Vis-SVM : approche coopérative en fouille de données

Thanh-Nghi Do, François Poulet

ESIEA – Pôle ECD
38, rue des Docteurs Calmette et Guérin
Parc Universitaire de Laval - Changé
53000 Laval
(dothanh, poulet)@esiea-ouest.fr

Résumé. La compréhension des résultats en sortie d'un algorithme de fouille de données est aussi importante que d'obtenir de bons taux de précision. Malheureusement, les modèles obtenus par les algorithmes de support vector machines ou séparateurs à vaste marge (SVM) fournissent seulement les vecteurs support qui sont utilisés comme une « boîte noire » pour classifier efficacement les données avec de bons taux de précision. Il est donc indispensable d'améliorer la compréhensibilité des modèles de SVM. Cet article présente différentes coopérations entre des méthodes de visualisation et des algorithmes de SVM en fouille de données. En post-traitement d'algorithmes de SVM, nous présentons une approche coopérative graphique interactive pour interpréter des résultats de classification, régression et détection d'individus atypiques. Nous étendons l'approche d'interprétation graphique pour améliorer les résultats obtenus par la classification de SVM. Nous présentons ensuite une approche coopérative permettant d'impliquer plus significativement l'utilisateur dans la tâche de classification à l'aide de SVM. Ce type d'approche présente notamment comme avantage la possibilité d'utiliser les capacités humaines en reconnaissance de formes par le biais de méthodes de visualisation. L'utilisateur a une meilleure compréhension du modèle construit et une meilleure confiance dans ce modèle parce qu'il a participé activement à sa construction. Nous montrons comment l'utilisateur peut utiliser des outils coopératifs pour construire des modèles de SVM. Une étape de prétraitement est également utilisée dans notre outil coopératif pour pouvoir traiter de grands ensembles de données. Nous évaluons les performances de la nouvelle approche coopérative sur les ensembles de données de l'UCI, Delve, Statlog et biomédicales.

1. Introduction

La fouille de données est un domaine récent de l'informatique dont le développement est lié aux masses de données de plus en plus importantes qui sont stockées à l'heure actuelle. D'après [Fayyad et al., 1996], la définition de l'ECD est : « un processus non trivial d'identification de connaissances inconnues, valides, potentiellement exploitables et compréhensibles dans les données ». Ce processus est complexe, il vise à exploiter des techniques venant de différents domaines de recherches (intelligence artificielle, apprentissage automatique, statistique, analyse de données, visualisation d'informations, bases de données) pour l'extraction de connaissances. Parmi elles, on trouve les arbres de

décision, les règles d'association ou les SVM. Nous nous intéressons particulièrement à une classe récente d'algorithmes d'apprentissage : les SVM [Vapnik, 1995]. Ils donnent de bons résultats en comparaison avec ceux obtenus par d'autres méthodes de fouille de données. Ils ont pour objectif de rechercher le meilleur hyperplan *(w,b)* de séparation des données en deux classes. Le plan est représenté par le vecteur de l'ensemble de ses coefficients *w* et le scalaire *b*. La classification d'un nouvel individu *x* est donnée par sa position par rapport à l'hyperplan, c'est-à-dire le signe de $w.x - b$. On peut utiliser différents types de fonctions de noyau (d'autres formes de frontières) comme une fonction polynomiale de degré *d*, sigmoïdale ou RBF (Radial Basis Function). Le lecteur intéressé par les détails peut consulter [Bennett et Campbell, 2000] ou [Cristianini et Shawe-Taylor, 2000] pour une explication complète. En fournissant des outils de classification supervisée, classification non supervisée, régression et détection d'individus atypiques, les SVM sont une méthodologie générale utilisable pour plusieurs problèmes. Les SVM ont montré leur efficacité dans de nombreux domaines d'applications comme la reconnaissance de chiffres manuscrits, la classification de textes ou la bioinformatique [Guyon, 1999]. Mais leurs résultats ne sont pas facilement interprétables, les seules informations fournies sont en général soit les vecteurs support sans aucune autre indication soit les coefficients de l'hyperplan de séparation (et éventuellement le taux de bonne classification). L'utilisateur sait qu'il peut classifier de manière efficace ses données grâce aux vecteurs support (ou les coefficients de l'hyperplan) mais il est par exemple très difficile d'expliquer les résultats. Des méthodes permettant l'interprétation des résultats de SVM sont donc indispensables. Une première méthode de visualisation des résultats de SVM a été proposée par [Caragea et al., 2001]. Elle effectue la projection en 2D des vecteurs support à l'aide d'un algorithme de Grand Tour [Asimov, 1985]. Une seconde méthode a été proposée par [Poulet, 2002b] utilisant une série de projections 2D « scatter-plot matrices » [Cleveland, 1993] montrant à la fois les individus et les intersections du plan de séparation avec les matrices 2D. Une évolution fut ensuite de ne plus représenter les individus à l'aide toujours d'une série de projections 2D mais avec un histogramme des individus classés en fonction de leur distance à l'hyperplan de séparation [Poulet, 2003a].

Cet article présente différentes coopérations entre des méthodes de visualisation et des algorithmes de SVM en fouille de données. D'abord, nous présentons une approche coopérative graphique interactive [Do et Poulet, 2003a], [Do et Poulet, 2004b] pour interpréter les résultats en sortie de classification, régression et détection d'individus atypiques à l'aide d'algorithmes de SVM. Nous étendons l'approche d'interprétation graphique pour améliorer les résultats obtenus par la classification de SVM [Do et Poulet, 2003b]. Nous présentons ensuite une approche coopérative [Do et Poulet, 2004c] à l'aide de méthodes de visualisation et d'algorithmes de SVM permettant d'impliquer plus significativement l'utilisateur dans la tâche de classification. Ce type d'approche présente notamment comme avantage la possibilité d'utiliser les capacités humaines en reconnaissance de formes par le biais de méthodes de visualisation. L'utilisateur a une meilleure compréhension du modèle construit et une meilleure confiance dans le modèle parce qu'il participe activement à sa construction. Nous montrons comment l'utilisateur peut utiliser des outils coopératifs pour construire des modèles de SVM. Une étape de prétraitement [Do et Poulet, 2004d] est également utilisée dans notre outil coopératif pour pouvoir traiter de grands ensembles de données. Nous évaluons les performances de la nouvelle approche coopérative sur les ensembles de données de l'UCI [Blake et Merz, 1998], Statlog [Michie et al., 1994], Delve [Delve, 1996] et biomédicales [Jinyan et Huiqing, 2002].

Le paragraphe 2 présente les méthodes de visualisation de résultats de classification, régression et détection d'individus atypiques à l'aide de SVM. Nous présentons ensuite l'algorithme d'amélioration des résultats obtenus par le SVM dans le paragraphe 3. Une approche coopérative à l'aide de méthodes de visualisation et d'algorithmes de SVM en classification est présentée dans le paragraphe 4. Puis, le paragraphe 5 présente une étape de prétraitement pour pouvoir traiter de grands ensembles de données, avant la conclusion et les extensions futures de ce travail dans le paragraphe 6.

2. Interprétation graphique des résultats de SVM

Les SVM ont illustré leur efficacité en fouille de données. Ils donnent de bons résultats en classification, régression et détection d'individus atypiques. Mais ces résultats ne sont pas facilement compréhensibles, l'utilisateur obtient souvent les seuls vecteurs support ou les coefficients de l'hyperplan sans aucune autre indication. Il peut classifier de manière efficace ses données mais il est très difficile d'expliquer les résultats.

Nous proposons une approche graphique interactive pour interpréter les résultats de SVM en classification, régression et détection d'individus atypiques. La visualisation des résultats de SVM se base sur la visualisation interactive multi-vue pour expliquer les modèles obtenus par les SVM.

2.1. Visualisation interactive multi-vue, linking, brushing

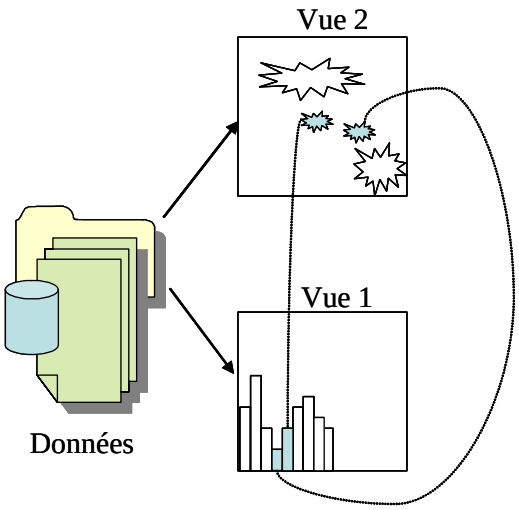

FIG. 1 – *Visualisation de données avec la visualisation multi-vue, linking et brushing*

Les méthodes de visualisation de données sont de plus en plus nombreuses. Cependant, la plupart des techniques sont développées dans différents domaines d'application, les buts à atteindre sont également différents, par exemple la visualisation de données ayant une grande quantité d'exemples ou la visualisation de données avec un grand nombre de dimensions ou la visualisation de données taxonomiques, séries temporelles, etc. Chaque technique présente des atouts et des inconvénients. Il n'est pas toujours facile de choisir une bonne technique

permettant de travailler de manière efficace sur un ensemble de données. Dans tous les cas, l'approche multi-vue qui rassemble un ensemble de méthodes de visualisation permettant d'éviter les limites de l'utilisation d'une seule comme montré sur l'exemple de la figure 1. La même information est présentée dans différentes vues graphiques. L'utilisateur trouve les informations intéressantes à partir des vues les plus appropriées aux données ou aux relations entre données. De plus, les techniques interactives de linking et brushing sont aussi utilisées dans la visualisation multi-vue pour combiner les méthodes de visualisation.

En utilisant la technique interactive de « brushing », on peut se focaliser sur un sous-ensemble de données par sélection, suppression ou filtrage d'un sous-ensemble des données sur une vue graphique. Et ensuite, la technique de « linking » permet de relier les méthodes différentes de visualisation. On peut ainsi voir les comportements des différentes vues sur un sous-ensemble de données sur lequel on s'est focalisé par la méthode interactive de « brushing ».

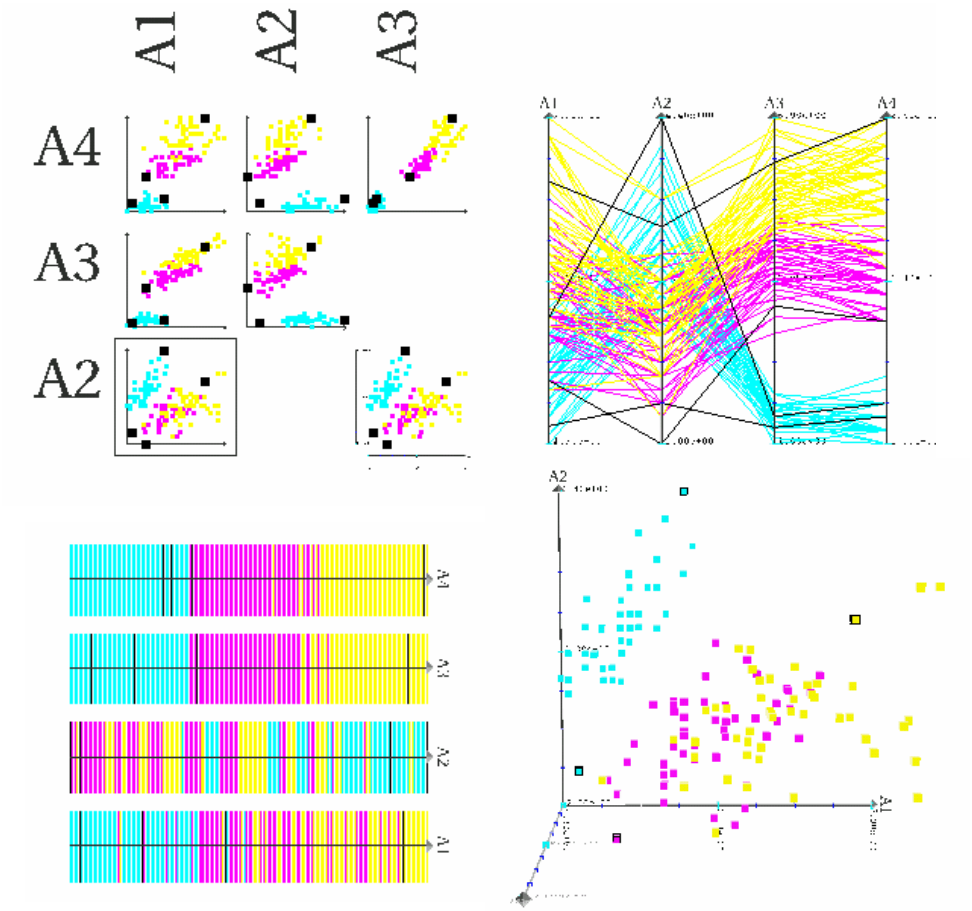

FIG. 2 – *Visualisation des données Iris avec les matrices de scatter-plot en 2D, les coordonnées parallèles, les barres rectangulaires et la vue 3D.*

Par exemple sur la figure 2, nous utilisons quatre méthodes de visualisation : les matrices 2D, les coordonnées parallèles, les barres rectangulaires et la vue 3D pour visualiser les données Iris (*150* individus en *4* dimensions et *3* classes correspondant à *3* couleurs). Les matrices 2D représentent toutes les projections possibles en 2D des données. Dans les coordonnées parallèles, chaque dimension est représentée par un axe vertical, l'individu x_i est représenté par une ligne polygonale dont l'intersection avec l'axe *Dim-j* coïncide avec la valeur de la dimension *j* pour cet individu. Les barres rectangulaires ordonnent les valeurs selon chaque dimension dans une barre, une valeur est représentée par une ligne verticale (ou un pixel). La vue 3D représente la projection des données en trois dimensions.

Ces méthodes de visualisation représentent les données Iris de manières différentes. Les vues sont liées : l'utilisateur peut sélectionner des points dans une vue et ces points sont automatiquement sélectionnés dans les autres vues disponibles. Ainsi, la visualisation multi-vue fournit plus d'informations qu'une seule vue. L'utilisateur peut mettre en évidence des clusters, des tendances ou des corrélations.

L'utilisateur peut choisir les méthodes de visualisation les mieux appropriées à l'analyse de ses données. Nous montrons ensuite comment interpréter les résultats de SVM à l'aide des méthodes de visualisation multi-vue.

2.2. Visualisation des résultats de classification

Dans la classification à l'aide d'algorithmes de SVM, la compréhension de la marge est très importante parce que la marge représente la frontière la plus large possible entre les deux classes. La robustesse des modèles est mesurée par la taille de la marge et les erreurs obtenues par le modèle. Donc, il est intéressant de voir les individus les plus proches de la marge. Ces individus représentent naturellement la frontière de séparation des données en deux classes. L'utilisateur a aussi des informations sur la largeur de la marge.

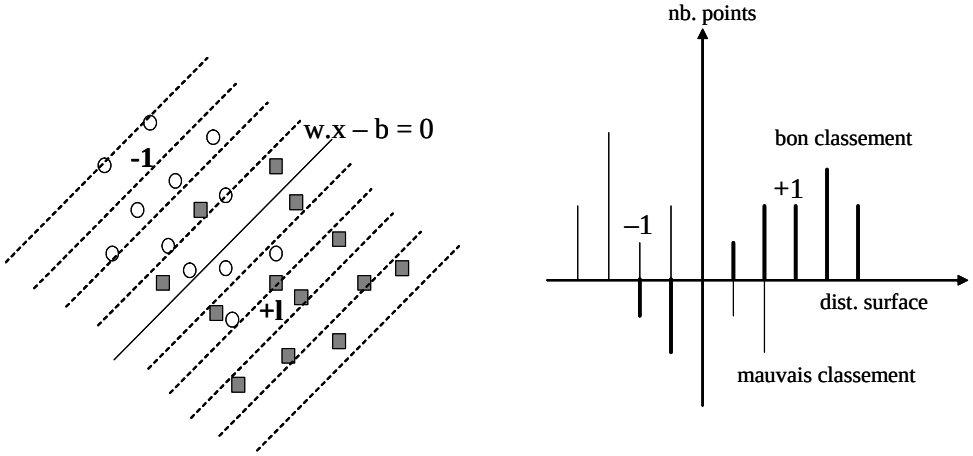

FIG. 3 – *Distribution de données en fonction de la distance à la séparation*

Pour pouvoir visualiser les individus les plus proches de la marge, nous utilisons la visualisation multi-vue en se basant sur la distribution des individus en fonction de leur distance à la surface de séparation. Nous commençons par calculer la distribution des

individus pendant la classification des données, avec en positif, les individus bien classés et en négatif les individus mal classés, la couleur représentant la classe. La figure 3 est un exemple du calcul de distribution des données en fonction de la distance à l'hyperplan de séparation. Cette distribution est ensuite affichée sous la forme d'un histogramme.

Lorsque l'on sélectionne les barres de l'histogramme dans la vue de la distribution (par exemple, les points les plus proches de la frontière de séparation), ces points sont alors automatiquement sélectionnés dans les autres vues. Cette approche donne à l'utilisateur des informations sur la marge. Il peut trouver quelles sont les dimensions intéressantes dans le modèle obtenu et si ces dimensions présentent une frontière claire de séparation des données.

Sur la figure 4, nous avons utilisé le résultat de la classification de la classe *6* contre le reste avec les données Segment. Dans la visualisation de la distribution des individus, on sélectionne les individus les plus proches de la frontière de séparation (les barres de l'histogramme les plus proches de l'origine), ces individus sont automatiquement sélectionnés (les points noirs) dans la vue scatter-plot 2D. Ils représentent la marge de séparation de données. La projection correspondant aux dimensions *2* et *16* présente une frontière claire de séparation des données, ces deux dimensions sont intéressantes dans le modèle pour déterminer l'appartenance ou non à la classe *6*.

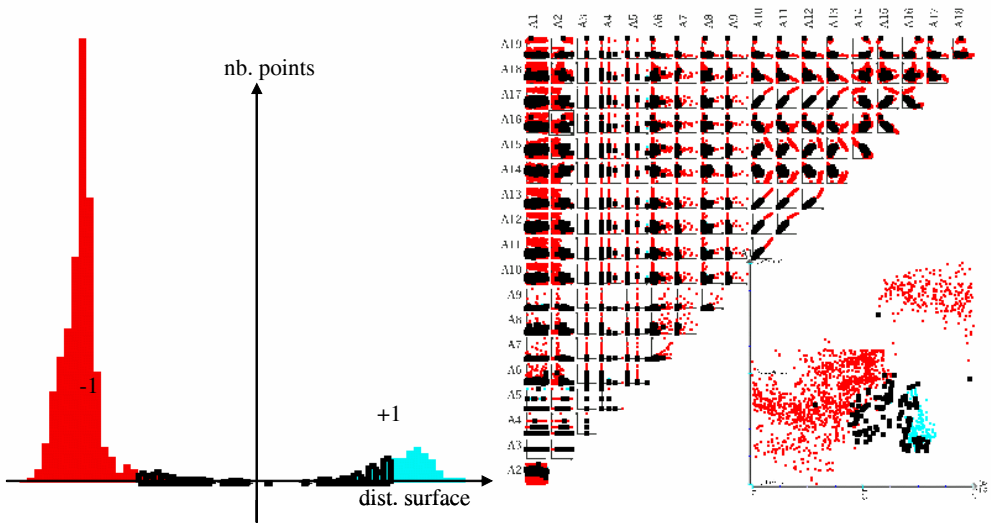

FIG. 4 – *Visualisation du résultat de SVM sur les données Segment (classe 6 contre reste) avec l'histogramme et les matrices de scatter-plot en 2D*

Cette idée est ensuite étendue pour visualiser les résultats de SVM sur les données biomédicales ayant un très grand nombre de dimensions grâce à une étape de sélection de dimensions de l'algorithme de SVM norme-1. Par exemple, les données Lung Cancer contiennent *181* individus en *12 533* dimensions avec *2* classes. L'algorithme de SVM norme-1 ne garde que *9* dimensions pour la classification. La visualisation du résultat est représentée sur la figure 5. Dans ce cadre d'utilisation de la visualisation multi-vue, la vue 3D présente de manière plus claire la frontière de séparation des données que les autres méthodes de visualisation (les matrices 2D et les coordonnées parallèles). Les trois

dimensions (2, 3 et 4) dans la vue 3D sont intéressantes dans le modèle parce qu'elles déterminent l'appartenance ou non à la classe 1.

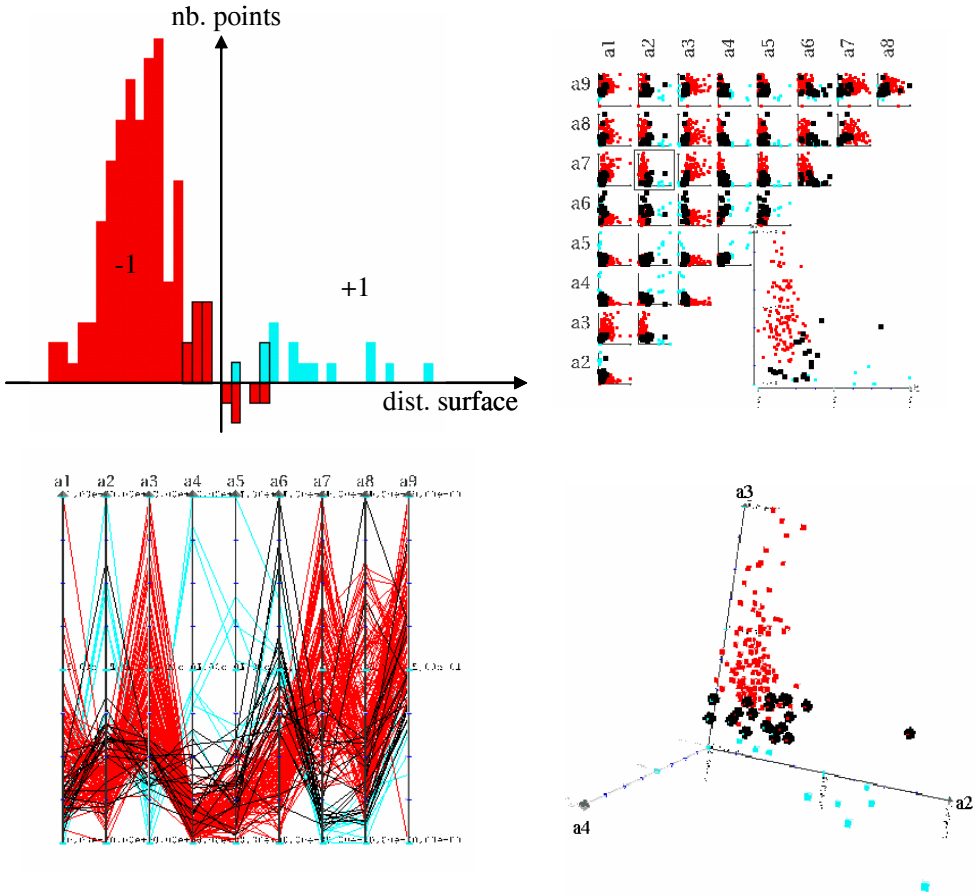

FIG. 5 – *Visualisation du résultat de SVM sur les données Lung Cancer avec l'histogramme, les matrices 2D, les coordonnées parallèles et la vue 3D*

2.3. Visualisation des résultats de régression

Dans la tâche de régression, on recherche une fonction *f* permettant de prédire des valeurs continues en fonction des dimensions des individus. Soit la dérive ε, la régression par SVM cherche une fonction de prédiction *f* telle que les valeurs prédites varient d'au maximum ε autour des valeurs réelles : $-\varepsilon \leq f(x_i) - y_i \leq \varepsilon$.

Nous proposons de visualiser les résultats de la régression à l'aide de SVM pour permettre à l'utilisateur d'évaluer la qualité des résultats. Nous fournissons à l'utilisateur la visualisation des individus les plus éloignés de la fonction de régression. L'utilisateur peut avoir des informations intéressantes sur la trajectoire de la fonction de régression. Il sait comment la fonction de régression suit ses données.

Nous calculons d'abord la distribution de données en fonction de la distance entre les individus et la fonction de la régression. Ensuite, l'histogramme de la distribution de ces distances est lié à d'autres méthodes de visualisation pour interpréter les résultats de régression. L'utilisateur obtient des informations sur la qualité de la régression et sur la fonction de régression, il peut trouver les dimensions importantes dans le modèle obtenu.

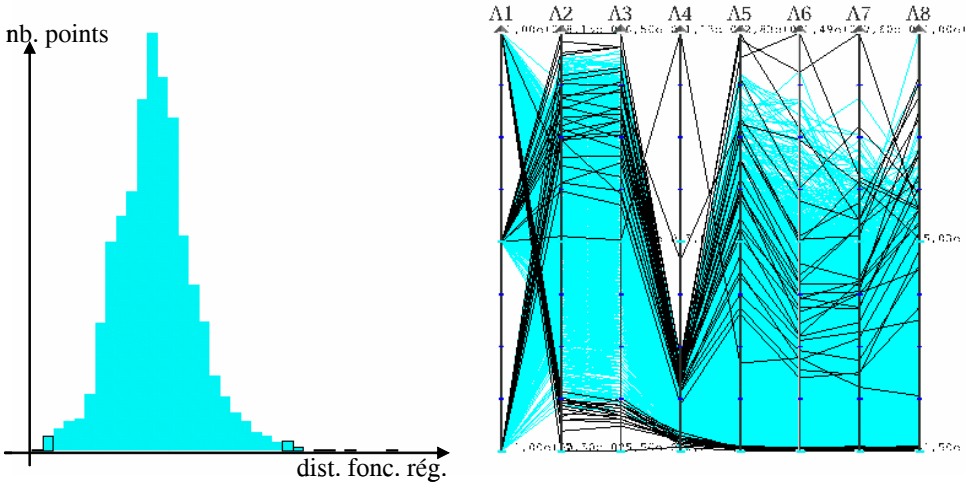

FIG. 6 – *Visualisation du résultat de régression de SVM sur les données Abalone avec l'histogramme et les coordonnées parallèles*

Nous avons effectué une régression linéaire sur l'ensemble de données Abalone de l'UCI [Blake et Merz, 1998] ayant *4 177* individus en *8* dimensions. La visualisation de ce résultat est présentée sur la figure 6 avec l'histogramme et les coordonnées parallèles. Nous sélectionnons les barres les plus éloignées de la fonction de régression (les barres noires) dans l'histogramme, les individus correspondants (les lignes noires) sont sélectionnés dans les coordonnées parallèles. Si ces individus les plus éloignés de la fonction de régression sont atypiques, alors la fonction de régression suit bien les données. L'utilisateur peut avoir des informations sur la qualité de la régression. Avec les coordonnées parallèles, on s'aperçoit que les dimensions *2*, *3*, *4* et *5* sont importantes dans le modèle de régression parce qu'elles présentent clairement les individus les plus éloignés de la fonction de la régression. L'utilisateur a des informations intéressantes sur la fonction de régression.

2.4. Visualisation des résultats de détection d'individus atypiques

La tâche de détection d'individus atypiques ou SVM une classe a pour objectif de rechercher l'hypersphère (de rayon minimal *r* et de centre *o*) qui contient la presque totalité des individus. Un nouvel individu est atypique s'il est à l'extérieur de l'hypersphère. La question se pose ici de comment valider les individus atypiques.

Nous avons utilisé une approche de visualisation multi-vue pour visualiser les individus en fonction de la distance à l'hypersphère obtenue par l'algorithme de SVM. L'utilisateur peut voir et interpréter ou qualifier les individus atypiques.

Nous calculons d'abord la distribution des données en fonction de la distance à l'hypersphère obtenue par l'algorithme. Ensuite, nous affichons cette distribution sous la forme d'un histogramme comme sur l'exemple de la figure 7. Lorsque l'on sélectionne les barres de l'histogramme (les points les plus éloignés de l'hypersphère), ces points sont alors automatiquement sélectionnés dans les autres vues. L'utilisateur peut valider les individus atypiques.

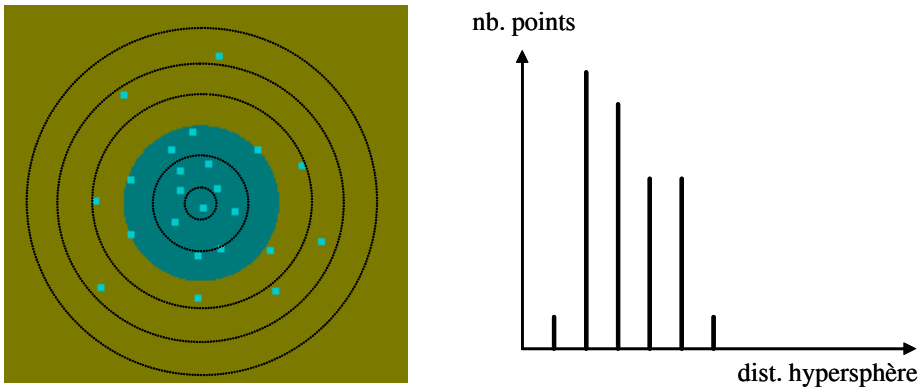

FIG. 7 – *Distribution des données en fonction de la distance à l'hypersphère*

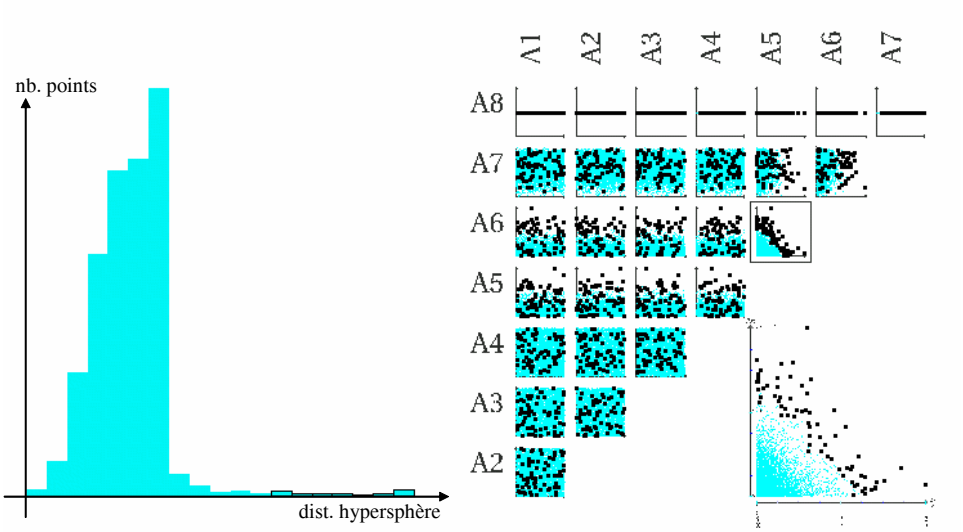

FIG. 8 – *Visualisation des résultats de la détection d'individus atypiques sur les données Bank8FM avec l'histogramme et les matrices 2D*

La figure 8 est un exemple de visualisation des résultats obtenus par l'algorithme de SVM une classe sur les données Bank8FM [Torgo, 2003]. Cet ensemble de données est constitué de *4 499* individus en *8* dimensions (nous avons utilisé un noyau RBF avec $\gamma = 0.5$). Dans la visualisation de la distribution des individus en fonction de la distance à

l'hypersphère, on sélectionne les individus les plus éloignés de l'hypersphère, ces individus sont automatiquement sélectionnés (les points noirs) dans la vue scatter-plot 2D. Ils sont vraiment atypiques sur la projection des deux dimensions *5* et *6*. Donc, ces deux dimensions sont celles qui ont un rôle important pour la détermination d'individus atypiques.

La visualisation multi-vue de l'histogramme et des méthodes de visualisation permet d'interpréter les résultats de SVM. L'utilisateur a une meilleure compréhension des modèles obtenus par les tâches de classification, régression et détection d'individus atypiques.

Nous étendons cette approche coopérative pour améliorer les résultats de classification automatique à l'aide d'algorithme de SVM.

3. Amélioration interactive des résultats de SVM

3.1. Post-traitement graphique interactif des résultats de SVM

Dans la classification multi-classes (plus de *2* classes), les algorithmes de SVM peuvent utiliser l'approche « un contre le reste » ou « un contre un ». Chaque technique présente des atouts et des inconvénients.

Si on a *k* classes, la classification « un contre un » consiste à construire *k(k-1)/2* modèles, où chaque modèle sépare une classe d'une autre classe. Un nouvel individu *x* est étiqueté comme appartenant à la classe *c* s'il est plus éloigné de la surface correspondant à cette classe que de toutes les autres. Cette approche est très coûteuse en temps calcul dans le cas où l'on a beaucoup de classes.

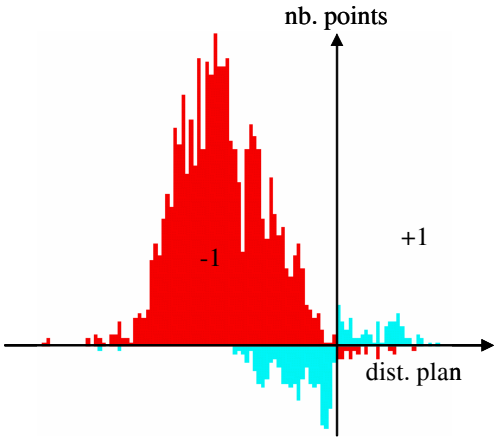

FIG. 9 – *Distribution des données en fonction de la distance au plan*

Dans le cas « un contre le reste », toujours avec *k* classes, la classification consiste à construire *k* modèles, où chaque modèle sépare une classe des *(k-1)* autres. Un nouvel individu *x* est étiqueté comme appartenant à la classe *c* s'il est plus éloigné de la surface correspondant à cette classe que de toutes les autres. Cette approche est très simple mais les résultats sont moins robustes. De plus, on se trouve souvent dans le cas où la taille de la classe *+1* est très petite par rapport à celle de la classe *-1*. Cela peut entraîner que les données

sont entièrement considérées comme appartenant à la classe *-1*, le taux de précision globale n'étant pas affecté. Par exemple, pour séparer la classe *5* des autres dans les données Segment, on obtient *89,03 %* de précision globale avec un noyau linéaire mais le taux d'erreur de la classe *+1* est de *69,59 %*.

Pour remédier à ce problème, il est possible de déplacer la surface obtenue de manière parallèle à elle-même, cela permet d'améliorer le taux de précision en réduisant le taux d'erreurs. Au contraire du même travail réalisé de manière automatique (le réglage des paramètres), notre approche met en œuvre une méthode interactive. L'utilisateur choisit lui-même la réduction du taux d'erreur. A partir de la visualisation de la distribution des données en fonction de la distance à la surface de séparation (figure 9), nous affichons les courbes de taux d'erreur (le taux global et les taux pour les individus des classes *±1*) comme sur l'exemple de la figure 10. Si on déplace la surface sur le minimum du taux global d'erreur, on diminue de *30 %* le taux d'erreur de la classe *+1*, on obtient alors un taux global de précision de *91,43 %* (soit une augmentation de *2,4 %*). Et si l'on répète le même processus pour l'ensemble des classes, le taux global de précision est augmenté de *4,87 %*.

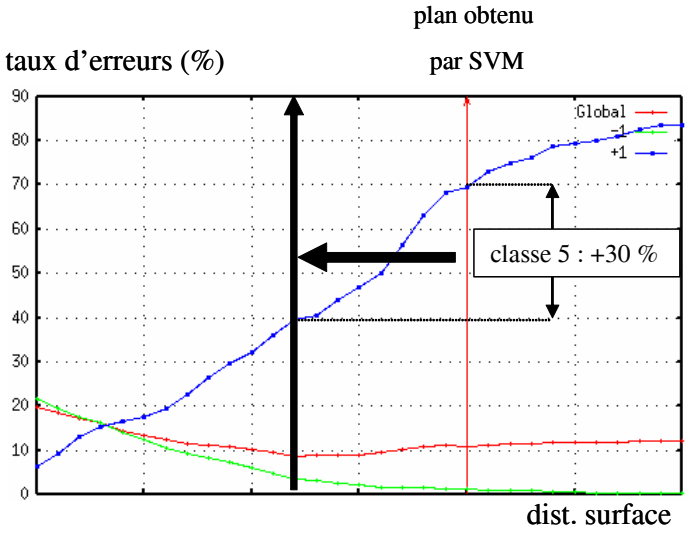

FIG. 10 – *Amélioration interactive du résultat obtenu par l'algorithme automatique de SVM*

3.2. Equivalence du post-traitement graphique interactif avec l'approche automatique d'amélioration de résultats de SVM

L'apprentissage sur des classes minoritaires à l'aide d'algorithmes automatiques de SVM a été étudié dans [Fung et Mangasarian, 2001b] ou [Shanahan et Roma, 2003a]. Nous faisons le lien entre notre approche coopérative de post-traitement et ces approches automatiques pour en montrer la validité théorique.

Nous commençons par la méthode très utilisée qui donne des poids différents aux erreurs de deux classes pour régler le problème de classe minoritaire. Soit $c_{+1} = c*p_{+1}$, $c_{-1} = c*p_{-1}$ correspondant aux constantes utilisées pour régler les erreurs de la classe *+1* et *-1*, on cherche simultanément à maximiser la marge et minimiser les erreurs avec des poids

différents. La marge est fixée, par contre le scalaire de la surface de séparation change. C'est-à-dire que la surface de séparation des données est déplacée parallèlement à elle-même en la comparant avec celle obtenue par des poids égaux entre les deux classes. La figure 11 illustre la méthode.

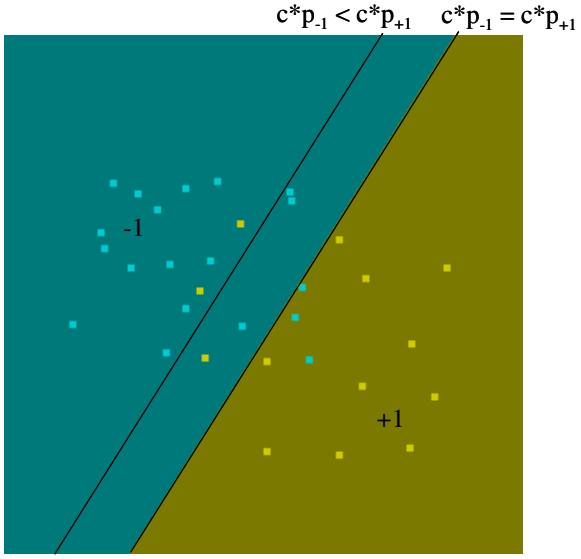

FIG. 11 – *Ajustement de poids d'erreurs des deux classes*

D'autres méthodes automatiques proposées dans [Chen et Mangasarian, 1996] ou [Fung et Mangasarian, 2001b] visent à améliorer les résultats en sortie de SVM. Elles déplacent la surface obtenue de manière parallèle à elle-même en utilisant la méthode de Newton. Une méthode proposée par [Shanahan et Roma, 2003a] ou [Shanahan et Roma, 2003b] déplace la surface obtenue en sortie de l'algorithme de SVM en changeant le scalaire de cette surface pour améliorer la performance. Ces méthodes ont montré de nettes améliorations des résultats de SVM dans les applications où il y a une classe minoritaire. Cependant, il est nécessaire d'adapter ces paramètres, au contraire, notre approche fonctionne de manière graphique, interactive et l'utilisateur peut trouver les paramètres optimums (le minimum global d'une courbe) de manière fiable avec un coût presque nul.

3.3. Résultats expérimentaux

	#classes	#individus	#dims	protocole de test
Wine	3	178	13	10-fold
Adn	3	3186	180	2000 trn – 1186 tst
Satimage	6	6435	36	4435 trn – 2000 tst
Vehicle	4	846	18	9-fold
Segment	7	2310	19	10-fold

TAB 1 – *Description des données de Statlog et de l'UCI*

Pour évaluer ce travail, nous présentons les résultats expérimentaux obtenus sur les ensembles de données de Statlog et de l'UCI décrits dans le tableau 1.

Nous avons utilisé l'algorithme PSVM [Fung et Mangasarian, 2001a] de classification multi-classes en utilisant le noyau linéaire. Notre outil graphique interactif a été utilisé en coopération avec les PSVM linéaire pour améliorer les résultats obtenus qui sont présentés dans le tableau 5.2 (les meilleurs résultats sont en caractères gras).

	PSVM	PSVM + post-traitement graphique interactif
Wine	98,90 %	**99,40 % (+ 0,50 %)**
Adn	94,44 %	**95,45 % (+ 1,01 %)**
Satimage	80,90 %	**82,40 % (+ 1,50 %)**
Vehicle	75,77 %	**78,02 % (+ 2,25 %)**
Segment	85,93 %	**90,80 % (+ 4,87 %)**

TAB 2 – *Performance du post-traitement graphique interactif*

On remarque que les taux de précision obtenus par le post-traitement graphique interactif sont dans tous les cas supérieurs à ceux obtenus par le PSVM linéaire original. Ceci montre la performance de notre approche de post-traitement graphique interactif. De plus, au contraire des autres approches automatiques de post-traitement, aucun réglage des paramètres n'est nécessaire. L'utilisateur trouve lui-même le résultat optimal (le minimum d'une courbe d'erreur) de manière visuelle et fiable avec un coût presque nul.

Nous avons présenté une méthode de coopération entre visualisation et algorithme automatique de SVM dans l'étape de post-traitement pour d'une part interpréter les résultats de classification et d'autre part améliorer ces résultats.

Nous présentons ensuite une autre approche coopérative [Do et Poulet, 2004c] à l'aide de méthodes de visualisation et d'algorithmes de SVM permettant d'impliquer plus significativement l'utilisateur dans la tâche de classification.

4. Vis-SVM : approche coopérative visualisation-RSVM pour la classification de données

Le but de l'extraction de connaissances à partir de données est de pouvoir extraire des informations intéressantes de grands ensembles de données pour une application connue a priori. L'intérêt des connaissances extraites est validé en fonction du but de l'application. L'utilisateur détermine l'efficacité des résultats obtenus si son but est atteint. Donc, l'outil d'extraction de connaissances doit être fortement interactif. L'idée ici est d'augmenter l'implication de l'utilisateur en le faisant participer à la construction du modèle à l'aide de techniques graphiques interactives. Ce type d'approche est centré sur l'utilisateur qui peut être l'expert du domaine des données. Dans ce dernier cas, cela offre la possibilité d'utiliser les compétences et connaissances du spécialiste des données lors de la construction du modèle. Cela permet de construire efficacement un bon modèle. L'utilisateur participe activement à la construction du modèle, il a donc une meilleure compréhension du modèle construit et une meilleure confiance dans ce modèle. L'avantage d'une plus grande implication de l'humain dans le processus de fouille de données est aussi d'utiliser les capacités humaines en reconnaissance de formes par le biais de méthodes de visualisation adéquates.

Nous présentons une nouvelle approche coopérative entre une visualisation multi-vue et l'algorithme automatique de Reduced SVM (RSVM [Lee et Mangasarian, 2000]) qui permet d'augmenter plus significativement le rôle de l'utilisateur dans la construction du modèle de classification.

Nous commençons par décrire l'algorithme automatique de RSVM basé sur l'algorithme de PSVM que nous utilisons dans l'approche coopérative en classification.

4.1. Algorithme de RSVM

Soit une tâche de classification linéaire d'un ensemble de données x_i *(i = 1, 2, ..., m)* en deux classes $y_i = \pm 1$. Chaque individu est représenté dans un espace de dimension n (nombre d'attributs). Le meilleur plan de séparation des données est le plan qui classifie correctement les données (lorsque c'est possible) et qui se trouve le plus loin possible des deux classes. Le but de l'algorithme est de maximiser la marge de séparation ou la distance les entre deux plans support des deux classes ($x.w - b = +1$ est le plan support pour la classe $+1$ et $x.w - b = -1$ est le plan support pour la classe -1). Tous les individus d'une classe sont d'un seul côté du plan support de cette classe. L'hyperplan optimal passe au milieu des deux plans support. Dans le cas où des exemples ne sont pas linéairement séparables, on ne relaxe que les contraintes pour que chaque exemple soit du côté approprié de son plan support. Les distances des erreurs sont notées par des variables de ressort $(z_i \geq 0\ ;\ i=1, 2, ..., m)$. Si l'exemple x_k est du bon côté de son plan support, alors z_k est égal à 0. La recherche de l'hyperplan optimal se ramène à maximiser la marge mais aussi minimiser les distances aux erreurs. La formulation primale d'un algorithme de SVM standard est la suivante :

$$\min\ \Psi(w, b, z) = (1/2)\ ||w||^2 + c \sum_{i=1}^{m} z_i$$

$$\text{avec :} \qquad y_i(w.x_i - b) + z_i \geq 1 \qquad z_i \geq 0 \quad (i=1, 2, ..., m) \qquad (1)$$

où une constante $c > 0$ est utilisée pour contrôler la marge et les erreurs.

L'hyperplan (w,b) est obtenu par la résolution de (1). La classification d'un nouvel individu x est calculée par le signe de $(x.w - b)$.

L'algorithme de proximal SVM [Fung et Mangasarian, 2001a] modifie la formule de l'algorithme de SVM standard en :

- maximisant la marge par $(1/2)\ ||w, b||^2$
- minimisant les erreurs par $(c/2)\ ||z||^2$
- remplaçant l'inégalité par l'égalité : $y_i(w.x_i - b) + z_i = 1$

En substituant z à w et b dans la fonction objectif Ψ, on obtient un problème d'optimisation non contrainte. La condition nécessaire et suffisante pour que Ψ soit minimale est que les dérivées premières en w et b soient nulles. Il faut alors résoudre un système linéaire à $(n+1)$ inconnues (les n coordonnées de w et le scalaire b) au lieu du programme quadratique. La complexité du PSVM varie linéairement avec le nombre d'individus et avec le carré du nombre d'attributs. Le PSVM classifie $2x10^6$ individus avec 10 attributs et 2 classes en 15 secondes sur un PC (P4-2,4 GHz, 256 Mo RAM). Son taux de précision est équivalent aux autres algorithmes. La version incrémentale en ligne de PSVM

peut traiter sans difficulté des fichiers de très grandes tailles en ligne sur des machines standard où les autres algorithmes nécessiteront des capacités en mémoire vive beaucoup plus importantes et donc des machines spécifiques (type serveurs). De plus il est facilement parallélisable, [Poulet et Do, 2003] ont traité des fichiers d'un milliard d'individus en moins de sept minutes sur dix PC standard (P4-2,4 GHz, 256 Mo RAM). Nous avons appliqué le théorème de Sherman-Morrison-Woodbury [Golub et Van Loan, 1996] à l'algorithme de PSVM linéaire pour adapter le PSVM aux ensembles de données ayant soit un très grand nombre de dimensions soit un très grand nombre d'individus. Les performances de l'algorithme [Do et Poulet, 2003b] ont été évaluées sur des ensembles de données biomédicales. Pour pouvoir traiter des ensembles de données de très grande taille en nombre d'individus et en nombre d'attributs, nous avons étendu l'algorithme de PSVM linéaire en utilisant un algorithme de boosting [Do et Poulet, 2004a]. Cependant, ces algorithmes se limitent à la classification linéaire de données.

Pour pouvoir classifier des données non linéairement séparables, l'algorithme de PSVM utilise une matrice de noyau *K[mxm]* au lieu de la matrice *A[mxn]* représentant les données. On peut construire facilement une matrice de noyau *K[mxm]* en entrée du PSVM en utilisant l'ensemble des *m* individus. La taille de cette matrice varie avec le carré du nombre d'individus. Toutes les données sont utilisées comme vecteurs support, et on ne génère donc pas une bonne surface de séparation. De plus, la mise en œuvre d'un algorithme de PSVM dans le cas non linéaire est coûteuse en temps et mémoire.

Pour remédier à ce problème, l'algorithme de Reduced SVM [Lee et Mangasarian, 2000] utilise un échantillon aléatoire de taille *s* (comme ensemble de vecteurs support) pour créer une matrice de noyau *K[mxs] (s << m)*. Mais la question qui se pose ici à l'utilisateur est de savoir quel type de fonction de noyau il doit utiliser pour obtenir de bons résultats. Par ailleurs, le choix des paramètres d'un type de noyau basé sur la validation croisée est connu comme une tâche très coûteuse en temps de calcul.

Nous proposons donc d'utiliser un ensemble de méthodes de visualisation permettant à l'utilisateur de sélectionner les vecteurs support pour la classification à l'aide de RSVM. L'utilisateur utilise l'information représentée graphiquement pour construire la matrice de noyau.

4.2. Approche coopérative pour la classification de données

L'approche coopérative utilise une visualisation multi-vue pour visualiser les données en se basant sur des méthodes graphiques interactives. Elle essaie d'impliquer plus significativement l'utilisateur dans la construction du modèle de SVM. Ce type d'approche présente au moins les avantages suivants :

- on a la possibilité d'utiliser les capacités humaines en reconnaissance de formes par le biais de méthodes de visualisation adéquates,
- l'utilisateur participe activement à la construction du modèle, donc il a une meilleure compréhension du modèle construit et une meilleure confiance dans ce modèle,
- enfin le dernier avantage d'une plus grande implication de l'humain dans le processus de traitement des données et que l'on peut utiliser les compétences et connaissances du spécialiste du domaine des données lors de l'ensemble du processus de fouille, si l'utilisateur est le spécialiste des données.

Le point de départ est la visualisation de l'ensemble de données. L'utilisateur choisit les méthodes les plus appropriées à la présentation de ses données. L'outil coopératif fournit des

mécanismes de zoom, rotation, linking ou brushing permettant à l'utilisateur de sélectionner les individus proches de la frontière de séparation des données. Ces individus sont utilisés comme vecteurs support en entrée de l'algorithme de RSVM.

Le tableau 3 présente l'algorithme coopératif Vis-SVM.

Entrée :
- les m individus en n dimensions et 2 classes

Construction coopérative du modèle à l'aide de SVM :
1- l'utilisateur choisit les méthodes de visualisation de données appropriées,
2- il sélectionne les individus les plus proches de la frontière de séparation de données pour servir de vecteurs support,
3- à partir de la visualisation de données, l'utilisateur a quelques informations sur le choix de la fonction de noyau et les paramètres d'entrée de l'algorithme de RSVM,
4- l'algorithme de RSVM effectue la classification des données,
5- si l'utilisateur n'est pas satisfait du résultat alors
il répète ce processus en retournant au pas 1, 2 ou 3,
sinon
c'est terminé

TAB 3 – Algorithme coopératif de Vis-SVM

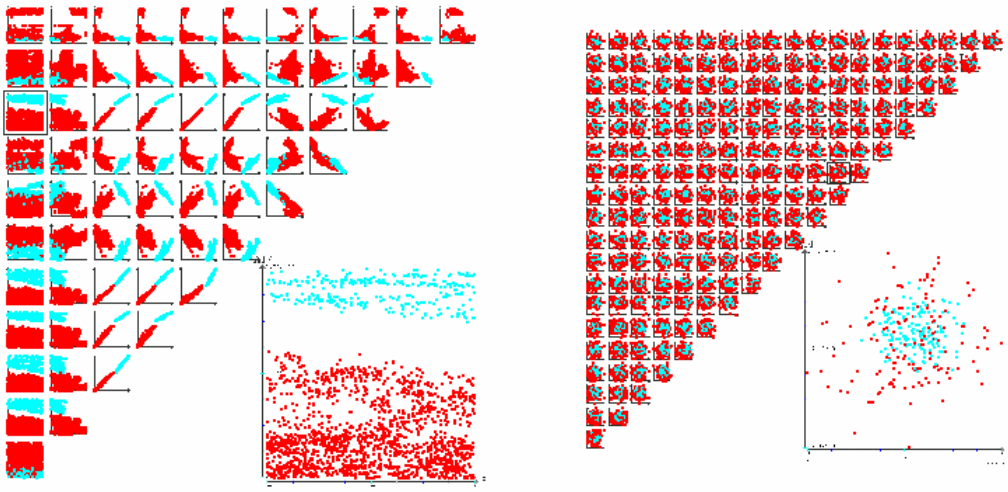

FIG. 12 – *Visualisation de données linéairement et non linéairement séparables*

De plus, la visualisation de données fournit également des informations sur la complexité de la frontière de séparation des données. L'utilisateur a donc des informations pour le choix

de la fonction de noyau et des paramètres d'entrée de l'algorithme. Avec les algorithmes automatiques de SVM, cette étape d'adaptation du noyau et des paramètres est connue comme une tâche coûteuse en temps d'exécution. La visualisation utilise les capacités humaines en reconnaissance de formes et permet de réduire de manière significative le coût de l'opération.

Par exemple, si la visualisation des données montre que l'ensemble est linéairement séparable comme sur la figure 12 (à gauche), l'utilisateur peut alors choisir un noyau linéaire pour séparer les données. Si la visualisation montre que la frontière de séparation n'est pas linéaire comme sur la figure 12 (à droite), l'utilisateur peut utiliser des fonctions de noyau non linéaires comme un RBF ou une fonction polynomiale. Enfin, l'utilisateur obtient les résultats en sortie de l'algorithme de RSVM sur l'ensemble de données. Si nécessaire, il peut répéter ce processus pour améliorer le modèle obtenu.

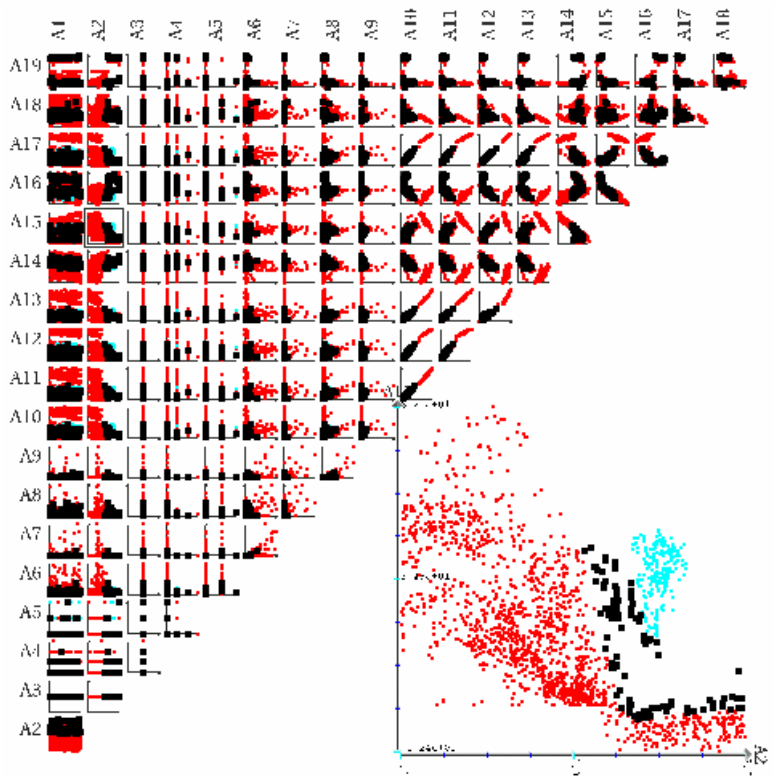

FIG. 13 – *Sélection interactive des points à la frontière pour servir de vecteurs support*

La figure 13 montre la visualisation des données Segment de l'UCI avec les matrices de scatter-plot en 2D. Cet ensemble de données contient *2 310* individus en *19* dimensions avec *7* classes. Au départ, l'outil coopératif présente les données dans une vue et la sélection interactive des vecteurs support commence. Si par exemple, l'utilisateur veut séparer les individus de la classe *6* (considéré comme la classe *+1* avec les points en clair) des individus des autres classes (considéré comme la classe *-1* avec les points en foncé). Il sélectionne les points les plus proches de la frontière de séparation qui seront ensuite utilisés comme

vecteurs support. La visualisation montre que la séparation de la classe *6* du reste est non linéaire et simple. Cela apporte des informations pour le choix de la fonction de noyau et de ses paramètres. Ici, on peut choisir d'utiliser un noyau RBF (i.e. $K(x,y)= exp(-\gamma||x-y||^2)$ mesure la similarité entre les individus x et y) avec une valeur faible du paramètre $\gamma = 0.0001$ (on peut choisir une valeur faible de γ si la frontière de séparation des données est simple et on augmente la valeur de γ si l'on veut obtenir une frontière plus serré autour de la classe *+1*). L'algorithme de RSVM classifie cet ensemble de données (classe *6* contre le reste) avec *100,00 %* de taux de bon classement. L'utilisateur peut répéter ce processus pour améliorer le résultat ou bien passer à la classe suivante.

4.3. Résultats expérimentaux

Nous avons développé un environnement graphique de fouille de données en C/C++ sous IRIX (station SGI-O2) et Linux (PC) [Poulet, 2002a]. L'environnement contient une dizaine de méthodes de visualisation et plusieurs algorithmes de classification, dont des algorithmes de SVM pour la fouille de très grands ensembles de données.

Nous présentons les performances obtenues par l'approche coopérative (Vis-SVM). Nous avons utilisé notre méthode coopérative et l'algorithme automatique LibSVM pour classifier les ensembles de données de l'UCI, Statlog et de Delve sur un PC (Pentium-4, 2,4 GHz, 512 Mo RAM, Linux). Le tableau 4 décrit les ensembles de données.

	#classes	#individus	#dims	protocole de test
Bupa	2	345	6	10-fold
Pima	2	768	8	10-fold
Twonorm	2	7400	20	300 trn – 7100 tst
Ringnorm	2	7400	20	300 trn – 7100 tst
Segment	7	2310	19	10-fold
Satimage	6	6435	36	4435 trn – 2000 tst

TAB 4 – *Description des données de l'UCI, Statlog et de Delve*

	Vis-SVM	LibSVM
Bupa	**76.18 %**	73.62 %
Pima	**78.86 %**	77.34 %
Twonorm	97.28 %	**97.35 %**
Ringnorm	97.15 %	**97.28 %**
Segment	96.02 %	**97.10 %**
Satimage	91.70 %	**92.05%**

TAB 5 – *Résultats sur les données de l'UCI, Statlog et de Delve*

Nous avons obtenu les résultats concernant les taux de précision présentés dans le tableau 5. Nous avons utilisé l'approche « un contre le reste » pour traiter les ensembles de données multi-classes. Cela est moins performant que l'approche « un contre un » utilisée par LibSVM, mais le temps d'exécution est également moins important. Nous avons choisi les fonctions de noyau RBF pour tous les ensembles de données. Les meilleurs résultats sont en caractères gras dans le tableau 5.

Ces résultats montrent que notre méthode coopérative Vis-SVM donne de bons taux de précision comparables à l'algorithme automatique LibSVM. Mais, la construction des matrices de noyau (le choix du type de noyau et les paramètres) pour un algorithme automatique de SVM est une tâche très coûteuse en temps d'exécution. Avec notre outil coopératif, les vues appropriées de l'ensemble de données apportent des informations pertinentes pour construire une bonne fonction de noyau.

L'approche coopérative implique plus significativement l'humain dans la construction du modèle de SVM grâce aux méthodes de visualisation et ainsi, l'utilisateur a une meilleure compréhension des données et du modèle construit. La principale limite de notre approche comme de beaucoup d'autres approches graphiques concerne la capacité à traiter de grands ensembles de données

Pour pouvoir traiter de très grands ensembles de données avec une approche coopérative, nous proposons d'utiliser une méthode de prétraitement de données [Do et Poulet, 2004d].

5. Prétraitement de données très volumineuses

Le développement du matériel de stockage permet à de nombreuses organisations de constituer de très grands ensembles de données. Les chercheurs de l'université Berkeley ont estimé que la quantité d'informations dans le monde augmente d'environ un exa octet tous les ans. Dans plusieurs domaines, les données arrivent plus rapidement que l'on peut découvrir des connaissances dans ces données. Nous citons ici quelques exemples : l'entrepôt de données de Walmart enregistre *20* millions de transactions par jour, le moteur de recherche sur internet Google a *70* millions de recherches par jour ou la compagnie AT&T produit *275* millions d'appels par jour, etc. Il en est de même pour les entrepôts de données scientifiques, par exemple : la base de données Astronomy contient deux milliards d'objets stellaires, la base Satellite Photos stocke des tera octets de données, une centaine de giga octets est disponible dans la base d'El Nino, etc.

La plupart des techniques de visualisation de données ne peuvent pas traiter tel quel les ensembles de grande taille. Une limite incontournable est la résolution de l'affichage. Par exemple avec une résolution de *1600x1200*, on peut présenter *1 920 000* pixels sur l'écran, la meilleure méthode de pixelisation ne peut pas visualiser plus de *100 000* individus en *20* dimensions. Plusieurs approches peuvent permettre de remédier à ce problème :

- utiliser une représentation de plus haut niveau : on ne s'intéresse pas à visualiser les données, mais une représentation de plus haut niveau des données qui présente des méta-données à un niveau plus abstrait. Par exemple, les méthodes qui reposent sur la définition d'une mesure de similarité permettant de regrouper ou filtrer les données.

- la représentation de données symboliques : on peut aussi utiliser une représentation symbolique des données, plusieurs travaux sur l'analyse de données symboliques [Bock et Diday, 1999] ont déjà été étudiés. L'utilisation d'attributs symboliques comme des histogrammes permet de pouvoir représenter sous une forme résumée l'ensemble de données traité. Il est donc possible dans ce cas de traiter des ensembles de données de très grandes tailles puisque ce ne sont pas les données elles-mêmes qui sont visualisées, mais une représentation plus succincte de celles-ci. Les problèmes à résoudre sont alors d'adapter les algorithmes interactifs et ou automatiques (par exemple de classification supervisée ou non supervisée) à ce nouveau type de variables.

- la coopération entre méthodes automatiques et méthodes de visualisation : une autre solution est d'utiliser des méthodes de visualisation en coopération avec des méthodes

automatiques, par exemple on peut utiliser des méthodes automatiques pour effectuer une sélection ou une réduction des données (soit en nombre d'individus, soit en nombre de dimensions). On bénéficie alors des avantages des deux types de méthodes, une meilleure compréhension par les méthodes graphiques et une capacité de traitement de grandes quantités de données avec des méthodes automatiques.

Nous présentons une approche de coopération entre méthodes automatiques et méthodes de visualisation pour traiter de grands ensembles de données.

5.1. Prétraitement pour un grand nombre d'individus

Pour pouvoir traiter des ensembles de données ayant un très grand nombre d'individus, nous proposons une méthode pour réduire la taille des ensembles de données. Nous créons d'abord des clusters en utilisant l'algorithme des k-moyennes [MacQueen, 1967] ou un algorithme de carte auto-organisatrice [Kohonen, 1995]. Ensuite, on effectue un échantillonnage à partir des clusters. Le résultat de cet échantillonnage est alors utilisé avec l'outil coopératif décrit précédemment.

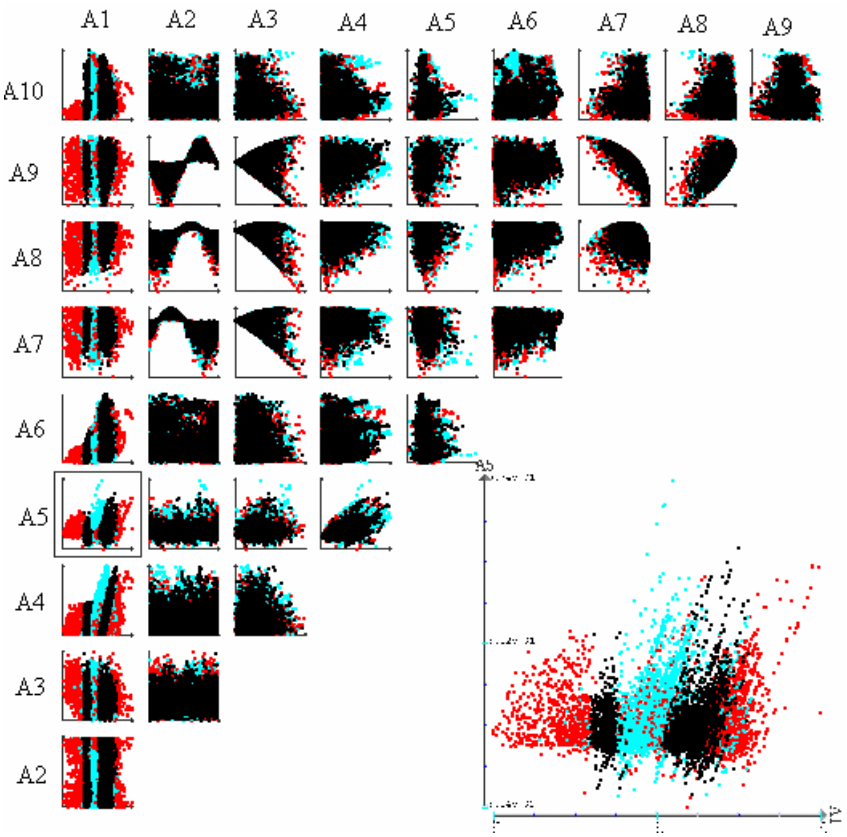

FIG. 14 – *Sélection interactive des vecteurs support des données Forest Cover Type*

Nous illustrons l'efficacité de notre approche sur un grand ensemble de données : Forest Cover Type de l'UCI. Cet ensemble de données contient *581 012* individus en *54* dimensions avec *7* classes. C'est un problème reconnu comme difficile à traiter avec des algorithmes de SVM. Collobert et ses collègues [Collobert et al., 2002] ont classifié la classe *2* contre le reste en utilisant SVMTorch [Collobert et Bengio, 2001] avec une fonction de noyau RBF. Ils ont pour cela utilisé *100 000* individus pour l'ensemble d'apprentissage et *50 000* individus pour l'ensemble de test. L'apprentissage a nécessité *2* jours et *5* heures sur un PC (Athlon, 1,2 GHz, 512 Mo RAM) pour obtenir *83,24 %* de taux de précision sur l'ensemble de test.

Nous avons classifié la classe *2* contre le reste sur un PC (Pentium-4, 2,4 GHz, 512 Mo RAM). Nous avons utilisé *500 000* individus pour l'apprentissage et le reste (*81 000* individus) pour le test. Avec un noyau RBF, LibSVM [Chang et Lin, 2003] n'a pas terminé l'apprentissage après plusieurs jours.

Pour traiter cet ensemble de données avec l'outil coopératif, nous avons eu besoin d'une heure pour pré-traiter l'ensemble de données. Nous avons créé *200* clusters (*100* clusters pour chaque classe) et puis nous avons créé un échantillon de *2 %* des individus de chaque cluster. Nous avons approximativement obtenu *10 000* individus. Ensuite, nous avons pu sélectionner interactivement les vecteurs support comme montré sur la figure 14. Une matrice de noyau RBF avec la valeur $\gamma = 2$ a été créée en entrée de l'algorithme de RSVM. La tâche d'apprentissage a nécessité *8* heures pour un taux de bon classement de *84,32 %* sur l'ensemble de test. C'est un premier résultat prometteur en ce qui concerne le taux de précision. En ce qui concerne le temps d'exécution, notre approche est *75* fois plus rapide que l'approche proposée par [Collobert et al., 2002], ceci est dû au fait que la complexité de l'algorithme de SVM utilisé est proportionnelle au carré du nombre d'individus et que notre machine est *deux* fois plus rapide que celle de Collobert.

5.2. Prétraitement pour un grand nombre de colonnes

Dans certaines applications, comme par exemple l'analyse d'expression de gènes, on traite des ensembles de données ayant un très grand nombre de dimensions (de l'ordre de 10^3 à 10^5) et un nombre plus restreint d'individus (de l'ordre de 10^2 à 10^4). La plupart des algorithmes de classification supervisée ont alors beaucoup de difficulté à traiter de tels ensembles de données. La démarche générale dans ce cas est de chercher à réduire le nombre de dimensions. On peut faire une différence entre les algorithmes de réduction de dimensions et de sélection de dimensions. Dans ce dernier cas, les dimensions qui sont éliminées n'apportent pas d'information sur la classification, on ne perd donc pas d'information contrairement au cas de la réduction de dimensions. On cherche donc à éliminer les dimensions qui n'apportent que peu (ou pas) d'information pour la classification. Le but de cette étape de prétraitement des données est de sélectionner un sous-ensemble de dimensions sans pour autant perdre trop d'information. Nous utilisons un algorithme particulier de SVM, le SVM norme-1 avec un noyau linéaire [Fung et Mangasarian, 2002]. En résumé, cet algorithme cherche le meilleur hyperplan de séparation en maximisant la marge et minimisant les erreurs. Pour cela, il cherche à minimiser la norme-1 de l'hyperplan (au lieu de la norme-2 de l'hyperplan dans le cas des SVM standard). C'est un algorithme très efficace pour sélectionner un sous-ensemble de dimensions. Les résultats obtenus par le SVM norme-1 sont comparables à ceux obtenus par les SVM standards, la particularité de

cet algorithme de SVM norme-1 est qu'il donne un hyperplan de séparation avec la plupart des coefficients nuls. Les dimensions correspondant aux coefficients nuls sont supprimées.

Pour évaluer ce travail, nous présentons les résultats obtenus sur des ensembles de données biomédicales [Jinyan et Huiqing, 2002] ayant un très grand nombre de dimensions. Nous avons utilisé le programme LibSVM avec un noyau linéaire pour classifier les données dans le cas où toutes les dimensions sont traitées. Les ensembles de données utilisés sont décrits dans le tableau 6.

	#classes	#individus	#dims	protocole de test
AML-ALL Leukemia	2	72	7129	38 trn – 34 tst
Breast Cancer	2	97	24481	78 trn – 19 tst
Colon Tumor	2	62	2000	Leave-1-out
Lung Cancer	2	181	12533	32 trn – 149 tst
Ovarian Cancer	2	253	15154	Leave-1-out

TAB 6 – *Description des ensembles de données biomédicales*

	précision classe +1		précision classe -1		Précision	
	sélection	sans sélection	Sélection	sans sélection	sélection	sans Sélection
AML-ALL Leukemia	**100 %** 5-dim	95 %	85.71 % 5-dim	**92.86 %**	**94.12 %** 5-dim	**94.12 %**
Breast Cancer	**91.67 %** 10-dim	83.33 %	**57.14 %** 10-dim	**57.14 %**	**78.95 %** 10-dim	73.68 %
Colon Tumor	**95.45 %** 19-dim	86.36 %	**97.5 %** 19-dim	92.5 %	**96.77 %** 19-dim	90.32 %
Lung Cancer	**100 %** 9-dim	**100 %**	96.27 % 9-dim	**98.51 %**	96.64 % 9-dim	**98.66 %**
Ovarian Cancer	**100 %** 13-dim	**100 %**	**100 %** 10-dim	**100 %**	**100 %** 13-dim	**100 %**

TAB 7 – *Performance en terme de taux de précision sur des données biomédicales*

Les résultats obtenus après avoir sélectionné les dimensions sont comparés avec ceux obtenus par une classification sur l'ensemble des dimensions. Les résultats concernant le taux de précision sont présentés dans le tableau 7 (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. Il est intéressant aussi de constater que le nombre de dimensions utilisées est réduit de manière très significative : par exemple sur Breast Cancer on passe de *24 481* dimensions à *10* dimensions sans perte de précision (c'est même l'inverse qui se produit puisque le taux de précision est amélioré de *5 %*) et sur AML-ALL Leukemia, on passe de *7 129* à *5* dimensions (soit une diminution d'un facteur *1 400*) en conservant exactement le même taux de précision. Ensuite, les outils graphiques interactifs peuvent être utilisés pour interpréter les résultats de SVM sur les sous-ensembles de dimensions.

6. Conclusion-perspectives

Nous avons présenté différentes coopérations entre les méthodes de visualisation et les algorithmes automatiques de SVM permettant d'augmenter le rôle de l'humain dans le processus d'ECD. Nous avons proposé une approche graphique interactive pour interpréter les résultats de SVM. On utilise la distribution des individus en fonction de la distance à la surface avec d'autres méthodes graphiques dans le cadre de la visualisation multi-vue pour interpréter les résultats de SVM en classification, régression et détection d'individus atypiques. Cette méthode améliore la compréhensibilité de l'utilisateur dans les modèles obtenus par les SVM. Les dimensions intéressantes dans le modèle sont acquises de manière visuelle grâce à la représentation graphique des résultats de SVM.

Nous avons étendu l'approche d'interprétation graphique pour améliorer les résultats obtenus par les algorithmes de SVM dans le cadre de la classification multi-classes.

Nous avons présenté une approche coopérative pour construire des modèles de classification de SVM. Cette approche utilise des méthodes de visualisation et des algorithmes de SVM pour pouvoir impliquer plus significativement l'utilisateur dans la tâche de classification. Avec les méthodes de visualisation appropriées, on peut bénéficier des capacités humaines en reconnaissance de formes. Nous avons montré comment l'utilisateur peut interactivement utiliser la méthode coopérative pour sélectionner les vecteurs support. La visualisation de données fournit également des informations intéressantes pour construire la matrice de noyau en entrée de l'algorithme automatique de RSVM. Les résultats expérimentaux sur les données de l'UCI, Statlog et de Delve montrent que notre approche coopérative donne de bons taux de précision comparés à l'algorithme automatique de LibSVM. L'utilisateur a une bonne compréhension des données et du modèle construit.

Enfin des méthodes de pré-traitement de données ont été proposées pour traiter de très grands ensembles de données.

L'extension la plus immédiate de ce travail est de comparer ce type d'approche avec les autres approches permettant de traiter de grandes quantités de données, comme par exemple utiliser des données symboliques pour avoir une représentation de plus haut niveau. Bien entendu, il serait aussi intéressant d'étudier les possibilités de mixage des différentes approches en espérant améliorer ainsi le processus et ou les résultats.

Références

[Ankerst et al., 2000] M. Ankerst, M. Ester, et H-P. Kriegel. Towards an Effective Cooperation of the Computer and the User for Classification. Proceeding. of KDD'00, 6th ACM SIGKDD, Boston, USA, 2000, pp. 179-188.

[Asimov, 1985] D. Asimov. The Grand Tour: a Tool for Viewing Multidimensional Data. *SIAM Journal on Scientific and Statistical Computing*, 6(1), pp. 128-143, 1985.

[Bennett et Campbell, 2000] K. Bennett et C. Campbell. Support Vector Machines: Hype or Hallelujah ?. *SIGKDD Explorations*, 2(2), pp. 1-13, 2000.

[Blake et Merz, 1998] C. Blake et C. Merz. UCI Repository of Machine Learning Databases. 1998. http://www.ics.uci.edu/~mlearn/MLRepository.html

[Bock et Diday, 1999] H.H. Bock et E. Diday. *Analysis of Symbolic Data*. Springer-Verlag, 1999.

[Caragea et al., 2001] D. Caragea, D. Cook et V. Honavar. Gaining Insights into Support Vector Machine Pattern Classifiers Using Projection-Based Tour Methods. Proceeding. of KDD'01, 7th ACM SIGKDD, San Francisco, USA, 2001, pp. 251-256.

[Caragea et al., 2003] D. Caragea, D. Cook et V. Honavar. Towards Simple, Easy-to-Understand, yet Accurate Classifiers. Proceeding of VDM@ICDM'03, the 3rd Int. Workshop on Visual Data Mining, Florida, USA, 2003, pp. 19-31.

[Chang et Lin, 2003] C-C. Chang et C-J. Lin. LIBSVM -- A Library for Support Vector Machines. 2003. http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[Chen et Mangasarian, 1996] C. Chen et O. Mangasarian. Hybrid Misclassification Minimization. *Advances in Computational Mathematics,* 5(2), pp. 127-136, 1996.

[Cleveland, 1993] W. Cleveland. *Visualizing Data*. AT&T Bell Laboratories, Murray Hill, NJ, Hobart Press, 1993.

[Collobert et Bengio, 2001] R. Collobert et S. Bengio. SVMTorch: Support Vector Machines for Large-Scale Regression Problems. *Journal of Machine Learning Research*, Vol. 1, pp. 143-160, 2001. ftp://ftp.idiap.ch/pub/learning/SVMTorch.tgz.

[Collobert et al., 2002] R. Collobert, S. Bengio et Y. Bengio. A Parallel Mixture of SVMs for Very Large Scale Problems. Advances in Neural Information Processing Systems, NIPS'02, MIT Press, 2002, Vol. 14, pp. 633-640.

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

[Delve, 1996] Delve. Data for Evaluating Learning in Valid Experiments. 1996. http://www.cs.toronto.edu/~delve.

[Do et Poulet, 2003a] T-N. Do et F. Poulet. Incremental SVM and Visualization Tools for Bio-medical Data Mining. Proceedings of Workshop on Data Mining and Text Mining in Bioinformatics, ECML/PKDD'03, Cavtat-Dubrovnik, pp. 14-19, 2003.

[Do et Poulet, 2003b] T-N. Do et F. Poulet. Interactive Visualization Tools for Visual Data-Mining. Proceeding of HCP'03, 14th Mini-EURO Conference, Human Centered Processes, Luxembourg, 2003, pp. 299-303.

[Do et Poulet, 2004a] T-N. Do et F. Poulet. Towards High Dimensional Data Mining with Boosting of PSVM and Visualization Tools. Proceeding of ICEIS'04, 6th Int. Conference on Entreprise Information Systems, Porto, Portugal, 2004, Vol. 2, pp. 36-41.

[Do et Poulet, 2004b] T-N. Do et F. Poulet. Interprétation graphique des résultats de SVM. SFDS'04, XXXVIe Journées de Statistiques, Montpellier, 2004.

[Do et Poulet, 2004c] T-N. Do et F. Poulet. Cooperation between Visualization Methods and SVM Algorithms for Data Mining. Proceeding of MCO'04, 5th Int. Conference on Computer Sciences, Modelling, Computation and Optimization in Information Systems and Management Sciences, Metz, France, 2004, pp. 569-576.

[Do et Poulet, 2004d] T-N. Do et F. Poulet. Enhancing SVM with Visualization. in Discovery Science 2004, E. Suzuki et S. Arikawa Eds., *Lecture Notes in Artificial Intelligence 3245*, Springer-Verlag, 2004, pp. 183-194.

[Fayyad et al., 1996] U. Fayyad, G. Piatetsky-Shapiro, et P. Smyth. From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3), pp. 37-54, 1996.

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

[Fung et Mangasarian, 2001a] G. Fung et O. Mangasarian. Proximal Support Vector Machine Classifiers. Proceedings of KDD'01, 7th ACM SIGKDD, San Francisco, 2001, pp. 77-86.

[Fung et Mangasarian, 2001b] G. Fung and O. Mangasarian. Multicategory Support Vector Machine Classifiers. Data Mining Institute Technical Report 01-06, Computer Sciences Department, University of Wisconsin, Madison, USA, 2001.

[Fung et Mangasarian, 2002] G. Fung et O. Mangasarian. 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.

[Golub et Van Loan, 1996] G. Golub et C. Van Loan. Matrix Computations. John Hopkins University Press, Balti-more, Maryland, 3rd edition, 1996.

[Guyon, 1999] I. Guyon. SVM Application List. 1999. http://www.clopinet.com/isabelle-/Projects/SVM/applist.html.

[Inselberg, 1985] A. Inselberg. The Plan with Parallel Coordinates. *Special Issue on Computational Geometry of The Visual Computer*, 1(2) , 1985, pp. 69-97.

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

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

[Keim, 1996] D. Keim. Databases and Visualization. *Tutorial Notes*, ACM-SIGMOD'96, 1996.

[Kohonen, 1995] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, Heidelberg, New York, 1995.

[MacQueen, 1967] J. MacQueen. Some Methods for classification and Analysis of Multivariate Observations. Proceeding of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1967, Vol. 1, pp. 281-297.

[Michie et al., 1994] D. Michie, D.J. Spiegelhalter et C.C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.

[Poulet, 2001a] F. Poulet. CubeVis: Voir pour Mieux Comprendre. Actes de SFDS'01, XXXIIIe Journées de Statistiques, Nantes, 2001, pp.637-640.

[Poulet, 2001b] F. Poulet. Construction Interactive d'Arbres de Décision. Actes de SFC'01, VIIIe Rencontres de la Société Francophone de Classification, Pointe à Pitre, 2001, pp. 275-282.

[Poulet, 2002a] F. Poulet. Full-View: A Visual Data Mining Environment. *International Journal of Image and Graphics*, 2(1), 2002, pp. 127-143.

[Poulet, 2002b] F. Poulet. Cooperation between Automatic Algorithms, Interactive Algorithms and Visualization Tools for Visual Data Mining. Proceeding of VDM@ECML/PKDD'02, 2nd Int. Workshop on Visual Data Mining, Helsinki, Finland, 2002, pp. 67-79.

[Poulet, 2003a] F. Poulet. Interprétation des résultats de SVM. Proceeding of SFC'2003. Actes de SFC'03, Xe Rencontres de la Société Francophone de Classification, Neuchâtel, Suisse, 2003, pp. 169-172.

[Poulet, 2003b] F. Poulet. Interactive Decision Tree Construction for Interval and Taxonomical Data. Proceeding of VDM@ICDM'03, 3rd Int. Workshop on Visual Data Mining, Florida, USA, 2003, pp. 183-194.

[Poulet, 2004] F. Poulet. Towards Visual Data Mining. Proceeding of ICEIS'04, 6th Int. Conference on Entreprise Information Systems, Porto, Portugal, 2004, Vol. 2, pp. 349-356.

[Poulet et Do, 2003] F. Poulet et T-N. Do. Mining Very Large Datasets with Support Vector Machine Algorithms. *Enterprise Information Systems V*, Kluwer Academic Publishers, 2003, pp.177-184.

[Shanahan et Roma, 2003a] J. Shanahan et N. Roma. Improving SVM Text Classification Performance through Threshold Adjustment. Proceeding of ECML'03, 14th European Conference on Machine Learning, Cavtat-Dubrovnik, Croatia, 2003, pp. 361-372.

[Shanahan et Roma, 2003b] J. Shanahan et N. Roma. Boosting support vector machines for text classification through parameter-free threshold relaxation. Proceeding of CIKM'03, the 12th International Conference on Information and Knowledge Management, New Orleans, USA, 2003, pp. 247-254.

[Torgo, 2003] L. Torgo. Regression Data Sets. 2003. http://www.liacc.up.pt/~ltorgo/Regression/DataSets.html

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

Summary

Understanding the result produced by a data-mining algorithm is as important as the accuracy. Unfortunately, support vector machine (SVM) algorithms provide only the support vectors used as "black box" to efficiently classify the data with a good accuracy. This paper presents a cooperative approach using visualization methods to gain insight into model construction task with SVM algorithms. A new visualisation tool based on a set of visualization methods (multiple-view, linking, brushing) can be used to explain the results obtained by automatic SVM algorithms in classification, regression and novelty detection tasks. This method can also be used in order to interactively improve SVM accuracy in multi-category classification task. We show how the user can interactively use cooperative tools to support the construction of SVM models and interpret them. A pre-processing step is also used for dealing with large datasets. The experimental results on Delve, Statlog, UCI and bio-medical datasets show that our cooperative tool is comparable to the automatic LibSVM algorithm, but the user has a better understanding of the obtained model.