Apprentissage non-supervisé relationnel dans l'espace des coordonnées barycentriques

Parisa Rastin, Basarab Matei, Guénaël Cabanes

LIPN-CNRS, UMR 7030, Université Paris 13
rastin@lipn.univ-paris13.fr

Résumé. Les approches basées sur les prototypes sont très populaires en apprentissage non supervisé, en raison de la compacité du modèle résultant (les prototypes), de la puissance descriptive de ces prototypes et de la faible complexité de calcul du modèle. Habituellement, le meilleur choix de prototype est le barycentre du cluster. Le prototype est alors défini comme l'objet minimisant la somme des distances carrées avec tous les objets du cluster. Cependant, dans de nombreux cas, les objets ne peuvent pas être facilement définis dans un espace euclidien sans perte d'information et/ou un pré-traitement coûteux, ce qui limite la construction des prototypes. Dans cet article, nous proposons une approche de K-moyennes relationnelle utilisant un ensemble unique de points de support basé sur le formalisme des coordonnées barycentriques, afin d'unifier la représentation des objets et des prototypes et permettant un processus d'apprentissage incrémental simple pour le clustering relationnel.

1 Introduction

L'apprentissage non supervisé (ou clustering) permet de calculer un modèle de la structure de données lorsqu'aucune autre information n'est connue. Les objets sont regroupés en "clusters", en fonction de leur similarité. Cette classification est une représentation compacte de la distribution sous-jacente des données. De nombreux algorithmes ont été proposés, qui peuvent être classés en plusieurs familles (Bishop et al., 1998). Dans cet article, nous nous intéressons à la famille des algorithmes à base de prototypes, dans laquelle chaque cluster est représenté par un prototype : un nouvel objet dans l'espace de représentation. Les approches basées sur des prototypes sont très populaires en raison de la compacité du modèle obtenu (les prototypes), du pouvoir descriptif de ces prototypes et de la faible complexité de calcul du modèle (chaque objet est comparé à un ensemble de prototypes généralement réduit). Cette faible complexité explique à elle seule la popularité des approches à base de prototypes pour des applications réelles. Habituellement, le meilleur choix de prototype est le barycentre du cluster. Le prototype est ainsi défini comme étant l'objet minimisant la somme des distances carrées avec tous les objets du cluster. Si les objets sont décrits comme des vecteurs numériques dans un espace euclidien, la définition des prototypes du cluster est simple. Dans ce cas, un prototype est un vecteur défini dans le même espace, calculé comme la moyenne vectorielle des objets appartenant à son cluster. En fait, la plupart des algorithmes basés sur des prototypes ne sont

adaptés qu'aux vecteurs définis dans un espace euclidien. Cependant, dans de nombreux cas, les objets ne peuvent pas être facilement définis dans un espace euclidien sans perte d'informations et/ou sans pré-traitement coûteux (images, réseaux, séquences, textes, par exemple). La similarité entre un tel objet n'est généralement pas une distance euclidienne et le calcul habituel des prototypes n'est plus valide. Une représentation commune pour ce type de données consiste à décrire les relations (les similarité) entre les objets à l'aide d'une matrice de dissimilarité ou de distance. Pour cette raison, elles sont parfois appelées données relationnelles. Peu de travaux ont encore été réalisés sur la classification relationnelle basée sur des prototypes, mais certains auteurs ont travaillé sur des adaptation de K-moyens (Ordonez et Omiecinski, 2004; Hathaway et al., 1989; Rossi et al., 2007; Cherki et al., 2016). Le problème principal de la classification relationnelle basée sur des prototypes est la définition des prototypes basée uniquement sur les distances entre les objets. Habituellement, les prototypes sont représentés par une combinaison linéaire des données d'entrée. Mais, en ce qui concerne la puissance de traitement et l'utilisation de la mémoire, cette mise en œuvre est très coûteuse. L'approche présentée dans le présent document a une faible complexité, afin de pouvoir traiter de gros volumes de données, tout en offrant une représentation appropriée de la structure des données en minimisant la perte d'informations. Les algorithmes de classification proposés sont indépendants de la représentation des objets, afin d'utiliser toutes l'information disponible. Nous proposons deux algorithmes basés sur cette approche : une version batch, dans laquelle l'ensemble de données est conservé en mémoire pendant tout le processus d'apprentissage et une version incrémentale dans laquelle les objets sont présentés un par un. Les deux approches sont beaucoup plus rapides et nécessitent beaucoup moins de mémoire que d'autres approches pour données relationnelles. Nous démontrons ces propriétés théoriquement et expérimentalement sur un ensemble de données relationnelles artificielles et réelles.

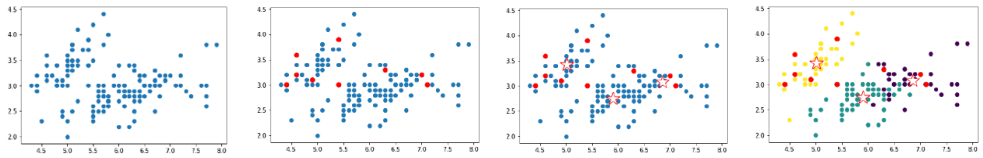
2 Clustering relationnelle basé sur Coordonnées Barycentrique

La Figure 1 est un exemple illustrant l'idée générale de l'approche proposé. Dans le système de coordonnées barycentriques Hille (2005), l'espace de représentation est défini par un ensemble unique de P points de support choisis parmi les objets O. Ces points de support peuvent être des objets choisis au hasard parmi O et représentent un espace virtuel de dimension $P-1$ (Figure 1.b). Soit $S \subset \{1,\ldots,N\}$ un sous-ensemble fini d'index, avec $P = |S| \ll N$. Nous définissons l'ensemble des points de support $O_S = \{o^i, i \in S\} \subset O$ associé à une représentation inconnue dans X par $X_S = \{\mathbf{s}^i; \mathbf{s}^i = \mathbf{x}^i, i \in S\} \subset X$. Nous visons à représenter chaque cluster par un prototype $\{\boldsymbol{\mu}^1, \boldsymbol{\mu}^2, ..., \boldsymbol{\mu}^K\}$ avec K le nombre de prototypes (Figure 1.c). Le prototype $\boldsymbol{\mu}^k$ du cluster k est défini comme une combinaison linéaire normalisée de X_S (les points de support) :

$$\boldsymbol{\mu}^k = \sum_{p=1}^{P} \beta_p^k \cdot \mathbf{s}^p, \text{ avec } \beta^k = (\beta_1^k, \ldots, \beta_p^k)^T \in \mathbb{R}^p \text{ et } \sum_{p=1}^{P} \beta_p^k = 1. \quad (1)$$

C'est également la définition des coordonnées barycentriques d'un objet dans l'espace défini par les points de support. En d'autres termes, β^k sont les coordonnées barycentriques de $\boldsymbol{\mu}^k$

par rapport au système de points de support X_S. Tout objet o de la base de données peut ainsi être défini à l'aide de coordonnées barycentriques : $o^i = \sum_{p=1}^{P} \beta_p^i \mathbf{s}^p$ avec les coordonnées β^i satisfaisant $\sum_{p=1}^{P} \beta_p^i = 1$. Pour évaluer la distance entre un objet o^i et un prototype $\boldsymbol{\mu}^k$, nous



(a) Données (b) Point de supports (c) Calcul des prototypes (d) Clustering final

FIG. 1 – *Exemple de clustering relationnel avec points de support sur les données "Iris".*

utilisons le déplacement de o^i à $\boldsymbol{\mu}^k$ et, en utilisant $\sum_{p=1}^{P}(\beta_p^i - \beta_p^k) = 1 - 1 = 0$, nous obtenons la distance suivante :

$$d^2(o^i, \boldsymbol{\mu}^k) = -\frac{1}{2}(\beta^i - \beta^k)^T \cdot D_S \cdot (\beta^i - \beta^k), \tag{2}$$

où $D_S = (d(o^i, o^j))_{i,j \in S}$ est la matrice de dissimilarité entre objets correspondant à S, l'ensemble d'index des points de support : c'est la matrice de dissimilarité entre les points supports. Nous avons par hypothèse toutes les dissimilarités $d(\mathbf{s}^i, \mathbf{s}^j)$ entre chaque paire de points de support. Cependant, pour calculer la distance décrite dans l'équation (2), nous devons calculer les coordonnées barycentriques des objets dans O. Afin d'obtenir les coordonnées β^i d'un objet o^i, en ce qui concerne le système de points de support O_S, nous considérons la matrice $P \times P$ suivante : $A = (A_{i,j})_{1 \leq i,j \leq P}$. A contient les différences entre dissimilarités de tous les points de support et celles du premier point de support uniquement. Nous considérons également J^i le vecteur de dissimilarité entre un objet o^i et les points de support X_S. Plus précisément $J^i = (J_p^i)_{1 \leq p \leq P}$ and $J_p^i = d(o,^i \mathbf{s}^1) - d(o^i, \mathbf{s}^{p+1})$ pour $1 \leq i \leq P-1$ et $J_P^i = 1$.

$$A = \begin{pmatrix} d(\mathbf{s}^1,\mathbf{s}^1)\text{-}d(\mathbf{s}^2,\mathbf{s}^1) & \dots & d(\mathbf{s}^1,\mathbf{s}^P)\text{-}d(\mathbf{s}^2,\mathbf{s}^P) \\ \cdot & \dots & \cdot \\ \cdot & \dots & \cdot \\ \cdot & \dots & \cdot \\ d(\mathbf{s}^1,\mathbf{s}^1)\text{-}d(\mathbf{s}^P,\mathbf{s}^1) & \dots & d(\mathbf{s}^1,\mathbf{s}^P)\text{-}d(\mathbf{s}^P,\mathbf{s}^P) \\ 1 & \dots & 1 \end{pmatrix}, \quad J^i = \begin{pmatrix} d(o^i,\mathbf{s}^1)\text{-}d(o^i,\mathbf{s}^2) \\ \cdot \\ \cdot \\ \cdot \\ d(o^i,\mathbf{s}^1)\text{-}d(o^i,\mathbf{s}^P) \\ 1 \end{pmatrix}. \tag{3}$$

En utilisant la symétrie de D_S, nous obtenons β^i comme solution du système linéaire suivant :

$$A \cdot \beta^i = J^i \Rightarrow \beta^i = A^{-1} \cdot J^i. \tag{4}$$

Notez que la dernière équation du système représente la contrainte normalisée $\sum_{p=1}^{P} \beta_p^i = 1$. Par conséquent, nous pouvons calculer les coordonnées barycentriques β^i pour chaque donnée o^i en utilisant (4). Nous sommes donc capables de calculer les distances entre un objet et un prototype à partir de l'équation (2). Le problème à optimiser pour trouver les coordonnées de chaque prototype reste une minimisation de l'inertie. Nous proposons deux algorithmes pour

Algorithme 1 : Algorithme proposé, version batch

Entrée : objets O, fonction d, K, P.
Sortie : coordonnées des prototypes β^k
Choisir au hasard P points de supports $O_S \subset O$.
Calculer β^i en utilisant (4) pour chaque $o^i \in O$.
Choisir au hasard K coordonnées β^i pour initialiser β^k.
while *la convergence n'est pas atteinte* **do**
    Assigner chaque objet au prototype le plus proche en utilisant (2).
    Mettre à jours β^k en utilisant (5).
end

calculer les coordonnées des prototypes : une version batch, où l'ensemble de données est conservé en mémoire et une version incrémentale, où les objets sont présentés un par un.

La version batch de l'algorithme proposé est décrite dans l'algorithme 1). On suppose que l'ensemble du jeu de données puisse être stocké dans la mémoire, ce qui permet de calculer et de stocker les coordonnées barycentriques β^i de tous les objets. En minimisant la distance carrée et en utilisant l'équation (2), nous calculons les coordonnées du prototype $\boldsymbol{\mu}^k$ du cluster k dans le système de coordonnées barycentriques défini par O_S . Les coordonnées barycentriques du prototype de C_k sont donné par :

$$\beta^k = \frac{1}{|C_k|} \sum_{i|o^i \in C_k} \beta^i \tag{5}$$

L'idée du processus incrémentale est de présenter les objets du jeu de données un par un, de manière aléatoire. La mise à jour des prototypes est calculée de manière incrémentielle pour chaque objet présenté. Comme nous pouvons calculer les coordonnées barycentriques de o^i en termes de points de support O_S (voir l'équation (4)), la règle de mise à jour de β^k peut être écrite ainsi :

$$\beta^k{}_{t+1} = \beta^k_t - \gamma(\beta^i - \beta^k_t). \tag{6}$$

où γ est le poids (ou le taux d'apprentissage) définissant l'importance de o^i dans les nouvelles coordonnées barycentriques. La procédure résultante est donnée dans l'algorithme 2.

Algorithme 2 : Algorithme proposé version incrémentale

Entrée : objets O, fonction d, K, P, γ.
Sortie : coordonnées des prototypes β^k
Choisir au hasard P points de supports $O_S \subset O$ et calculez A selon l'eq.(3).
Initialiser au hasard K coordonnées β^k tel que $\sum_{p=1}^{P} \beta^k_p = 1$.
while *la convergence n'est pas atteinte* **do**
    Sélectionner $o^i \in O$ au hasard et calculer β^i avec (4).
    Assigner o^i à son prototype le plus proche $\boldsymbol{\mu}^{k^*}$ en utilisant (2).
    Mettre à jours $\boldsymbol{\mu}^{k^*}$ en calculant β^{k^*} selon (6).
end

3 Validation expérimentale

Les algorithmes proposés sont comparés à sept algorithmes de classiques de clustering, adaptés aux matrices de dissimilarité. Pour étudier expérimentalement l'effet du nombre d'objets sur le temps de calcul et l'utilisation de la mémoire, nous avons généré des données gaussiennes avec 10 clusters et 10 dimensions. Nous avons augmenté progressivement le nombre d'objets pour observer l'augmentation du temps de calcul (Table 3). Notez que seule la version batch de l'approche proposée est présentée ici, car le temps de calcul de la version incrémentale est exactement proportionnel. Dans la table 3, "-" signifie que l'algorithme n'a pas assez de mémoire pour terminer. Comme prévu, le temps de calcul de l'algorithme proposé augmente beaucoup plus lentement que pour les autres approches. L'algorithme proposé peut traiter des ensembles de données volumineux pour un coût temporel et une consommation de mémoire très raisonnables. Ce n'est pas le cas pour les autres algorithmes.

Time (s)	**500**	**1,000**	**2,000**	**5,000**	**10,000**	**20,000**	**40,000**	**50,000**	**100,000**	**1,000,000**	**10,000,000**
Affinity	0.09	0.34	1.45	9.95	33.75	-	-	-	-	-	-
Spectral	0.05	0.16	0.56	3.91	19.44	-	-	-	-	-	-
HAC	0.00	0.03	0.13	0.80	3.16	13.55	-	-	-	-	-
HDBSCAN	0.08	0.11	0.30	1.52	6.17	29.66	-	-	-	-	-
KMed	0.00	0.03	0.06	0.44	1.97	7.66	35.32	-	-	-	-
Rel-KM	0.06	0.16	0.36	1.33	4.02	14.22	54.01	-	-	-	-
S-Rel-KM	0.14	0.30	0.61	1.70	4.02	11.45	37.10	-	-	-	-
BC-batch	0.20	0.34	0.61	1.45	2.98	5.45	12.49	13.95	27.80	274.86	2,763.08

TAB. 1 – *Temps de calcul en seconde pour chaque algorithme par rapport au nombre d'objets.*

Dans le tableau 3, nous avons examiné notre algorithme en version batch et incrémentale avec le score NMI (Normalized Mutual Information) (Strehl et Ghosh, 2003) et nous l'avons comparé à différents algorithmes testés sur 7 ensembles de données de types variés (vecteurs, textes, séquences et distributions). L'indice de qualité Silhouette donne des résultats similaires. Dans cette expérience, nous avons également utilisé 10 points de support. Les résultats démontrent la qualité des approches proposées par rapport aux algorithmes de l'état de l'art. Les qualités internes et externes de nos algorithmes sont, la plupart du temps, au moins aussi bonnes que celles de nos concurrents sur les jeux de données expérimentaux. Les versions batch et incrémentale sont de qualité très similaire.

NMI	**HAC**	**Affinity**	**HDBSCAN**	**Spectral**	**KMed**	**Rel-KM**	**S-Rel-KM**	**BC-batch**	**BC-stoch**
Art	0.94	0.63	0.98	0.80	0.87	0.83	0.86	0.92	0.92
Iris	0.74	0.47	0.76	0.75	0.79	0.76	0.80	0.80	0.80
Digits	0.04	0.48	0.70	0.69	0.64	0.73	0.70	0.69	0.65
Wine	0.11	0.16	0.09	0.41	0.42	0.39	0.32	0.40	0.41
Prot	0.86	0.77	0.86	0.86	0.86	0.86	0.86	0.81	0.86
Hist-shape	0.03	0.19	0.38	0.73	0.76	0.75	0.70	0.77	0.80
People	0.02	0.00	0.64	0.57	0.79	0.70	0.70	0.91	0.94

TAB. 2 – *Valeurs de NMI pour chaque ensemble de données et différents algorithmes.*

4 Conclusion

Dans cet article, nous avons présenté un nouveau formalisme pour la définition de prototypes basé sur la relation entre les objets. L'idée est de calculer les coordonnées barycentriques des objets en fonction de leurs différences avec un ensemble de "points de support" et de définir les prototypes dans l'espace barycentrique. Sur la base de cette idée, nous avons proposé un ensemble d'algorithmes de classification non-supervisée adaptés aux données relationnelles. Ces algorithmes ont une complexité et une utilisation de la mémoire inférieures à celles des approches existantes, sans perte de qualité.

Références

Bishop, C., M. Svensen, et C. K. Williams (1998). Gtm : The generative topographic mapping. *Neural Computation 10*, 215–234.

Cherki, S., P. Rastin, G. Cabanes, et B. Matei (2016). Improved sparse prototyping for relational k-means. In *2016 IEEE Symposium Series on Computational Intelligence, SSCI*, pp. 1–8.

Hathaway, R. J., J. W. Davenport, et J. C. Bezdek (1989). Relational duals of the c-means clustering algorithms. *Pattern Recognition 22*(2), 205–212.

Hille, E. (2005). *Analytic Function Theory*. Number vol.2 in AMS Chelsea Publishing Series. AMS Chelsea Publishing.

Ordonez, C. et E. Omiecinski (2004). Efficient disk-based k-means clustering for relational databases. *IEEE Trans. on Knowl. and Data Eng. 16*(8), 909–921.
english

Rossi, F., A. Hasenfuss, et B. Hammer (2007). Accelerating relational clustering algorithms with sparse prototype representation. In *Proceedings of the 6th International Workshop on Self-Organizing Maps*, Bielefeld (Germany).

Strehl, A. et J. Ghosh (2003). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res. 3*, 583–617.

Summary

Data clustering is a very important and challenging task in Artificial Intelligence (AI) field with many applications such as bio-informatics, medical, enhancing recommendation engines or fraud detection. Among the different families of clustering algorithms, one of the most widely used is the prototypebased clustering, because of its simplicity and reasonable computational time. In this study, we propose a prototype-based clustering algorithm for relational data based on the barycentric coordinates formalism. We compared experimentally the quality of the proposed approach on artificial and real data-sets. The experiments show the high quality of the algorithm in terms of clustering results. We also showed that our approach is a significant improvement in terms of computational and memory complexity compared to the state-of-the-art approaches. We consider that these results are encouraging and pave the road to numerous applications in data clustering.