Sélection de variables non supervisée sous contraintes hiérarchiques

Nhat-Quang Doan, Hanane Azzag, Mustapha Lebbah

Université Paris 13, Sorbonne Paris Cité,
Laboratoire d'Informatique de Paris-Nord (LIPN),
CNRS(UMR 7030),
99, avenue Jean-Baptiste Clément
93430 Villetaneuse, France
prénom.nom@lipn.univ-paris13.fr

Résumé. La sélection des variables a un rôle très important dans la fouille de données lorsqu'un grand nombre de variables est disponible. Ainsi, certaines variables peuvent être peu significatives, corrélées ou non pertinentes. Une méthode de sélection a pour objectif de mesurer la pertinence d'un ensemble utilisant principalement un critère d'évaluation. Nous présentons dans cet article un critère non supervisé permettant de mesurer la pertinence d'un sous-ensemble de variables. Ce dernier repose sur l'utilisation du score Laplacien auquel nous avons ajouté des contraintes hiérarchiques. Travailler dans le cadre non supervisé est un vrai challenge dans ce domaine dû à l'absence des étiquettes de classes. Les résultats obtenus sur plusieurs bases de tests sont très encourageants et prometteurs.

1 Introduction

La sélection de variables joue un rôle très important en classification lorsqu'un grand nombre de variables sont disponibles. Ainsi, certaines variables peuvent être peu significatives, corrélées ou non pertinentes. La sélection de variables permet également d'accélérer l'étape d'apprentissage et de réduire la complexité des algorithmes. Une méthode de sélection repose principalement sur un algorithme de recherche et un critère d'évaluation pour mesurer la pertinence des sous-ensembles potentiels de variables.

En apprentissage supervisé, la sélection de variables a largement été étudiée car il est connu que la sélection de variables peut améliorer la qualité d'un classificateur (Zhang et al., 2009). Parmi les méthodes supervisées, nous citons le coefficient de corrélation de Pearson (Rodgers et Nicewander, 1988), le score de Fisher (Duda et al., 2000) et le gain de l'information (Cover et Thomas, 2006). La sélection de variables a reçu peu d'attention en apprentissage non supervisé en comparaison au cas supervisé. Le problème devient plus difficile en raison de l'absence des étiquettes des classes pour guider la sélection. Ainsi se pose la question importante, comment évaluer la pertinence d'un sous-ensemble de fonctionnalités sans avoir recours aux étiquettes de classe ? Dans la littérature deux approches sont souvent utilisées pour évaluer la pertinence d'un sous-ensemble de variables sélectionnées, (Kohavi et John, 1997; Yu et

Liu, 2003) : l'approche de type filtrage (filter approach) et celle de type enveloppante (wrapper approach). Les approches enveloppantes évaluent les variables en utilisant un algorithme d'apprentissage qui sera finalement utilisé dans le processus de classement. Cependant, les méthodes enveloppantes sont généralement coûteuses en temps et ne peuvent pas être appliqués sur de grandes masses de données, (Kohavi et John, 1997). Ce sont les méthodes de type filtrage qui nous ont intéressés, car elles sont beaucoup plus efficaces. Les critères d'évaluation sont totalement indépendants du discriminateur utilisé. Les variables sont alors traitées avant le processus d'apprentissage. Les travaux de (Caruana et Freitag, 1994; John et al., 1994; Koller et Sahami, 1996) sur la sélection de variables montrent les différentes approches traitant ce problème d'optimisation. Parmi les méthodes de sélection d'attributs dans un contexte non supervisé, nous nous sommes intéressés principalement au score Laplacien qui est le critère d'évaluation le plus utilisé dans la littérature.

Plusieurs travaux ont tenté d'exploiter le principe du Score Laplacien (SL). Dans (Benabdeslem et Hindawi, 2011), les auteurs proposent une variante du SL qui utilise deux types de contraintes semi-supervisé sur les données : des contraintes Must-Link et des contraintes Cannot-Link. Ce score calcule la variance entre les données qui n'ont pas la même étiquette. Dans (Cai et al., 2010) les auteurs ont proposé un nouveau score appelé MCFS. Cette méthode vise à sélectionner les variables de manière à conserver la structure multi-cluster des données. MCFS mesure les corrélations entre variables d'une manière non supervisée, c'est une méthode efficace pour traiter de grande dimension, mais limitée par le choix du nombre de classes.

Dans (Zhang et hua Zhou Songcan Chen, 2007) les auteurs utilisent le même principe en proposant une nouvelle méthode appelée SSDR (Semi-supervised dimensionality reduction). Cette approche préserve la structure des données et utilise des contraintes semi-supervisés définies par les utilisateurs. D'autres auteurs proposent une méthode de sélection de variables semi-supervisé en combinant des scores calculés sur la base de données étiquetées et non étiquetés (Kalakech et al., 2011). La combinaison est simple, mais peut considérablement biaiser le résultat pour les variables ayant un meilleur score dans le cas supervisé et celles ayant de mauvais scores pour la partie non supervisée et vice-versa.

L'hypothèse sous-jacente au score Laplacien est que la structure des données dans l'espace des attributs est localement préservée dans l'espace d'attributs de sortie. En représentant cette structure par les graphes de similarité ou de distance, des données similaires dans l'espace d'entrée doivent aussi l'être quand elles sont projetées sur un vecteur d'attributs pertinents. Inspirés des travaux récents en classification non supervisée hiérarchique et aussi du modèle de classification hiérarchique AntTree (Azzag et al., 2003), nous nous sommes intéressés à l'étude du score laplacien auquel nous avons intégré de nouvelles contraintes non supervisées hiérarchiques. Le score que nous définissons est appelé SLH (Score laplacien hiérarchique). La principale contribution que nous proposons est d'utiliser une approche de construction de graphe, autre que celle qui se base sur le k-NN où k est fixé a priori. Dans notre approche nous utilisons un algorithme de classification hiérarchique autonome où la structure d'arbre fournie permet de définir un nouveau score intégrant des contraintes non supervisées basées sur l'arborescence.

2 Score Laplacien sous contraintes hiérarchiques

2.1 Le score Laplacien

Soit un ensemble de N observations $X = \{\mathbf{x_1}, ..., \mathbf{x_N}\}$. Une observation $\mathbf{x}_i$ est un vecteur de m dimensions (variables), f_{r_i} désigne le $i^{\text{ème}}$ échantillon de la $r^{\text{ème}}$ variable, $r = 1, ..., m$. Ainsi, nous définissons la $r^{\text{ème}}$ variable par $\mathbf{f_r} = [f_{r_1}, ..., f_{r_m}]^T$. Le score Laplacien sélectionne les variables pertinentes qui préservent au mieux la structure locale et qui produisent de grandes valeurs de variances. Nous supposons que les données appartenant à la même classe soient proches les unes des autres. Le SL de la $r^{\text{ème}}$ variable doit être ainsi minimisé avec la fonction suivante (He et al., 2005) :

$$SL_r = \frac{\sum_{i,j=1}^{N}(f_{r_i} - f_{r_j})^2 S_{ij}}{\sum_i (f_{r_j} - \mu_r)^2 D_{ii}} \tag{1}$$

où

$$S_{ij} = \begin{cases} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\lambda}} & \text{si } \mathbf{x}_i \text{ et } \mathbf{x}_j \text{ sont voisins} \\ 0 & \text{sinon} \end{cases}$$

et D est la matrice diagonale avec $D_{ii} = \sum_j S_{ij}$; L est la matrice Laplacienne $L = D - S$, $\mu_r = \frac{1}{N}\sum_i f_{r_i}$

2.2 Le score Laplacien Hiérarchique

L'idée de notre approche pour la sélection de variables est d'utiliser le principe des k-plus proches voisins fournis par la structure d'arbre d'AntTree (Azzag et al., 2003). Dans la littérature, de nombreux algorithmes d'apprentissage ont été proposés pour découvrir des structures sous-adjacentes dans les données en construisant un graphe de voisinage pour effectuer une analyse spectrale, (Belkin et Niyogi, 2001; Roweis et Saul, 2000; Tenenbaum et al., 2000). L'algorithme AntTree a l'avantage d'être complètement autonome et d'avoir une complexité très faible de $\theta(n \log n)$. Dans le modèle AntTree, chaque nœud de l'arbre (interne ou feuille) représente une donnée $\mathbf{x}_i$. Ainsi, les nœuds de l'arbre seront successivement ajoutés du plus haut niveau vers les niveaux inférieurs (figure 1). Toutes les données doivent passer un test de similarité où leur propriété de voisinage est vérifiée.

Soit une observation $\mathbf{x}_i$ qui va se connecter à un nœud de l'arbre $\mathbf{x}_{pos}$ si et seulement si cette action augmente la valeur de $T_{Dist}(\mathbf{x}_{pos})$. $T_{Dist}(\mathbf{x}_{pos})$ est la valeur maximale de distance (distance euclidienne) entre les nœuds fils de $\mathbf{x}_{pos}$. La règle consiste à comparer $\mathbf{x}_i$ avec son plus proche $\mathbf{x}_{i^+}$ ($\mathbf{x}_{i^+}$ est un nœud de $\mathbf{x}_{pos}$). Dans le cas où les deux nœuds sont suffisamment éloignés ($\|\mathbf{x}_i - \mathbf{x}_{i^+}\|^2 > T_{Dist}(\mathbf{x}_{pos})$), alors $\mathbf{x}_i$ se connecte à $\mathbf{x}_{pos}$. Sinon, $\mathbf{x}_i$ se déplace vers $\mathbf{x}_{i^+}$. Ainsi T_{Dist} augmente localement à chaque fois qu'un nœud se connecte à l'arbre.

Avec le nouveau score SLH, nous souhaitons définir un algorithme complètement autonome pour la sélection de variables. L'algorithme SLH est essentiellement basé sur le score Laplacien auquel nous avons ajoutés des contraintes hiérarchiques. Ainsi, au lieu d'utiliser le graphe des k plus proches voisins, nous proposons d'utiliser AntTree qui avec sa structure hiérarchique fournira automatiquement pour chaque observation $\mathbf{x_i}$ (nœud dans le cas d'AntTree) ses k_i voisins. Ces voisins représentent les nœuds du niveau inférieur qui sont directement

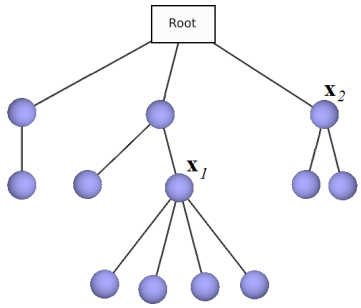


FIG. 1 – *Exemple sur la structure topologique d'arbre.*

connectés à $\mathbf{x}_i$. La figure 1 montre un exemple où l'observation $\mathbf{x}_1$ a quatre voisins ($k_1 = 4$), $\mathbf{x}_2$ a seulement deux voisins du niveau inférieur ($k_2 = 2$). En utilisant cette topologie, la nouvelle matrice d'adjacence est définie comme suit :

$$\mathbf{LH}_{ij} = \begin{cases} 1 & \text{si } \mathbf{x}_i \text{ et } \mathbf{x}_j (\forall j = [1, ..., N]) \text{ partage un lien direct} \\ 0 & \text{sinon} \end{cases}$$

Ainsi, le critère SLH du score Laplacien sous contraintes hiérarchiques est défini comme suit :

$$SLH_r = \frac{\sum_{i,j}(f_{r_i} - f_{r_j})^2 S_{ij}}{\sum_i (f_{r_i} - \alpha_r^i)^2 D_{ii}} \tag{2}$$

où

$$\alpha_r^i = \begin{cases} \frac{1}{k_i} \sum_{j, \in LH_{ij}=1} f_{r_j} & \text{s'il existe des voisins au niveau inférieur} \\ \mu_r & \text{sinon} \end{cases}$$

et

$$S_{ij} = \begin{cases} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\lambda}} & \text{si } LH_{ij}\text{=1} \\ 0 & \text{sinon} \end{cases}$$

En minimisant $\sum_{i,j}(f_{r_i} - f_{r_j})^2 S_{ij}$, nous donnons l'avantage aux attributs respectant la structure hiérarchique. En maximisant $\sum_i (f_{r_j} - \alpha_r)^2$, le score SLH sélectionne les variables ayant les plus grandes valeurs de variance locale, et qui sont les plus représentatives de la topologie de l'arbre construit. Le processus de démonstration est le même que celui présenté dans (He et al., 2005). L'Algorithme 1 présente les trois étapes nécessaires pour la sélection de variables par SLH.

3 Expérimentations

Dans cette section, plusieurs expérimentations ont été réalisées sur plusieurs bases de données réelles. Ces expérimentations sont présentées en deux parties : la qualité du clustering et la qualité de la classification supervisée en utilisant l'algorithme du plus proche voisin (1-NN). Les caractéristiques de ces bases de données sont résumées dans le tableau 3. Nous comparons

Algorithm 1 HLS.

Require: Ensemble de données X

1: Construire une structure hiérarchique utilisant l'algorithme AntTtree
2: pour r = 1 à m
    calculer SLH_r pour chaque variable r suivant l'Equation 2
   fin
3: Trier SLH_r par ordre croissant correspondant au score obtenu.

Base de données	**# Taille**	**# Variables**	**# Classes**
ARP10P	130	2400	10
Coil20 (Cai et al., 2011)	1440	1024	20
Isolet (Cai et al., 2011)	1559	617	26
Sonar (Frank et Asuncion, 2010)	208	60	2
Soybean (Frank et Asuncion, 2010)	47	35	4

TAB. 1 – *Propriété des bases données. La base ARP10P est disponible sur le lien http ://featureselection.asu.edu/datasets.php.*

les performances de notre approche avec le score Laplacien classique et l'algorithme MaxVariance qui permet de sélectionner les attributs qui maximisent la variance.

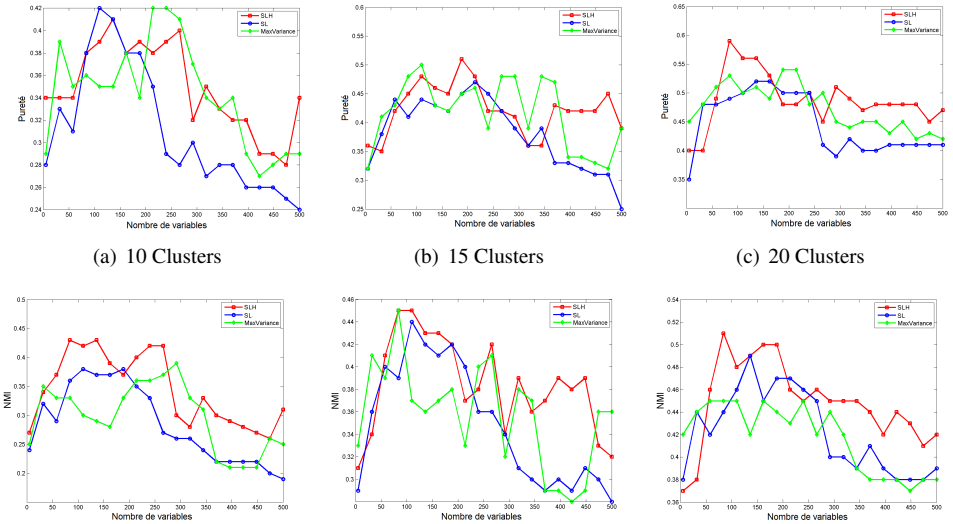
3.1 Comparaison dans un cadre non supervisé

Pour évaluer la qualité du clustering, nous utilisons deux mesures : la pureté et l'Information Mutuelle Normalisée (NMI - Normalized Mutual Information) (Strehl et al., 2002) ; chacune doit être maximisée. Pour faciliter la comparaison entre les méthodes nous appliquons l'algorithme *K-means* sur la base de données en prenant en compte que les variables sélectionnées.

Dans ces expérimentations, pour construire le graphe k-NN du SL nous fixons le paramètre $k = 5$. Nous évaluons ensuite la qualité du clustering avec différentes valeurs pour le nombre de clusters de la manière suivante : K = 10, 15, 20 pour AR10P, K = 10, 20, 30 pour Coil20, $K = 13$, 26, 39 pour Isolet et K = 4, 6, 8 pour Sonar et Soybean. Pour chaque valeur de K, nous exécutons les différents critères pour sélectionner les $m*$ variables ($m* = [1, ..., 500]$ pour AR10P, $m* = [1, ..., 200]$ pour Coil20 et Isolet, $m* = [1, ..., 60]$ pour Sonar et m $m* = [1, ..., 35]$ pour Soybean).

Les figures 2 jusqu'à 6 montrent les courbes des performances sur le clustering (Pureté et NMI) par rapport au nombre de variables sélectionnées. De manière générale notre approche SLH obtient de meilleurs résultats par rapport aux autres méthodes. Dans la figure 2, nous observons que l'algorithme SLH fournit des résultats raisonnables par rapport aux critères de Pureté et du NMI. Pour les données Coil20 (Fig. 3), SLH est meilleur en terme de Pureté et NMI lorsque le nombre de variables est au alentour de 50 à 100 pour les trois expérimentations. Nous notons que pour K = 30, notre algorithme est nettement meilleur que les autres. De plus, dans la figure 4, la pureté de SLH augmente de manière constante. Les mêmes remarques sont observées pour la base Sonar (Fig.5). Pour Soybean (Fig.6) SLH atteint une valeur de Pureté = 100% pour la plupart des cas ($K = 4, 6$ et 8) en utilisant seulement 9 variables.

Dans les tables 3.1 jusqu'à 3.1, nous résumons les résultats du clustering obtenus sur toutes les bases testées. Les résultats numériques obtenus avec la base AR10P (Tab. 3.1) montrent une amélioration des performances du NMI de SLH. Pour la Pureté, SLH est de même qualité que SL et MaxVariance. Dans le tableau 3.1 nous remarquons que les résultats fournis par coil20, pour 20 clusters et 100 variables sélectionnées, donnent une valeur de NMI de 68% pour SLH, ce qui est mieux que si on avait utilisé les 1024 variables (66,0%). Dans le tableau 3.1, les résultats de Pureté et du NMI en utilisant 100 variables ne sont pas les meilleurs, mais restent proches de ceux utilisant 617 variables. Pour la base *Sonar* SLH est la seule méthode qui permet d'obtenir de meilleurs résultats.



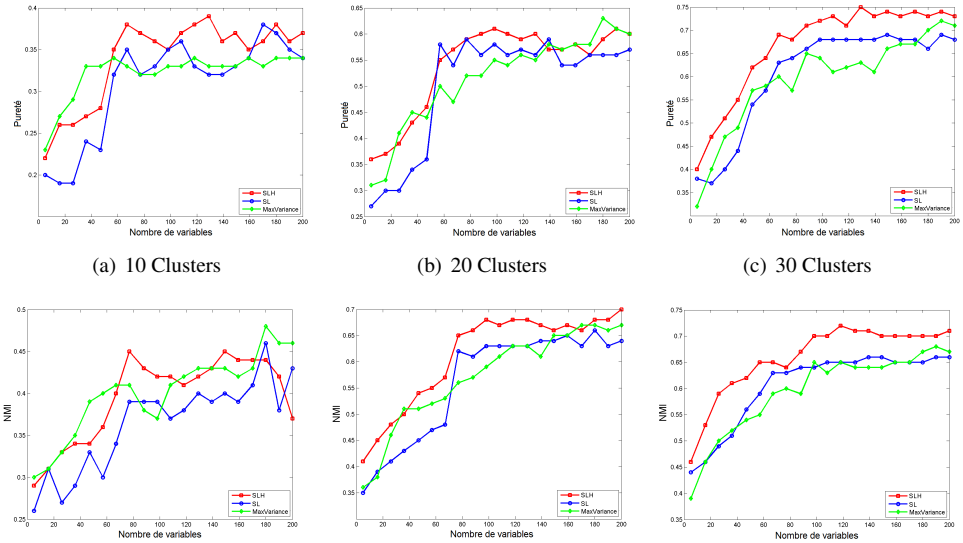
(a) 10 Clusters (b) 15 Clusters (c) 20 Clusters

(d) 10 Clusters (e) 15 Clusters (f) 20 Clusters

FIG. 2 – *Performance du clustering (Pureté et NMI) vs. le nombre de variables sélectionnées pour la base AR10P.*

	Pureté			NMI		
# Clusters	10	15	20	10	15	20
SLH	**0.392**	**0.415**	**0.500**	**0.419**	**0.382**	**0.448**
SL	0.292	0.453	0.500	0.329	0.355	0.459
MaxVariance	0.415	0.392	0.480	0.358	0.403	0.448
Toutes les variables	0.184	0.307	0.400	0.133	0.268	0.371

TAB. 2 – *Qualité du clustering utilisant 250 variables sélectionnées pour la base AR10P. La dernière ligne représente la performance obtenue sur toutes les variables (2400).*



(a) 10 Clusters (b) 20 Clusters (c) 30 Clusters

(d) 10 Clusters (e) 20 Clusters (f) 30 Clusters

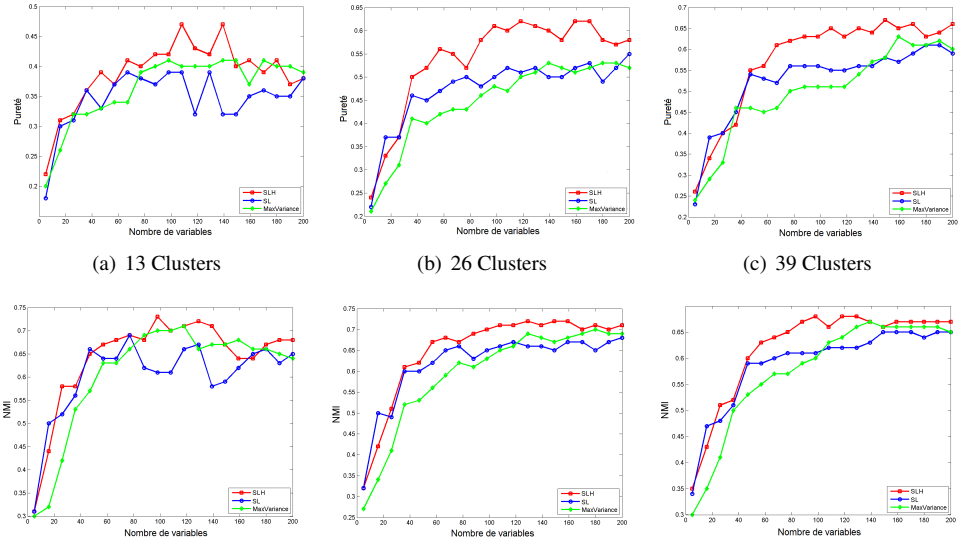
FIG. 3 – *Performance du clustering (Pureté et NMI) vs. le nombre de variables sélectionnées pour Coil20.*

	Pureté			NMI		
#Clusters	10	20	30	10	20	30
SLH	**0.346**	**0.608**	**0.722**	**0.425**	**0.680**	**0.695**
SL	0.348	0.576	0.683	0.390	0.628	0.640
MaxVariance	0.333	0.548	0.640	0.382	0.587	0.645
Toutes les variables	0.777	0.679	0.672	0.573	0.660	0.781

TAB. 3 – *Qualité du clustering utilisant 100 variables sélectionnées pour Coil20. La dernière ligne représente la performance obtenue sur toutes les variables (1024).*

	Pureté			NMI		
#Clusters	10	20	30	10	20	30
SLH	**0.423**	**0.606**	**0.626**	**0.726**	**0.702**	**0.675**
SL	0.385	0.499	0.559	0.606	0.646	0.610
MaxVariance	0.405	0.479	0.510	0.704	0.629	0.598
Toutes les variables	0.753	0.629	0.504	0.596	0.701	0.672

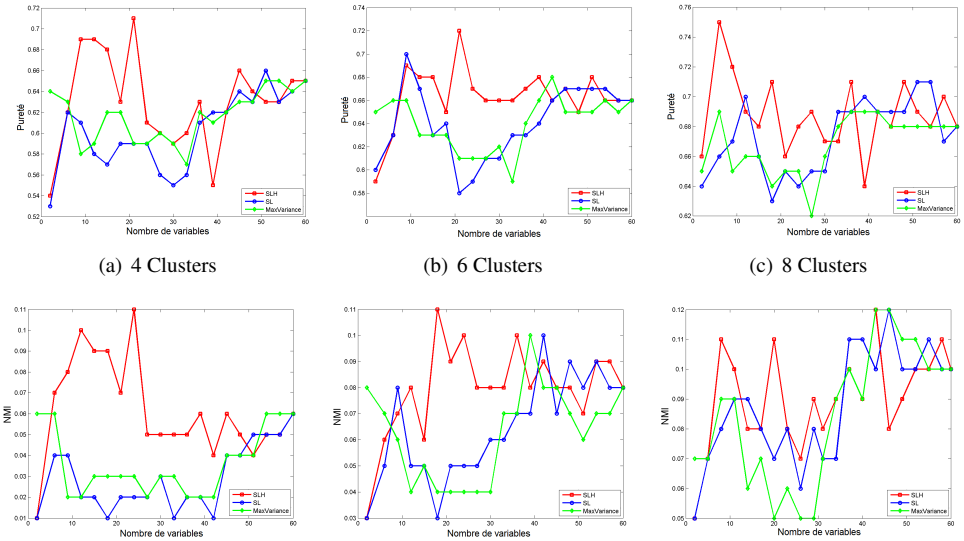
TAB. 4 – *Qualité du clustering utilisant 100 variables sélectionnées pour Isolet. La dernière ligne présente la performance utilisant toutes les variables (617).*



(a) 13 Clusters (b) 26 Clusters (c) 39 Clusters

(d) 13 Clusters (e) 26 Clusters (f) 39 Clusters

FIG. 4 – *Performance du clustering (Pureté et NMI) vs. Le nombre de variables sélectionnées pour Isolet.*



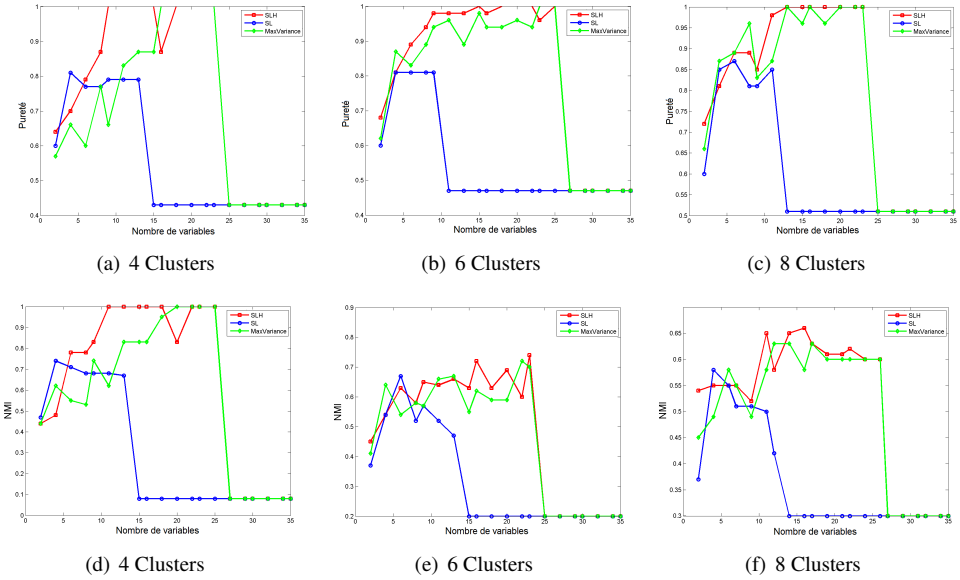
(a) 4 Clusters (b) 6 Clusters (c) 8 Clusters

(d) 4 Clusters (e) 6 Clusters (f) 8 Clusters

FIG. 5 – *Performance du clustering (Pureté et NMI) vs. Le nombre de variables sélectionnées pour Sonar.*

	Pureté			NMI		
# Clusters	4	6	8	4	6	8
SLH	**0.677**	**0.682**	**0.682**	**0.091**	**0.078**	**0.079**
SL	0.567	0.634	0.663	0.015	0.047	0.080
MaxVariance	0.620	0.625	0.663	0.031	0.046	0.070
Toutes les variables	0.649	0.658	0.682	0.055	0.080	0.097

TAB. 5 – *Qualité du clustering utilisant 15 variables sélectionnées pour Sonar. La dernière ligne présente la performance utilisant toutes les 60 variables.*



(a) 4 Clusters (b) 6 Clusters (c) 8 Clusters

(d) 4 Clusters (e) 6 Clusters (f) 8 Clusters

FIG. 6 – *Performance du clustering (pureté et NMI) vs. le nombre de variables sélectionnées pour Soybean.*

	Pureté			NMI		
# Clusters	4	6	8	4	6	8
SLH	**1**	**0.978**	**0.851**	**0.827**	**0.645**	**0.516**
SL	0.787	0.808	0.808	0.677	0.571	0.507
MaxVariance	0.659	0.936	0.829	0.737	0.573	0.497
Toutes les variables	0.425	0.468	0.510	0.079	0.196	0.303

TAB. 6 – *Qualité de clustering utilisant 9 variables sélectionnées pour Soybean. La dernière ligne présente la performance utilisant toutes les variables (35).*

3.2 Comparaison dans un cadre supervisé

Dans cette partie nous souhaitons évaluer les différents critères de sélection de variables en utilisant le classifieur 1-NN. Pour chaque donnée $\mathbf{x}_i$, nous cherchons le plus proche voisin $NN(\mathbf{x}_i)$.

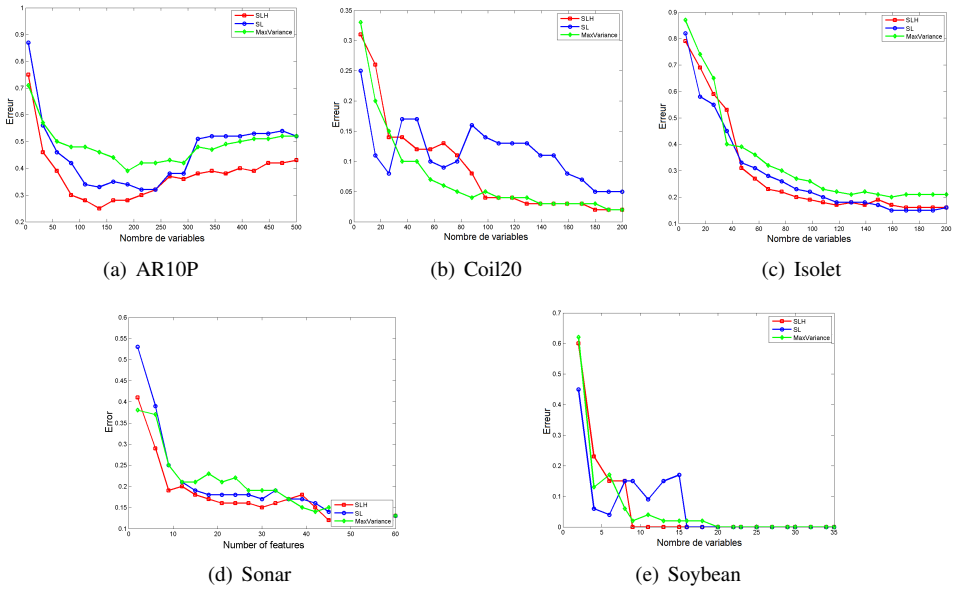
$$NN(\mathbf{x}_i) = argmin_{\forall j=1,..,N}(\mathbf{x}_i - \mathbf{x}_j)^2$$

Soit y'_i la classe de $\mathbf{x}_{NN(\mathbf{x}_i)}$ ($y'_i = NN(y_i)$). L'erreur de classification est calculée comme suivant :

$$Erreur = 1 - \frac{1}{N}\sum_{i=1}^{N}\delta(y'_i, y_i)$$

où $\delta(y'_i, y_i) = 1$ si $y'_i = y_i$ et 0 autrement.

La figure 7 montre les résultats de l'erreur de classification sur les cinq bases de données sélectionnées. De manière générale, les erreurs de classification de SLH sont meilleures que les erreurs du SL et de MaxVariance. Pour la base AR10P, la meilleure classification est obtenue en utilisant seulement 125 variables (Erreur = 0.246). Il est également intéressant de noter que pour la base Coil20, SLH obtient une bonne classification (*Erreur* = 0.043) en utilisant 100 variables. Pour la base Isolet, SLH converge aux alentours de 50 variables. Pour Sonar et Soybeans, SLH obtient de bons résultats en utilisant uniquement 9 variables. Le tableau 3.2 présente les erreurs de classification obtenues avec différentes valeurs de variables pour chaque base de test. On remarque que SLH produit des résultats comparables à ceux obtenus par une classification avec toutes les variables.



(a) AR10P (b) Coil20 (c) Isolet

(d) Sonar (e) Soybean

FIG. 7 – *Erreur de classification vs. le nombre de variables sélectionnées.*

# Variables	AR10P 250	Coil20 100	Isolet 100	Sonar 15	Soybean 9
SLH	**0.315**	**0.043**	**0.192**	**0.182**	**0**
SL	0.310	0.143	0.221	0.192	0.148
MaxVariance	0.415	0.045	0.258	0.256	0.02
Toutes les variables	0.500	0	0.124	0.125	0

TAB. 7 – *Erreur de classification en utilisant un nombre de variables. La dernière ligne présente la performance utilisant toutes les variables.*

4 Conclusions et perspectives

Etudier la sélection de variables en mode non supervisé est un vrai challenge pour la communauté scientifique en raison du manque d'informations sur les labels des données. Pour relever ce défi, nous avons proposé une approche autonome de sélection de variables nommée SLH, qui est une variante du score Laplacien et utilise la structure et la topologie de l'arbre défini par AntTree. Notre algorithme est autonome et ne nécessite aucun paramètre. Les résultats expérimentaux sur plusieurs jeux de données montrent que l'algorithme SLH réalise des performances plus élevées en mode supervisé et non supervisé. Comme perspective, nous nous sommes fixés comme objectif d'introduire de nouvelles contraintes non supervisées hiérarchiques pour la sélection de variables. L'idée est d'utiliser un autre type de lien dans le graphe qui représenterait des liens faibles.

Références

Azzag, H., N. Monmarché, M. Slimane, G. Venturini, et C. Guinot (2003). Anttree : a new model for clustering with artificial ants. In *IEEE CEC 2003*, Canberra, Australia, pp. 2642–2647.

Belkin, M. et P. Niyogi (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pp. 585–591. MIT Press.

Benabdeslem, K. et M. Hindawi (2011). Constrained laplacian score for semi-supervised feature selection. In *Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases - Volume Part I*, ECML PKDD'11, Berlin, Heidelberg, pp. 204–218. Springer-Verlag.

Cai, D., X. He, J. Han, et T. S. Huang (2011). Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Pattern Anal. Mach. Intell. 33*(8), 1548–1560.

Cai, D., C. Zhang, et X. He (2010). Unsupervised feature selection for multi-cluster data. In *16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'10)*.

Caruana, R. et D. Freitag (1994). Greedy attribute selection. In *ICML*, pp. 28–36.

Cover, T. M. et J. A. Thomas (2006). *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience.

Duda, R. O., P. E. Hart, et D. G. Stork (2000). *Pattern Classification (2nd Edition)*. Wiley-Interscience.

Frank, A. et A. Asuncion (2010). UCI machine learning repository.

He, X., D. Cai, et P. Niyogi (2005). Laplacian score for feature selection. In *Advances in Neural Information Processing Systems 18*.

John, G. H., R. Kohavi, et K. Pfleger (1994). Irrelevant Features and the Subset Selection Problem. In *International Conference on Machine Learning*, pp. 121–129.

Kalakech, M., P. Biela, L. Macaire, et D. Hamad (2011). Constraint scores for semi-supervised feature selection : A comparative study. *Pattern Recogn. Lett. 32*(5), 656–665.

Kohavi, R. et G. H. John (1997). Wrappers for Feature Subset Selection. *Artificial Intelligence 97*(1-2), 273–324.

Koller, D. et M. Sahami (1996). Toward optimal feature selection. In *ICML*, pp. 284–292.

Rodgers, J. L. et A. W. Nicewander (1988). Thirteen Ways to Look at the Correlation Coefficient. *The American Statistician 42*(1), 59–66.

Roweis, S. T. et L. K. Saul (2000). Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE 290*, 2323–2326.

Strehl, A., J. Ghosh, et C. Cardie (2002). Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research 3*, 583–617.

Tenenbaum, J. B., V. de Silva, et J. C. Langford (2000). A global geometric framework for nonlinear dimensionality reduction. *Science (New York, N.Y.) 290*(5500), 2319–2323.

Yu, L. et H. Liu (2003). Feature selection for high-dimensional data : A fast correlation-based filter solution. In *in ICML*, pp. 856–863.

Zhang, D. et Z. hua Zhou Songcan Chen (2007). Semi-supervised dimensionality reduction. In *In : Proceedings of the 7th SIAM International Conference on Data Mining*, pp. 11–393.

Zhang, M.-L., J. M. Peña, et V. Robles (2009). Feature selection for multi-label naive bayes classification. *Inf. Sci. 179*(19), 3218–3229.

Summary

In this paper, we address the problem of unsupervised feature selection, which is an important challenge due to the absence of class labels that would guide the search for relevant information. Thus, we define a new Laplacian score by constraining the Laplacian score using tree topology structure. The main interest to use tree structure is to automatically discover local structure and local neighbors. Experimental results and comparison studies over various data sets have demonstrated the effectiveness of the proposed algorithm in clustering and classification applications.