SSC : Statistical Subspace Clustering

Laurent Candillier[1,2], Isabelle Tellier[1], Fabien Torre[1], Olivier Bousquet[2]

[1] GRAppA - Université Charles de Gaulle - Lille 3
candillier@grappa.univ-lille3.fr
http ://www.grappa.univ-lille3.fr
[2] Pertinence - 32 rue des Jeûneurs -75002 Paris
olivier.bousquet@pertinence.com
http ://www.pertinence.com

Résumé. Cet article se place dans le cadre du *subspace clustering*, dont la problématique est double : identifier simultanément les clusters et le *sous-espace spécifique* dans lequel *chacun* est défini, et caractériser chaque cluster par un nombre minimal de dimensions, permettant ainsi une présentation des résultats compréhensible par un expert du domaine d'application.

Les méthodes proposées jusqu'à présent pour cette tâche ont le défaut de se restreindre à un cadre numérique. L'objectif de cet article est de proposer un algorithme de *subspace clustering* capable de traiter des données décrites à la fois par des attributs continus et des attributs catégoriels.

Nous présentons une méthode basée sur l'algorithme classique EM mais opérant sur un modèle simplifié des données et suivi d'une technique originale de sélection d'attributs pour ne garder que les dimensions pertinentes de chaque cluster. Les expérimentations présentées ensuite, menées sur des bases de données aussi bien artificielles que réelles, montrent que notre algorithme présente des résultats robustes en termes de qualité de la classification et de compréhensibilité des clusters obtenus.

Introduction

Face aux quantités d'informations qui ne cessent d'augmenter dans les bases de données du monde entier, l'extraction automatique de connaissances à partir de ces bases et les techniques de visualisation des résultats sont devenues indispensables. C'est la raison d'être de la *fouille de données*. Dans ce cadre, l'apprentissage non supervisé (ou *clustering*) est depuis longtemps utilisé pour identifier les groupes (ou *clusters*) d'éléments similaires (cf. survey de Berkhin 2002). Une problématique supplémentaire apparaît face à des bases de données de grande dimensionnalité : dans ce cas, les groupes peuvent être caractérisés uniquement par certains sous-ensembles de dimensions et ces dimensions pertinentes peuvent être différentes d'un groupe à l'autre. Sur de tels problèmes, les techniques classiques de *clustering* fonctionnent mal car, fondées sur une distance entre objets définie globalement dans l'espace de description, elles ne peuvent pas appréhender le fait que la notion de similarité varie d'un groupe à l'autre.

Une nouvelle problématique a donc émergé récemment, celle du *subspace clustering*, dont l'enjeu est de cibler les groupes d'objets et, pour chacun, le *sous-espace spécifique* dans

lequel il est défini [1]. Et cet objectif s'accompagne d'un second : celui de fournir une description compréhensible des groupes identifiés. Les méthodes proposées pour cette tâche se sont focalisées sur le premier objectif et ont négligé le second. De plus, la partie expérimentale de ces travaux porte exclusivement sur des données numériques.

L'objectif de cet article est de proposer un algorithme de *subspace clustering* capable de traiter des données décrites à la fois par des attributs continus et des attributs catégoriels, demandant à l'utilisateur de régler le moins de paramètres possible, et fournissant en sortie une représentation simple des clusters identifiés. Nous nous basons pour cela sur l'algorithme EM adapté au *clustering* (Ye et al. 2003). Nous en proposons une version simplifiée en ajoutant l'hypothèse que les données sont générées selon des distributions indépendantes sur chaque dimension. Ceci nous permet d'en dériver une théorie compréhensible sous forme de règles puisque chaque dimension est caractérisée indépendemment des autres. La suite de l'article est organisée comme suit : dans la section 1, nous présentons notre méthode de *subspace clustering* ; les expérimentations présentées ensuite dans la section 2, menées sur des bases de données aussi bien artificielles que réelles, présentent les résultats de notre algorithme ; et nous terminons dans la section 3 par quelques conclusions et perspectives ouvertes par ce travail.

1 Algorithme SSC

Dans (Parsons et al. 2004), les auteurs ont étudié et comparé les méthodes existantes de *subspace clustering*. Toutes sont capables de retrouver efficacement les clusters et leur sous-espace spécifique, mais elles nécessitent souvent des paramètres difficiles à régler par l'utilisateur et influant sur leurs performances (seuil de densité, nombre moyen de dimensions caractéristiques des clusters, distance minimale entre clusters, etc.) De plus, tous les tests ont été effectués sur des bases de données exclusivement numériques. Enfin, aucune proposition aboutie de présentation simple des résultats n'a été effectuée. Ce point est pourtant crucial car même si la dimensionnalité des clusters a été réduite dans les sous-espaces qui leur sont propres, celle-ci peut encore être trop élevée pour qu'un expert du domaine d'application puisse appréhender le résultat. Or il est souvent possible d'ignorer certaines de ces dimensions, tout en conservant le même partitionnement des objets.

1.1 Modèle probabiliste

Afin de fournir en sortie de notre algorithme de *subspace clustering* une description simple des clusters trouvés, nous choisissons de les représenter sous forme de règles (hypercubes dans des sous-espaces de l'espace original), représentation reconnue comme facilement interprétable. Pour intégrer cette contrainte dans l'algorithme EM classique, nous proposons d'ajouter l'hypothèse que les données sont générées selon des distributions indépendantes sur chaque dimension. Cette hypothèse a comme effet d'affaiblir le modèle classique, prenant également en compte les corrélations possibles entre dimensions, mais ainsi, la modélisation est adaptée à la présentation sous forme de règles des clusters trouvés

1. la différence avec la problématique de la sélection d'attributs (ou *feature selection*) est que le sous-espace ciblé est local à chaque cluster, et non global à tous (Parsons et al. 2004).

car chaque dimension est caractérisée indépendemment des autres. De plus, l'algorithme est alors plus rapide que l'algorithme classique, car le nouveau modèle nécessite moins de paramètres ($O(M)$ au lieu du $O(M^2)$ classique, pour M le nombre de dimensions), et les opérations matricielles sont évitées.

Dans notre modèle, nous supposons que les données ont été générées selon des distributions gaussiennes sur les dimensions continues et selon des distributions multinomiales sur les dimensions discrètes. Le modèle est donc composé des paramètres suivants pour chaque cluster C_k : son poids W_k ; pour chaque dimension d continue, sa moyenne μ_{kd} et sa variance σ_{kd} ; et pour chaque dimension d discrète, les fréquences de chaque modalité $Freqs_{kd}$. Il suppose la donnée du nombre K de clusters recherchés.

1.2 Algorithme

Nous utilisons l'algorithme EM classique sur notre modèle. Les paramètres cachés du modèle correspondent aux probabilités d'appartenance de chaque objet à chaque cluster. Dans notre cas, les dimensions étant supposées indépendantes, la probabilité d'appartenance $P(\overrightarrow{X_i}|C_k)$ d'un objet $\overrightarrow{X_i}$ à un cluster C_k correspond au produit des probabilités $P(X_{id}|C_{kd})$ sur chaque dimension d : $\frac{1}{\sqrt{2\pi}\sigma_{kd}} exp(-\frac{1}{2}(\frac{X_{id}-\mu_{kd}}{\sigma_{kd}})^2)$ si d est continue, et $Freqs_{kd}(X_{id})$ si d est discrète. Et pour éviter qu'une probabilité nulle sur une dimension n'annule la probabilité globale, nous utilisons une constante positive très faible ϵ qui constitue une borne minimale sur les probabilités $P(X_{id}|C_{kd})$.

L'algorithme EM est connu pour converger lentement dans certains cas. Pour l'accélérer, nous proposons d'ajouter l'heuristique suivante : s'arrêter lorsque les attributions de clusters aux objets ne changent pas. Ce critère d'arrêt ressemble alors fortement au critère d'arrêt de K-means. À chaque itération, il faut donc également évaluer les attributions de clusters aux objets comme suit : $Cluster(\overrightarrow{X_i}) = ArgMax_k P(\overrightarrow{X_i}|C_k)$. Finalement, l'algorithme est relancé un certain nombre de fois avec des solutions initiales aléatoires. Puis la partition maximisant la fonction $E = \sum_i log(P(\overrightarrow{X_i}))$ est conservée.

1.3 Présentation du résultat

Afin que le résultat soit le plus compréhensible possible, nous souhaitons nous donner une seconde vue sur chaque cluster, correspondant à sa représentation simplifiée sous forme de règle, chacune décrite par le moins de dimensions possible. Dans un premier temps, chaque cluster est représenté par l'intervalle minimum contenant l'ensemble des valeurs des objets inclus dans le cluster sur les dimensions continues, et par la modalité la plus probable sur les dimensions discrètes. Ensuite, le support de la règle est calculé (l'ensemble des objets compris dans la règle). Puis un poids W_{kd} est attribué à chacune des dimensions d du cluster C_k, en fonction de la dispersion relative des objets sur la dimension. Pour les dimensions continues, il s'agit du rapport entre variance locale et variance globale par rapport à μ_{kd} (N correspond au nombre d'objets de la base). Et pour les dimensions discrètes, il s'agit de la fréquence relative de la modalité la plus probable ($Modalites_d$ correspond à l'ensemble des modalités possibles sur la dimension d, et $Frequences_d$ à l'ensemble des fréquences de chacune de ces modalités sur l'ensemble de la base).

$$W_{kd} = \begin{cases} 1 - \frac{\sigma_{kd}^2}{\sigma_d^2} \text{ , avec } \sigma_d^2 = \frac{\sum_i (X_{id} - \mu_{kd})^2}{N} & \text{si } d \text{ continue} \\ \frac{Freqs_{kd}(mod) - Frequences_d(mod)}{1 - Frequences_d(mod)} & \text{si } d \text{ discrète} \\ \text{avec } mod = ArgMax_{\{m \in Modalites_d\}} Freqs_{kd}(m) \end{cases}$$

Puis la sélection des dimensions pertinentes s'effectue comme suit : pour toutes les dimensions, présentées dans l'ordre croissant de leur poids, supprimer la dimension si sa suppression ne modifie pas le support de la règle.

Enfin, afin de visualiser graphiquement les résultats obtenus, nous proposons de calculer un poids associé à chaque couple de dimensions présentes dans la description des clusters : $V_{ij} = \sum_k max(W_{ki}, W_{kj})$. Plus ce poids est important, plus les règles projetées sur ces deux dimensions sont spécifiques.

2 Expérimentations

2.1 Tests sur données artificielles

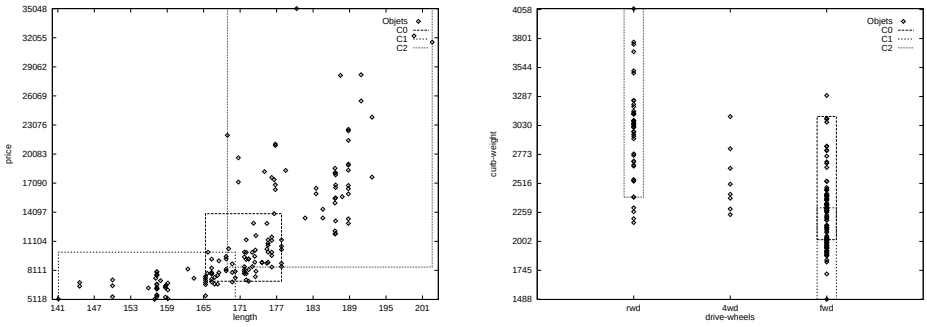
Afin de nous comparer aux méthodes existantes de *subspace clustering*, nous proposons de mener ces expériences sur des bases uniquement numériques. Parmi les plus récentes, LAC de (Domeniconi et al. 2004) est une méthode efficace qui, comme la nôtre, nécessite un seul paramètre utilisateur : le nombre de clusters recherchés. Nous proposons de nous comparer à cet algorithme et utilisons des bases artificielles pour évaluer les taux d'erreurs de notre algorithme et de LAC en classification. À chaque partition est associée la pureté moyenne des clusters produits (la pureté correspond au pourcentage maximum d'objets du cluster qui appartiennent au même concept initial).

K points d'ancrage $(\overrightarrow{O_1}, ..., \overrightarrow{O_K})$ sont tirés aléatoirement dans l'espace de description à M dimensions, et sont utilisés comme centroïdes des clusters $(C_1, ..., C_K)$ à générer. À chacun de ces clusters est associée une partie des N objets, et un sous-ensemble des M dimensions constituant ses dimensions caractéristiques. Puis les coordonnées des objets appartenant à un cluster C_k sont générées selon une loi normale de centre O_{kd} et d'écart type e_k sur toute dimension d caractéristique de C_k ; elles sont générées selon une loi uniforme dans l'espace de description des dimensions non caractéristiques.

Les expériences menées en faisant varier les paramètres de génération des bases artificielles ont mis en avant la robustesse de notre méthode. En particulier, elle se révèle efficace pour faire face au bruit existant dans les données (la pureté moyenne des clusters est de 90% pour 20% de bruit dans la base, contre 70% pour LAC). Concernant le temps d'exécution de la méthode, l'heuristique que nous avons proposée permet d'obtenir, pour des résultats de qualité similaire, des temps de calcul plus proches de ceux de K-means (connu pour sa rapidité) que de ceux de EM. Concernant le seul paramètre de l'algorithme, le nombre de clusters recherchés, si celui-ci est inférieur au nombre réel de clusters, alors quelques concepts sont fusionnés mais le résultat ne s'éloigne pas complètement de la solution réelle. S'il est supérieur au nombre réel, alors plusieurs concepts se recouvrent. Enfin, notons que les résultats de notre algorithme sont tout aussi robustes si les données ont été générées selon des lois uniformes dans les intervalles de définition de leurs dimensions caractéristiques, au lieu de lois gaussiennes.

2.2 Tests sur données réelles

Des expériences ont également été menées sur des bases de données réelles. Parmi elles, la base *Automobile*, issue des bases de données de l'UCI (Blake et Merz 1998), contient la description (numérique et catégorielle) d'un ensemble de voitures. Sur cette base, les visualisations graphiques correspondant à deux couples de dimensions de poids maximum sont fournies figure 1. L'algorithme met ainsi en avant que le prix des voitures augmente fortement lorsque leur longueur dépasse les 170 (figure 1(a)), que les voitures ayant une traction arrière (*rwd*) ont un poids à vide supérieur aux tractions avant et 4 roues motrices (figure 1(b)), et que la majorité des voitures les plus chères sont à traction arrière (correspondance entre les deux figures concernant le cluster C_2). Pour plus de détails sur les expérimentations, voir (Candillier et al. 2005).



(a) Projections sur *longueur* et *prix*. (b) Projections sur *traction* et *poids*.

FIGURE 1 – Résultats de SSC sur la base *Automobile*, pour $K = 3$.

3 Conclusions et perspectives

Nous avons présenté dans cet article une nouvelle méthode de *subspace clustering* basée sur l'algorithme EM en ajoutant l'hypothèse que les données ont été générées selon des distributions indépendantes sur chaque dimension. Cette idée a déjà été étudiée dans (Pelleg et Moore 2000). Il existe plusieurs différences entre notre méthode et la leur. La première apparaît dans la modélisation : au lieu de supposer la distribution gaussienne sur une dimension continue, les auteurs la supposent uniforme à l'intérieur d'un intervalle donné, et utilisent une *queue* de distribution aux bords de cet intervalle, dépendant d'un paramètre σ qui évolue au cours de l'algorithme. Cette différence se retrouve ensuite dans la méthode finale de clustering. En particulier, leur méthode n'est pas capable de mettre à jour son modèle de façon incrémentale, alors que la nôtre peut s'adapter à la présentation de nouveaux exemples. De plus, nous avons intégré effectivement la problématique catégorielle qui n'était évoquée qu'à titre de perspectives dans l'article et nous avons proposé une méthode originale de sélection d'attributs permettant de fournir en sortie un résultat compréhensible et visuel des clusters identifiés.

Nous avons également défini une heuristique originale pour accélérer l'algorithme. Pour poursuivre la recherche dans ce sens, il semble intéressant de s'inspirer de l'article de (Bradley et al. 1998) qui traite de l'accélération de l'algorithme EM dans le cas général. Une autre piste possible est d'éviter de considérer toutes les dimensions au cours de l'algorithme, en ne sélectionnant que les dimensions de poids maximum.

Notons enfin que notre méthode nécessite la donnée d'un paramètre de la part de l'utilisateur : K, le nombre de clusters recherchés. Une amélioration possible consisterait à identifier automatiquement ce paramètre. Pour cela, il est classique d'utiliser le critère BIC (Ye et al. 2003). Dans notre cas, une autre piste originale serait d'utiliser le fait que lorsque K est supérieur au nombre réel de clusters recherchés, alors les règles associées aux clusters se chevauchent.

Références

Berkhin P. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, California, 2002.

Blake C.L. et Merz C.J. (1998). UCI Repository of machine learning databases [http ://www.ics.uci.edu/∼mlearn/MLRepository.html].

Bradley P., Fayyad U., et Reina C. Scaling EM (Expectation-Maximization) Clustering to Large Databases. Microsoft Research Report, MSR-TR-98-35, Aug. 1998.

Candillier L., Tellier I., Torre F. et Bousquet O. SSC : Statistical Subspace Clustering. Rapport technique GRAppA, 2005. [http ://grappa.univ-lille3.fr/∼candillier/publis].

Domeniconi C., Papadopoulos D., Gunopolos D. et Ma S. Subspace clustering of high dimensional data. In *SIAM Int. Conf. on Data Mining*, 2004.

Parsons L., Haque E. et Liu H. Evaluating subspace clustering algorithms. In *Workshop on Clustering High Dimensional Data and its Applications, SIAM Int. Conf. on Data Mining*, pp 48-56, 2004.

Pelleg D. et Moore A. Mixtures of rectangles : Interpretable soft clustering. In C. Brodley and A. Danyluk editors, ICML 2001, pp 401-408.

Ye L. et Spetsakis M.E. Clustering on Unobserved Data using Mixture of Gaussians. Technical Report, York University, Oct. 2003.

Summary

In this paper, we focus on the task of *subspace clustering*, that has two goals : simultaneously identify the clusters and the *subspaces* in which *each* of them is defined, and describe each cluster with as few dimensions as possible, so that the results are easily interpretable by a human user.

One default of existing methods is that they only consider numerical databases. The aim of this paper is to propose a new *subspace clustering* algorithm, able to tackle databases that may contain continuous as well as discrete attributes.

We present a method based on the classical EM algorithm, but applied to a simplified model, and followed by an original technique of feature selection that only keeps dimensions that are relevant to each cluster.

Experiments, conducted on artifical as well as real databases, show that our algorithm gives robust results, in terms of classification and interpretability of the output.