SNOW, un algorithme exploratoire pour le subspace clustering

Sylvain Dormieu *, Nicolas Labroche *

* UPMC Univ Paris 06, UMR 7606, LIP6
4 place Jussieu, 75005 Paris, France
sylvain.dormieu@gmail.com,
nicolas.labroche@lip6.fr

Résumé. Cet article propose un nouvel algorithme pour le problème de subspace clustering dénommé SNOW. Contrairement aux approches descendantes classiques, il ne repose pas sur l'hypothèse de localité et permet l'affectation d'une donnée à plusieurs clusters dans des sous-espaces différents. Les expérimentations préliminaires montrent que notre approche obtient de meilleurs résultats que l'algorithme COPAC sur une base de référence et a été appliquée sur une base de données réelles.

1 Introduction

Les méthodes de classification non supervisée - ou clustering - classiques synthétisent l'information en construisant des groupes de données. Ces données sont le plus souvent définies par un ensemble d'attributs et les clusters résultants sont donc déterminés également dans l'espace des attributs. Plusieurs méthodes comme la pondération ou la sélection d'attributs, ou des métriques adaptées permettent de modifier, limiter ou de supprimer l'influence de certains attributs, mais l'ensemble des clusters est généralement défini dans un même espace. Cependant, certains groupes peuvent n'être pertinents que dans un sous-ensemble des attributs. Ce sous-ensemble d'attributs caractéristiques est appelé le *sous-espace* du cluster. Une donnée peut donc appartenir à plusieurs clusters définis dans des sous-espaces différents. Comme indiqué par (Kriegel et al., 2009), l'objectif des méthodes de subspace clustering est de découvrir tous les clusters dans tous les sous-espaces.

Par exemple, pour des données représentant des objets de différentes formes et de différentes couleurs, il est possible de déterminer plusieurs sous-espaces évidents basés sur la couleur, ou bien sur la forme ou enfin sur les deux attributs à la fois. Dans cet exemple, un carré rouge devrait pouvoir appartenir à la fois au cluster *rouge* si le sous-espace est limité à l'attribut de couleur et au cluster *carré* s'il est limité au sous-espace de forme.

Ce papier est organisé comme suit : la section 2 rappelle les principaux travaux conduits dans le domaine du subspace clustering. La section 3 décrit l'algorithme SNOW. La section 4 présente des résultats comparatifs avec l'algorithme COPAC sur des données artificielles et illustre les résultats de Snow sur des données réelles issues du UCI Machine Learning Repository. La section 5 conclut l'article et présente les perspectives.

2 Travaux existants

Le subspace clustering est un domaine assez récent (Parsons et al., 2004), (Kriegel et al., 2009) qui vise à déterminer conjointement les clusters et leurs sous-espaces associés. Contrairement aux approches classiques de clustering dans lesquelles, la phase de partitionnement peut être précédée d'une phase de sélection ou de pondération des attributs, le subspace clustering ne dissocie pas la définition de l'espace et celle du groupe de données. En conséquence, une donnée peut théoriquement appartenir à plusieurs clusters, dès lors que ceux-ci sont définis dans un sous-espace qui leur est propre.

Le subspace clustering a été défini dans deux principaux travaux (Parsons et al., 2004), (Kriegel et al., 2009) qui clarifient la terminologie et distinguent le subspace clustering d'autres domaines proches comme le biclustering et le coclustering. On distingue plusieurs méthodes de subspace clustering en fonction du mécanisme de sélection des attributs lors de la construction des clusters. Certains algorithmes reposent sur des mécanismes de pondération des attributs, d'autres recherchent tous les sous-espaces potentiels de manière ascendante (des espaces de 1 dimension vers l'espace contenant toutes les dimensions) ou inversement descendante.

Les méthodes qui reposent sur un mécanisme de sélection/pondération des attributs appartiennent au domaine du *soft subspace clustering* (Gustafson et Kessel, 1979), (Candillier et al., 2005). L'idée principale de ces méthodes est d'affecter un poids à chaque attribut et d'utiliser une optimisation alternée pour rechercher un maximum local à une fonction objectif. Il existe toutefois de nombreuses limitations à ces méthodes : définition du nombre de clusters a priori, pas de garantie d'une convergence vers un optimum global, affectation (éventuellement floue) de chaque donnée à un (ou plusieurs) cluster(s) défini(s) dans un unique sous-espace.

D'autres approches reposent sur une exploration systématique de tous les sous-espaces éligibles en partant des sous espaces les plus petits. Ces approches *ascendantes* sont basées sur un mécanisme de recherche d'itemsets fréquents. Par exemple, l'algorithme CLIQUE (Agrawal et al., 1998) intègre un mécanisme d'agrégation de sous-ensembles denses de basse dimensionnalité pour retrouver les sous-ensembles denses de plus haute dimensionnalité. Toutefois, la complexité de ce type d'algorithme est grande par rapport au nombre d'attributs.

À l'inverse des méthodes ascendantes, les méthodes *descendantes* commencent par étudier l'ensemble des attributs avant de déterminer et de sélectionner les attributs caractéristiques pour réduire le nombre de dimensions. Ce type d'algorithme est efficace lorsque la répartition des données vérifie l'hypothèse de localité définie dans Kriegel et al. (2009) : "une sélection locale des données suffit à estimer une orientation locale des données".

Cette définition de localité repose sur des calculs de type k plus proches voisins qui utilisent l'ensemble des attributs pour définir le voisinage local. Cette hypothèse ne semble pas pertinente dans la pratique car, dans le cas d'un espace de grande dimension, de nombreux attributs non caractéristiques affectent le calcul du voisinage et donc le choix des attributs caractéristiques. De nombreux algorithmes utilisent l'heuristique des k plus proches voisins (Achtert et al., 2007), (Friedman et Meulman, 2004). Plusieurs paramètres sont estimés localement pour chaque cluster comme l'orientation du voisinage, et utilisés ensuite pour agréger au cluster les données vérifiant une relation de proximité. Toutefois, comme précédemment, ces algorithmes affectent une donnée à un unique cluster et son sous-espace.

Enfin, l'algorithme CASH (Achtert et al., 2008) diffère des approches descendantes précédentes car il ne repose pas sur l'hypothèse de localité. Dans ce modèle, les clusters sont modélisés par des hyperplans. L'espace des hyperplans contenant au minimum une donnée est divisé

en grille et parcouru afin de déterminer quels sont les hyperplans contenant de nombreuses données. La réitération de ce calcul et la modélisation en hyperplan permet de construire les sous-espaces. Cette méthode possède cependant une complexité rédhibitoire.

Nous décrivons dans la section suivante le modèle de l'algorithme SNOW qui, comme l'algorithme CASH ne repose pas sur l'hypothèse de localité et possède une complexité moindre.

3 Algorithme Snow

SNOW est un algorithme qui détermine à chacune de ses itérations un cluster et son sous-espace associé. Chaque itération est indépendante des précédentes et repose sur un processus en 4 étapes principales : (1) la génération aléatoire d'un cluster potentiel ; (2) la détermination de l'hyper-cube propre à ce cluster potentiel ; (3) le calcul d'un pas de densité des données pour chacun des attributs ; (4) l'extension de l'hyper-cube à partir du pas de densité pour obtenir un cluster maximal.

Génération aléatoire d'un cluster potentiel. Contrairement aux approches de l'état de l'art qui déterminent les clusters à partir du voisinage d'une seule donnée, notre approche se base sur une sélection aléatoire de plusieurs données pour former la graine du premier cluster potentiel. Cette sélection de plusieurs données amène plus de robustesse dans la détermination des attributs caractéristiques du cluster car, contrairement au voisinage local, elle permet de considérer des plages de valeurs plus importantes et d'être donc moins sensible aux variations locales de densité des attributs ou aux points aberrants. Enfin, à chaque itération, la distribution aléatoire initiale des points favorise l'émergence d'attributs caractéristiques différents ce qui assure une bonne couverture de l'espace des solutions. Comme CASH, SNOW recherche un modèle reliant les données et non les données mutuellement proches.

Détermination de l'hyper-cube du cluster potentiel. On définit l'hypercube $\mathcal{H}_C$ du cluster potentiel C comme le produit des intervalles $\mathcal{I}_j^C$ sur chacun des attributs j de l'espace initial $\mathcal{R}^m$. Chaque intervalle $\mathcal{I}_j^C$ sur l'attribut j pour le cluster C est défini comme l'intervalle minimal englobant l'ensemble des valeurs des points $x \in C$ sur l'attribut j :

$$\mathcal{H}_C = \prod_{j\in[1,m]} \left[\min_{x\in C} x_j ; \max_{x\in C} x_j\right] \text{ aussi noté } \prod_{j\in[1,m]} [l_j, h_j] \tag{1}$$

où x_j désigne la valeur de l'attribut j du point x. À cette étape, on ajoute au cluster potentiel l'ensemble des données contenues dans l'hypercube.

Calcul d'un pas de densité. Cette étape vise à déterminer une *densité locale* au cluster potentiel. Pour chaque attribut j, on définit la séquence $S_{\mathcal{I}_j^C}$ comme l'ensemble ordonné des valeurs x_j de l'attribut j pour tout $x \in C$ sur l'intervalle $\mathcal{I}_j^C$. Le pas de densité δ_j est ensuite simplement défini comme la distance maximale observée sur l'attribut j entre deux valeurs consécutives de $S_{\mathcal{I}_j^C}$ (s_i désigne le $i^{\text{ème}}$ élément de la séquence $S_{\mathcal{I}_j^C}$) :

$$\delta_j = \max_{i\in[1,|S_{\mathcal{I}_j^C}|-1]} s_{i+1} - s_i \tag{2}$$

Détermination du cluster et de l'hypercube maximal. Dès lors que le pas δ_j est déterminé pour tout attribut $j \in [1, m]$, notre algorithme agrège itérativement au cluster potentiel les points dont les coordonnées sont situées à une distance inférieure à δ_j des frontières de son hypercube $\mathcal{H}_C$ pour tous les attributs j. L'hypercube associé au cluster est ensuite mis à jour et le processus d'agrégation de nouveaux points est réitéré jusqu'à ce qu'aucun candidat ne puisse plus être ajouté au cluster, qui est alors maximal.

Paramétrage. L'algorithme SNOW repose sur deux paramètres fixés par l'utilisateur. Le premier est le nombre maximal d'itérations τ. Il permet d'optimiser la couverture, la qualité des clusters et de leurs sous-espaces associés par rapport au temps de calcul.

Le second paramètre k est le nombre de données sélectionnées pour générer les graines de clusters potentiels. Une petite valeur de k diminue le temps de calcul mais une plus grande valeur de k permet une meilleure estimation de la densité des attributs du cluster potentiel et donc d'obtenir de meilleures performances en conjonction avec le paramètre τ.

4 Expérimentations

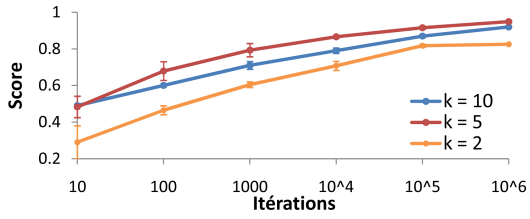
Cette section présente les deux expérimentations conduites pour valider notre algorithme. La première expérimentation propose la comparaison de notre algorithme SNOW avec la méthode COPAC (Achtert et al., 2007) à l'aide d'une mesure moyenne des F1-mesures des clusters découverts les plus pertinents. Nous utilisons l'implémentation de COPAC fournie par le framework ELKI (Achtert et al., 2008). 2 bases de données sont utilisées : (1) un premier jeu de données artificiel de référence nommé $Parsons$ (Parsons et al., 2004), dont les clusters rapprochés peuvent poser problème aux méthodes comme COPAC reposant sur l'hypothèse de localité ; (2) le même jeu de données modifié en séparant plus les clusters afin qu'il soit plus favorable à la méthode COPAC reposant sur l'hypothèse de localité.

La seconde expérimentation vise à évaluer la capacité de la méthode SNOW à produire des clusters interprétables sur la base de données réelles nommée Auto MPG (Quinlan, 1993).

Résultats comparatifs sur la base $Parsons$. Nous utilisons une méthode d'évaluation, qui consiste, à partir d'un ensemble de clusters cibles connus, à intégrer dans le score total chaque meilleur cluster généré par rapport à chaque cluster cible. Soit $C_1, \ldots, C_p$ la liste des clusters générés par l'algorithme et $S_1, \ldots, S_q$ les clusters cibles. Pour chaque cluster cible S_i , on note C_j le cluster ayant le meilleur score F1 par rapport à S_i. Nous proposons comme score général de l'algorithme la moyenne des meilleurs scores $F1$ par rapport à chaque cluster cible S_i :

$$\text{Score} = \frac{\sum_i F1(C_j, S_i)}{q}$$

Discussion des résultats sur la base $Parsons$ et $Parsons$ modifiée. Les meilleurs résultats obtenus par l'approche COPAC lors de nos expérimentations sont de 49,84%. Ce résultat est dû à la proximité des clusters 2 à 2. En effet, le calcul du voisinage inclut un attribut dans le sous-espace, et par conséquent le calcul de l'orientation du voisinage intègre des données de clusters différents, faussant le résultat final. SNOW, quand à lui, n'est pas sensible à l'hypothèse



	COPAC	SNOW
Parsons	49.84%	94.84%
Parsons modifié	93.16%	97.4%

FIG. 1 – *Score de* SNOW *sur* $Parsons$ *en fonction de* τ *et de* k *et par rapport à COPAC.*

	MPG	#Cylindres	Volume depl.	Puissance	Poids	Acc.	Année	Origine
Cluster 1	22.3 - 46.6	4	85-140	52 - 113	1975 - 2890	11.6 - 22.2	70 - 82	1 - 3
Cluster 2	11 - 22	6 - 8	171 - 400	97 - 210	2984 - 4498	12 - 19.2	70 - 79	1

TAB. 1 – *2 des clusters générés par* SNOW *sur la base Auto-MPG.*

de localité. Pour $k = 5$ et $\tau = 10^5$ (i.e. quand on génère 10^5 clusters candidats aléatoirement), le score est de 94.84%.

Pour $Parsons$ modifiée, COPAC obtient un score de 93.16%. La proximité des clusters est donc bien la cause de la mauvaise performance de COPAC sur le premier jeu de données. SNOW obtient un score légèrement supérieur à COPAC, de 97.4%.

Jeu de données réelles Auto MPG. Contrairement aux données de $Parsons$, il n'existe pas d'étiquettes de clusters théoriques permettant une évaluation objective des résultats. Nous proposons donc, à l'image de la démarche suivie dans (Candillier et al., 2005), d'étudier avec cette base réelle la pertinence et l'interprétabilité des clusters découverts. SNOW produisant un grand nombre de clusters, nous avons retenu expérimentalement deux clusters parmi les plus denses dans leur sous-espace pour conduire notre interprétation. La densité des clusters a été calculée à partir du nombre de données du cluster divisé par le volume du cluster (la longueur des intervalles de l'hypercube est bornée pour ce calcul au minimum à 0.1). SNOW a été lancé avec comme valeurs de paramètres $\tau = 30000$ et $k = 10$. Les clusters ayant un effectif inférieur à 50 sont supprimés et les clusters restants sont triés par ordre de densité décroissante. Les 10 premiers clusters de ce classement sont relativement homogènes. L'analyse rapportée dans le tableau 1 illustre les deux profils principaux de clusters ainsi découverts.

Discussion des résultats de SNOW. D'après le tableau 1, le premier cluster représente le segment des petites voitures, légères et économiques, le deuxième cluster celui des grosses voitures plus puissantes. Toutefois, on remarque l'accélération, l'année et l'origine ne sont pas des attributs caractéristiques du premier cluster, tandis que pour le deuxième cluster seule l'accélération ne semble pas caractéristique. De manière inattendue et d'après SNOW, l'accélération n'est pas une caractéristique importante des voitures identifées comme puissantes.

5 Conclusion et perspectives

Dans cet article, nous avons présenté SNOW, un algorithme de subspace clustering qui ne repose pas sur l'hypothèse de localité. Les expériences ont montré l'amélioration des résultats par rapport à COPAC et la pertinence des résultats sur une base de données réelles.

Références

Achtert, E., C. Böhm, J. David, P. Kröger, et A. Zimek (2008). Robust clustering in arbitrarily oriented subspaces. In *8th SIAM International Conference on Data Mining (SDM)*.

Achtert, E., C. Böhm, H. P. Kriegel, P. Kröger, et A. Zimeck (2007). Robust, complete, and efficient correlation clustering. In *7th SIAM International Conference on Data Mining (SDM)*.

Achtert, E., H.-P. Kriegel, et A. Zimek (2008). ELKI : A Software System for Evaluation of Subspace Clustering Algorithms Scientific and Statistical Database Management. In *Scientific and Statistical Database Management (SSDBM)*, Berlin, Heidelberg.

Agrawal, R., J. Gehrke, D. Gunopulos, et P. Raghavan (1998). Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD Rec. 27*(2), 94–105.

Candillier, L., I. Tellier, F. Torre, et O. Bousquet (2005). SSC : Statistical Subspace Clustering Machine Learning and Data Mining in Pattern Recognition. In *4th International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM'2005)*.

Friedman, J. H. et J. J. Meulman (2004). Clustering objects on subsets of attributes. *J. Royal Statist. Soc. Series B (Statistical Methodology) 66*(4), 815–849.

Gustafson, D. E. et W. C. Kessel (1979). Fuzzy clustering with a fuzzy covariance matrix. In *IEEE CDC*, Volume 17, pp. 761–766. IEEE.

Kriegel, H. P., P. Kröger, et A. Zimek (2009). Clustering high-dimensional data : A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowl. Discov. Data 3*(1), 1–58.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. Mathematical Statist. Probability*, pp. 281–297.

Parsons, L., E. Haque, et H. Liu (2004). Subspace clustering for high dimensional data : a review. *SIGKDD Explor. Newsl. 6*(1), 90–105.

Quinlan, J. R. (1993). Combining instance-based and model-based learning. In *Machine Learning. Proceedings of the Tenth International Conference*, pp. 236–243. Morgan Kaufmann.

Summary

This article introduce a new subspace clustering algorithm called SNOW. Unlike top-down approaches, it does not assume the locality assumption and can assign a data to several clusters in different subspaces. Preliminary experiments show that our approach can provide better results than COPAC algorithm on a reference dataset.