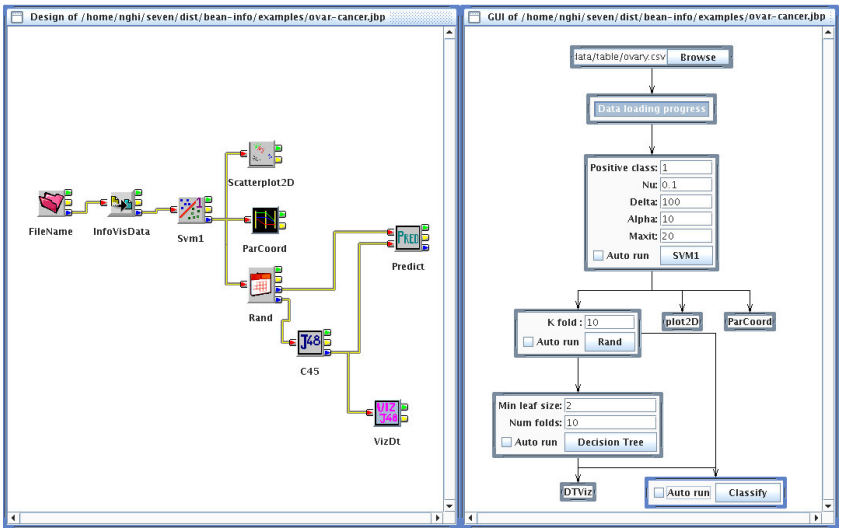

Figure 4. *Tâche de la fouille des données bio-médicales avec l'algorithme de SVM-1, C4.5 et les méthodes de visualisation scatterplot-2D, les coordonnées parallèles*

Pour la classification des données Ovarian Cancer ayant 253 individus en 15154 dimensions et 2 classes, le déroulement du processus est de charger d'abord l'ensemble de données, puis sélectionner un sous-ensemble de dimensions avec l'algorithme SVM-1 et finalement visualiser l'ensemble de données avec les Scatterplot-2D, les coordonnées parallèles ou classifier avec l'algorithme d'arbre de décision C4.5. L'algorithme SVM-1 ne garde que 9 dimensions importantes et la visualisation est ensuite possible (figures 5 et 6).

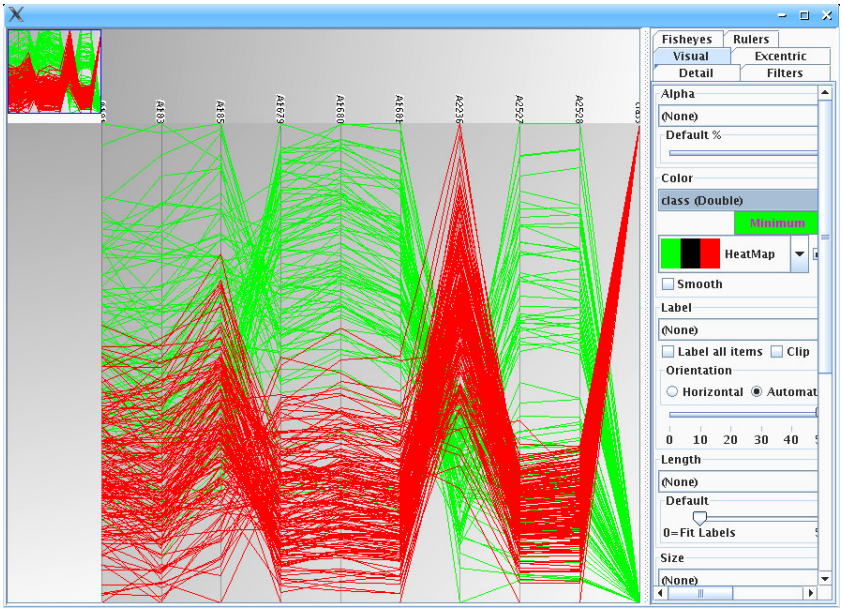

Figure 5. *Visualisation des données Ovarian Cancer avec les coordonnées parallèles*

Il est intéressant de remarquer que l'approche coopérative entre la visualisation et l'apprentissage automatique permet d'obtenir des résultats qu'il serait impossible d'obtenir avec un seul outil. Le SVM-1 sélectionne un sous-ensemble de dimensions significatives à partir du très grand nombre de dimensions, puis la visualisation sert à interpréter les résultats obtenus. Les visualisations des coordonnées parallèles et des Scatterplot-2D permettent à l'utilisateur de mettre en évidence les dimensions importantes (par exemple les dimensions 181 et 1679 dans la figure 6). L'algorithme d'arbre de décision C4.5 peut aussi classifier efficacement ces données dans le sous-ensemble de dimensions. L'arbre résultant contient 7 nœuds et est représenté sur la figure 7.

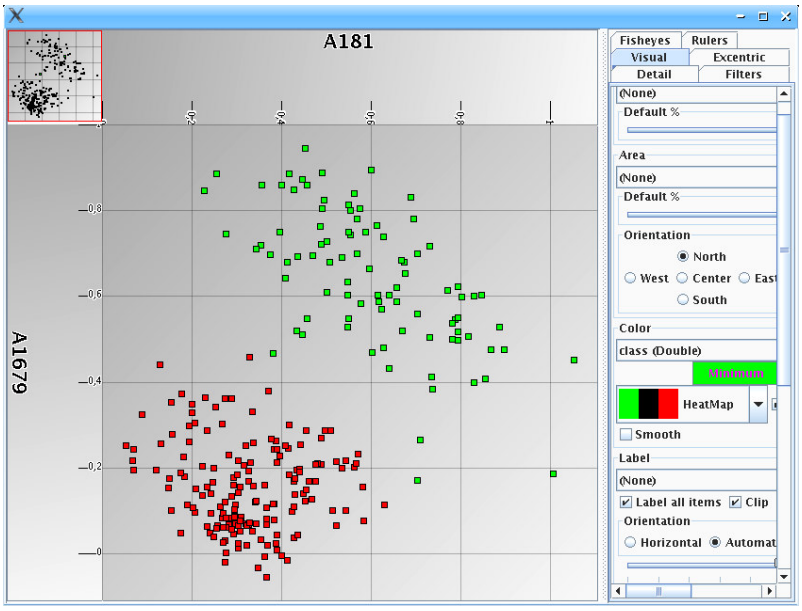

Figure 6. *Visualisation des données Ovary Cancer avec le scatterplot-2D*

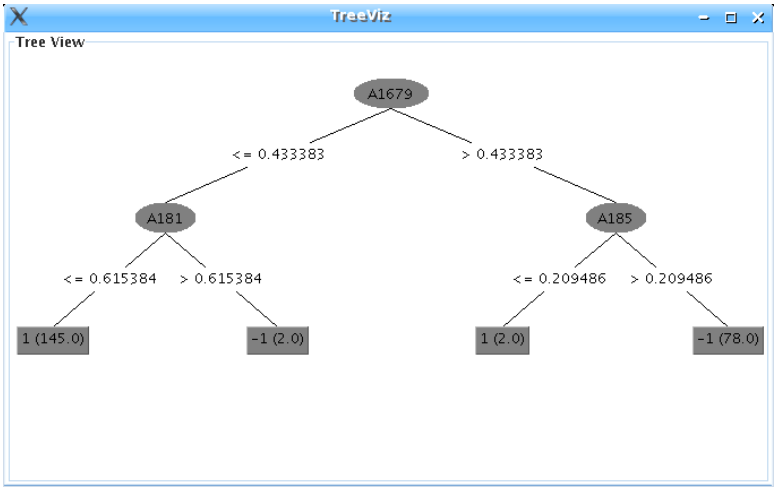

Figure 7. *Arbre de décision pour la classification des données Ovary Cancer*

A partir du petit arbre de décision obtenu, on extrait 4 règles d'induction sous la forme si-alors qui sont facilement compréhensibles par tout utilisateur.

1. SI ((A1679 <= 0,43383) & (A181 <= 0,615384)) ALORS classe = Cancer
2. SI ((A1679 <= 0,43383) & (A181 > 0,615384)) ALORS classe = Normal
3. SI ((A1679 > 0,43383) & (A185 <= 0,209486)) ALORS classe = Cancer
4. SI ((A1679 > 0,43383) & (A185 > 0,209486)) ALORS classe = Normal

Nous avons utilisé le même flot visuel pour classifier les autres ensembles de données. Les résultats obtenus après avoir sélectionné les dimensions sont comparés avec ceux obtenus par une classification sur l'ensemble des dimensions par les algorithmes de SVM.

Les résultats concernant le taux de précision sont donnés dans le tableau 2 (les meilleurs résultats sont en caractères gras). On remarque que pour tous les ensembles de données traités sauf un, les résultats sont meilleurs lorsque l'on utilise un sous-ensemble de dimensions plutôt que l'ensemble complet de dimensions.

	SVM dim., erreurs	SVM-1+Vis dim., erreurs	SVM-1+C4.5 dim., erreurs	SVM-1+SVM dim., erreurs
Colon Tumor	2000, 6	8, 4	8, 5	**8, 2**
Ovarian Cancer	15154, 0	**9, 0**	**9, 0**	**9, 0**
Lung Cancer	**12533, 2**	6, 5	6, 5	6, 5
Leukemia	7129, 2	**9, 2**	**9, 2**	**9, 2**

Tableau 2. *Résultats de la classification des données bio-médicales*

Il est intéressant aussi de constater que le nombre de dimensions utilisées est réduit de manière très significative : par exemple sur Colon Tumor, on passe de 2000 à 8 dimensions (soit une diminution d'un facteur 250) mais le taux de précision est amélioré, sur Ovarian Cancer on passe de 15154 à 9 dimensions (soit une diminution d'un facteur 1684) sans perte de précision et sur Leukemia, on passe de 7129 à 9 dimensions (soit une diminution d'un facteur 792) en conservant exactement le même taux de précision. Pour Lung Cancer on passe de 12533 à 6 dimensions (soit une diminution d'un facteur 2089) mais on perd un petit peu d'information (1,34 % de taux de précision) par rapport à l'utilisation de l'ensemble complet de dimensions. On ne présente pas dans le tableau 2 les autres facteurs importants qui sont d'une part la réduction de la complexité de la tâche de fouille de données et d'autre part l'amélioration de la compréhensibilité des résultats obtenus.

5. Conclusion et perspectives

Nous avons présenté un nouvel environnement de programmation visuelle (V4Miner) pour l'extraction interactive de connaissances à partir de données. La plate-forme utilise simultanément un ensemble de méthodes de visualisation et d'apprentissage automatique pour la fouille de données de manière intuitive, interactive et rétroactive. L'environnement permet d'impliquer plus significativement l'utilisateur dans le processus d'extraction de connaissances. Aucune connaissance préalable de la programmation n'est requise. Une tâche de fouille est exprimée par un flot visuel de données où une méthode est représentée par un composant graphique, la communication entre les méthodes se base sur un bus de données. L'utilisateur conçoit lui-même un flot visuel de données en choisissant les méthodes qui lui permettent d'atteindre son but. Tout fonctionne de manière graphique, intuitive et interactive. L'utilisateur peut contrôler totalement le processus d'extraction. Il peut lancer de manière complète ou partielle le flot visuel pour la fouille de données. La rétroaction est prise en compte. L'utilisateur a la possibilité de retourner en arrière à n'importe quelle étape et peut paramétrer visuellement les méthodes pour améliorer la qualité des résultats. Le système peut générer un programme exécutable en Java à partir du flot de données. Notre plate-forme V4Miner d'une part réduit donc la complexité et le coût d'une tâche de fouille de données et d'autre part améliore la qualité et la compréhensibilité des résultats obtenus. Nous avons obtenu des résultats expérimentaux intéressants en fouille de données bio-médicales.

Il serait intéressant de continuer à intégrer d'autres méthodes de fouille de données des librairies Weka, Yale ou des algorithmes d'analyse de données symboliques de SODAS (Diday, 2000) dans l'environnement de programmation visuelle pour la fouille de données.

6. Bibliographie

Agrawal R., Imielinski T., Swami A., « Mining Association Rules between Sets of Items in Large Databases », in *SIGMOD Conference on Management of Data*, 1993, p. 207-216.

Ankerst M., Breunig M-M., Kriegel H-P., Sander J., « OPTICS: Ordering Points To Identify the Clustering Structure », in *SIGMOD Conference on Management of Data*, 1999, p. 49-60.

Bliss C-I., « The Method of Probits », in *Science*, vol. 79, n° 2037, 1934, p. 38-39.

Breiman L., Friedman J., Olshen R., Stone C., *Classification and Regression Trees*, Chapman & Hall, New York, 1984.

Breiman L., « Bagging Predictors », in *Machine Learning*, vol. 24, n° 2, 1996, p. 123-140.

Breiman L., « Random Forests », in *Machine Learning*, vol. 45, n° 1, 2001, p. 5-32.

Cleveland W-S., *Visualizing Data*, Hobart Press, Summit NJ, 1993.

Cristianini N., Shawe-Taylor J., *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.

Demsar J., Zupan B., and Leban G., « Orange: From Experimental Machine Learning to Interactive Data Mining », White Paper, Faculty of Computer and Information Science, University of Ljubljana, 2004.

Delve, « Data for evaluating learning in valid experiments », 1996. http://www.cs.toronto-.edu/~delve.

Dempster A., Laird N., Rubin D., « Maximum likelihood from incomplete data via the EM algorithm », *Journal of the Royal Statistical Society*, Series B, vol. 39, n° 1, 1977, p. 1–38.

Diday E., « Symbolic Data Analysis and the SODAS project : purpose, history, perspective », in *Analysis of Symbolic Data Exploratory methods for extracting statistical information from complex data*, Springer-Verlag, 2000, p. 1-22.

Fayyad U., Piatetsky-Shapiro G., Smyth P., « From Data Mining to Knowledge Discovery in Databases », in *AI Magazine*, vol. 17, n° 3, 1996, p. 37-54.

Fayyad U., Piatetsky-Shapiro G., Uthurusamy R., « Summary from the KDD-03 Panel – Data Mining: The Next 10 Years », in *SIGKDD Explorations*, vol. 5, n° 2, 2004, p. 191-196.

Fayyad U., Grinstein G., Wierse A., *Information Visualization in Data Mining and Knowledge Discovery*, Morgan Kaufmann Publishers, 2001.

Fekete J-D., « The InfoVis Toolkit », in *IEEE InfoVis*, 2004, p. 167-174.

Fekete J-D., Plaisant C., « Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization », in Proceeding of the *SIGCHI conference on Human factors in computing systems*, 1999, p. 512-519.

Fisher R.A., « The Use of Multiple Measurements in Taxonomic Problems », in *Annals of Eugenics*, n° 7, 1936, p. 179-188.

Fix E., Hodges J., « Discriminatoiry Analysis: Small Sample Performance », in Technical Report 21-49-004, USAF School of Aviation Medicine, Randolph Field, USA, 1952.

Frank E., Hall M., « Visualizing Class Probability Estimators », in *PKDD*, *Lecture Notes in Computer Science 2838*, Springer, 2003, p. 168-179.

Freund Y., Schapire R., « Game Theory, On-line Prediction and Boosting », in Proceedings of *the Ninth Annual Conference on Computational Learning Theory*, 1996, p. 325–332.

Fung G., Mangasarian O., « Proximal Support Vector Machine Classifiers », in Proceedings of the 7th *ACM SIGKDD*, International Conference on Knowledge Discovery and Data Mining, San Francisco, 2001, p. 77-86.

Fung G., Mangasarian O., « A Feature Selection Newton Method for Support Vector Machine Classification », Data Mining Institute Technical Report 02-03, Computer Sciences Department, University of Wisconsin, Madison, USA, 2002.

Good I., *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*, MIT Press, 1965.

Greenacre M-J., *Theory and Applications of Correspondence Analysis*, Academic Press, London, 1984.

Henry N., Fekete J-D., « MatrixExplorer: Un système pour l'analyse exploratoire de réseaux sociaux », Proceedings of *IHM*, *ACM Press*, Canada, 2006, p. 67-74.

Inselberg A., « The Plane with Parallel Coordinates », in *Special Issue on the Computational Geometry of The Visual Computer*, vol. 1, n° 2, 1985, p. 69-97.

Jinyan L., Huiqing L., « Kent Ridge Bio-medical Data Set Repository », 2002. http://sdmc-.lit.org.sg/GEDatasets.

Johnson B., « TreeViz: Treemap visualization of hierarchically structured information Demonstration summary appears », in Proceeding of *ACM-CHI'92*, 1992, p. 369-370.

Keim D., « Information Visualization and Visual Data Mining », in *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, n° 1, p. 1-8.

Landwehr N., Hall M., Frank E., « Logistic Model Trees », in *ECML*, *Lecture Notes in Computer Science 2837*, Springer, 2003, p. 241-252.

Lee Y-J., Mangasarian O., « RSVM: Reduced Support Vector Machines », Data Mining Institute Technical Report 00-07, Computer Sciences Department, University of Wisconsin, Madison, USA, 2000.

MacQueen J., « Some methods for classification and analysis of multivariate observations », in Proceedings of *5th Berkeley Symposium on Mathematical Statistics and Probability*, *Berkeley*, *University of California Press*, vol. 1, 1967, p. 281-297.

Mangasarian O., « A Finite Newton Method for Classification Problems », Data Mining Institute Technical Report 01-11, Computer Sciences Department, University of Wisconsin, Madison, USA, 2001.

Mangasarian O., Musicant D., « Lagrangian Support Vector Machines », in *Journal of Machine Learning Research*, vol. 1, 2001, p. 161-177.

Mierswa I., Wurst M., Klinkenberg R., Scholz, M., Euler T., « YALE: Rapid Prototyping for Complex Data Mining Tasks », in Proceedings of *the 12th ACM SIGKDD, International Conference on Knowledge Discovery and Data Mining*, 2006, p. 935-940.

Mika S., Rätsch G., Weston J., Schölkopf B., Müller K-R., « Fisher Discriminant Analysis with Kernels », in *Neural Networks for Signal Processing*, vol. 9, 1999, p. 41-48.

Pearson K., « On Lines and Planes of Closest Fit to Systems of Points in Space », in *Philosophical Magazine*, vol. 2, n° 6, 1901, p. 559–572.

Platt J., « Fast Training of Support Vector Machines Using Sequential Minimal Optimization », in *Advances in Kernel Methods -- Support Vector Learning*, B. Schoelkopf, C. Burges, and A. Smola Eds., 1999, p. 185-208.

Quinlan J., *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.

Rakotomalala R., « TANAGRA : un logiciel gratuit pour l'enseignement et la recherche », *RNTI-E-3*, vol. 2, 2005, p. 697-702.

Ross G., Chalmers M., « A visual workspace for constructing hybrid multidimensional scaling algorithms and coordinating multiple views », in *IEEE InfoVis*, 2003, p. 247-257.

Schölkopf B., Smola A., Müller K-R., « Nonlinear Component Analysis as a Kernel Eigenvalue Problem », in *Neural Computation*, vol. 10, 1998, p. 1299-1319.

Suykens J., Vandewalle J., « Least Squares Support Vector Machines Classifiers », in *Neural Processing Letters*, vol. 9, n° 3, 1999, p. 293-300.

Takatsuka M., « JBeanStudio: A Component-Oriented Visual Software Authoring System for a Problem Solving Environment », 2002-2007. http://jbeanstudio.sourceforge.net.

Takatsuka M., Gahegan M., « GeoVISTA Studio: A Codeless Visual Programming Environment for Geoscientific Data Analysis and Visualization », in *Computers & Geosciences*, vol. 28, n° 10, 2002, p. 1131-1144.

Vapnik V., *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

Witten I-H., Frank E., *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, San Francisco, 2000.