Comparaisons structurelles de grandes bases de données par apprentissage non-supervisé

Guénaël Cabanes, Younès Bennani

LIPN-CNRS, UMR 7030,
99 Avenue J-B. Clément, 93430 Villetaneuse, France

Résumé. Dans le domaine de la fouille de données, mesurer les similitudes entre différents sous-ensembles est une question importante qui a été peu étudiée jusqu'à présent. Dans cet article, nous proposons une nouvelle méthode basée sur l'apprentissage non-supervisé. Les différents sous-ensembles à comparer sont caractérisés au moyen d'un modèle à base de prototypes. Ensuite, les différences entre les modèles sont détectées en utilisant une mesure de similarité.

1 Introduction

La croissance exponentielle des données engendre des volumétries de bases de données très importantes. Toutefois, la capacité à analyser les données reste insuffisante. Dans de nombreux cas, la capacité à mesurer des similitudes entre différents ensembles de données devient un élément important de l'analyse.

Les principaux enjeux pour l'étude de ce type de données sont, d'une part, l'obtention d'une description condensée des propriétés des données (Gehrke et al., 2001; Manku et Motwani, 2002), mais aussi la possibilité de détecter des variations ou des changements dans la structure des données (Cao et al., 2006; Aggarwal et Yu, 2007). Nous proposons dans cet article un algorithme capable de réaliser ces deux tâches [1]. Cet algorithme apprend d'abord une representation abstraite des sous-ensembles à comparer, puis évalue leur similarité en se basant sur cette représentation. La représentation abstraite est calculée par l'apprentissage d'une variante des SOM (Self-Organising Map, Kohonen (2001)), enrichie d'informations structurelles extraites des données. Nous proposons une méthode pour estimer, à partir de la SOM enrichie, une fonction de densité représentative des données. La mesure de dissimilarité entre sous-ensembles est alors une mesure de la divergence entre deux fonctions de densité.

L'avantage de cette méthode est la comparaison de structures par l'intermédiaire des modèles qui les décrivent, ce qui permet des comparaisons à n'importe quelle échelle sans surcharge de la mémoire de stockage. De plus, l'algorithme est très efficace en terme de temps d'exécution et de mémoire requise. Il est donc bien adapté pour la comparaison de grandes bases de données ou pour la détection de changement de structure d'un flux de données.

Le reste de cet article est organisé comme suit. La section 2 présente le nouvel algorithme. La section 3 décrit les tests de validation effectués et les résultats obtenus. Enfin, une conclusion est donnée dans la section 4.

[1] Ce travail a été soutenu en partie par le projet CADI (N°ANR - 07 TLOG 003), financé par l'ANR (Agence Nationale de la Recherche).

2 Un nouvel algorithme à deux niveaux pour modéliser et comparer la structure des données

Nous supposons ici que les données sont décrites sous la forme d'un vecteur numérique d'attributs. Pour commencer, les données sont modélisées à l'aide d'une SOM enrichie, de façon à construire une représentation abstraite de la structure des données. Ensuite, une fonction de densité est estimée à partir de la représentation abstraite.

2.1 Enrichissement des prototypes

Dans cette étape, certaines informations générales sont extraites à partir des données et stockées dans les prototypes lors de l'apprentissage de la SOM. Une SOM consiste en une carte à deux dimensions de M neurones qui sont connectés à n entrées, selon n connexions pondérées $w_j = (w_{1j}, ..., w_{nj})$ (aussi appelées prototypes). Chaque neurone est aussi connecté à ses voisins par des liens topologiques. L'ensemble d'apprentissage est utilisé pour organiser ces cartes selon des contraintes topologiques à partir de l'espace d'entrée.

Dans notre algorithme, les prototypes de la SOM vont être « enrichis » par l'addition de nouvelles valeurs numériques extraites de la structure des données : une mesure de la densité des données au voisinage du prototype (estimateur à noyaux Gaussiens), une mesure de la variabilité des données (distance moyenne entre le prototype et les données qu'il représente), ainsi qu'une mesure du voisinage entre deux prototypes (nombre de données ayant ces deux prototypes comme meilleurs représentants).

L'algorithme d'enrichissement procède en trois étapes :

Algorithme :

1. **Initialisation** :
   - Initialisation des paramètres de la SOM
   - $\forall i, j$ les densités locales (D_i), les valeurs de voisinages ($v_{i,j}$), les variabilités locales (s_i) et le nombre de données représentées par w_i (N_i) sont initialisés à zero.
2. **Tirage aléatoire d'une donnée** $x_k \in X$:
   - Calcul de $d(w_i, x_k)$, la distance euclidienne entre la donnée x_k et les prototypes w_i.
   - Recherche des deux meilleurs prototypes (BMUs : Best Match Units) w_{u^*} et $w_{u^{**}}$:

$$u^* = \arg\min_i(d(w_i, x_k)) \text{ et } u^{**} = \arg\min_{i \neq u^*}(d(w_i, x_k))$$

   .
3. **Mise à jour des informations structurelles** :
   - Nombre de données : $N_{u^*} = N_{u^*} + 1$.
   - Variabilité : $s_{u^*} = s_{u^*} + d(w_{u^*}, x_k)$.
   - Densité : $\forall i, D_i = D_i + \frac{1}{\sqrt{2\pi}h} e^{-\frac{d(w_i, x_k)^2}{2h^2}}$.
   - Voisinage : $v_{u^*,u^{**}} = v_{u^*,u^{**}} + 1$.
4. **Mise à jour des prototypes de la SOM** w_i comme défini dans Kohonen (2001).

5. **Répéter** T **fois les étapes** 2 **à** 4.

6. **Informations structurelles finales** : $\forall i, s_i = s_i/N_i$ et $D_i = D_i/N$.

Dans cette étude nous avons utilisé les paramètres par défaut de la SOMToolbox (Vesanto et al., 1999) pour l'apprentissage de la SOM, avec en particulier $M = 5 * \sqrt{N}$. Le choix de h est important pour de bons résultats, mais sa valeur optimale est difficile à calculer et coûteuse en temps de calcul (voir Sain et al. (1994)). Une heuristique qui semble pertinente et donne de bons résultats consiste à définir h comme la distance moyenne entre un prototype et son plus proche voisin (Cabanes et Bennani, 2008).

À la fin de cette étape, à chaque prototype est associé une valeur de densité et de variabilité, et à chaque paire de prototypes est associée une valeur de voisinage. De ce fait, une grande partie de l'information sur la structure des données est stockée dans ces valeurs. Il n'est plus nécessaire de garder les données en mémoire.

2.2 Estimation de la fonction de densité

L'objectif de cette étape est d'estimer la fonction de densité qui associe à chaque point de l'espace de représentation des données une densité. Nous connaissons la valeur de cette fonction au niveau des prototypes (D_i). Il faut en déduire une approximation de la fonction.

Nous supposons ici que cette fonction peut être correctement approximée par un mélange de noyaux Gaussiens sphériques ($\{K_i\}_{i=1}^{M}$), où K_i est une fonction Gaussienne centrée sur un prototype w_i et M est le nombre de prototypes. La fonction de densité peut alors s'écrire :

$$f(x) = \sum_{i=1}^{M} \alpha_i K_i(x) \qquad \text{avec} \qquad K_i(x) = \frac{1}{\sqrt{2\pi}.h_i} e^{-\frac{|w_i - x|^2}{2{h_i}^2}}$$

La méthode la plus populaire pour estimer un modèle de mélange (C'est à dire trouver h_i et α_i) est l'algorithme EM (Expectation-Maximization, Dempster et al. (1977)). Cependant, cet algorithme travaille dans l'espace des données. Ici nous avons seulement à disposition la SOM enrichie.

Nous proposons donc une heuristique pour choisir h_i :

$$h_i = \frac{\sum_j \frac{v_{i,j}}{N_i + N_j}(s_i N_i + d_{i,j} N_j)}{\sum_j v_{i,j}}$$

où N_i est le nombre de données représentées par le prototype w_i, s_i est la variabilité de w_i et $d_{i,j}$ est la distance euclidienne entre w_i et w_j.

Ainsi, puisque la densité D de chaque prototype w est connue ($f(w_i) = D_i$), nous pouvons utiliser une méthode de descente de gradient pour déterminer les poids α_i. Les α_i sont initialisés par les valeurs de D_i, puis ces valeurs sont réduites graduellement jusqu'à approcher au mieux $D = \sum_{i=1}^{M} \alpha_i K_i(w)$. Pour ce faire, nous optimisons le critère suivant :

$$\alpha = \arg\min_{\alpha} \frac{1}{M} \sum_{i=1}^{M} \left[\sum_{j=1}^{M} (\alpha_j K_j(w_i)) - D_i \right]^2$$

Ainsi, nous obtenons une fonction de densité qui est un modèle des données représentées par la SOM.

2.3 La mesure de dissimilarité

Il est maintenant possible de définir une mesure de dissimilarité entre deux ensembles de données A et B, représentés par deux SOMs :

$$SOM_A = \left[\{w_i^A\}_{i=1}^{M^A}, f^A\right] \qquad \text{et} \qquad SOM_B = \left[\{w_i^B\}_{i=1}^{M^B}, f^B\right]$$

Avec M^A et M^B le nombre de prototypes des modèles SOM_A et SOM_B, et f^A et f^B les fonctions de densité de A et B calculées au §2.2.

La dissimilarité entre A et B est :

$$CBd(A,B) = \frac{\sum_{i=1}^{M^A} f^A\left(w_i^A\right) log\left(\frac{f^A\left(w_i^A\right)}{f^B\left(w_i^A\right)}\right)}{M^A} + \frac{\sum_{j=1}^{M^B} f^B\left(w_j^B\right) log\left(\frac{f^B\left(w_j^B\right)}{f^A\left(w_j^B\right)}\right)}{M^B}$$

L'idée est de comparer les fonctions de densité f^A et f^B pour chaque prototype w de A et B. Si les distributions sont identiques, ces deux valeurs doivent être très proches.

Cette mesure est une adaptation de l'approximation pondérée de Monte Carlo de la mesure symétrique de Kullback–Leibler (voir Hershey et Olsen (2007)), en utilisant les prototypes de la SOM comme un échantillon de l'ensemble des données.

3 Validation

3.1 Description des distributions de données utilisées

De façon à démontrer les performances de la mesure de dissimilarité proposée, nous avons utilisé neuf générateurs de données artificielles et une base de données réelles.

Les générateurs « Rings » 1 à 3 et « Spirals » 1 et 2 génèrent cinq types d'ensembles de données non-convexes en deux dimensions, de densité et de variance différentes. « Noise 1 » à « Noise 4 » sont des distributions en deux dimensions, chacune composée d'une distribution Gaussienne accompagnée d'un bruit homogène très important. Pour finir, la base de donnée « Shuttle » vient du UCI repository. Il s'agit d'une base de données à neuf dimensions avec 58000 instances. Les données sont divisées en sept classes. À peu prés 80% des données appartiennent à la classe 1.

3.2 Validité de la mesure de dissimilarité

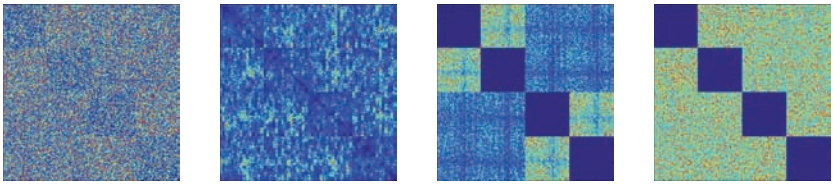
Si notre mesure de dissimilarité est performante, il devrait être possible de comparer différentes bases de données et de détecter la présence de distributions similaires. La dissimilarité des données générées selon la même loi de distribution doit être bien plus faible que la dissimilarité entre données générées selon des distributions très différentes.

Pour vérifier cette hypothèse nous avons appliqué le protocole suivant :

1. Nous avons généré 250 jeux de données de distribution « Ring 1 », « Ring 2 », « Ring 3 », « Spiral 1 » et « Spiral 2 » (50 de chaque). Ces jeux contiennent entre 500 et 50000 données.

2. Pour chacun de ces jeux de données, nous avons appris une SOM enrichie, de façon à obtenir un ensemble de prototypes représentatif de ces données. Le nombre de prototype varie entre 50 et 500.
3. Une fonction de densité a été estimée pour chaque SOM et toute ces fonctions ont été comparées les unes aux autres selon la mesure de dissimilarité proposée.
4. Pour finir, chaque SOM a été étiquetée en fonction de sa distribution (les étiquettes sont « ring 1 », « ring 2 » , « ring 3 » , « spiral 1 » et « spiral 2 »). Puis nous avons calculé un indice de compacité et de séparabilité des groupes de SOM de même étiquettes à l'aide de l'indice généralisé de Dunn (Dunn, 1974).

Le même protocole a été utilisé pour les distributions « Noise 1 » à « Noise 4 », et avec la base de données « Shuttle ». Deux types de distributions ont été extraites de la base « Shuttle », en utilisant un sous-échantillonnage aléatoire (tirage avec remise) des données de la classe 1 (« Shuttle 1 ») et des données des autres classes (« Shuttle 2 »).



(a) Ad (b) Md (c) Wd (d) CBd

FIG. 1 – *Visualisations de la matrice de dissimilarité entre différents jeux de données de distribution « Noise » 1 à 4. Cellule sombre = similarité élevée. Les comparaisons de jeux de même distribution apparaissent selon quatre carrées sur la diagonale.*

Nous avons comparé nos résultats avec certaines mesures de dissimilarité généralement utilisées pour comparer deux ensembles de données (ici on compare les deux séries de prototypes des deux SOM). Ces mesures sont la distance moyenne entre toutes les paires de prototypes dans les deux SOM (Ad), la plus petite distance entre les deux ensembles de prototypes (Md) et la distance de Ward (Wd). Les valeurs de l'indice de Dunn obtenues à partir de différentes mesures montrent que la mesure de dissimilarité proposée (CBd) est plus efficace que les mesures basées sur la distance (Table 1, voir aussi Fig. 1 pour un exemple). Ce résultat est valable pour les trois types de distributions testées : données non-convexes, données bruitées et données réelles.

TAB. 1 – *Indice de Dunn pour diverses mesures de dissimilarité sur différentes distributions.*

Distributions à comparer	Ad	Md	Wd	**CBd**
Ring 1 à 3 + Spiral 1 et 2	0.4	0.9	0.5	**1.6**
Noise 1 à 4	1.1	1.4	22.0	**115.3**
Shuffle 1 et 2	1.1	16.5	6.3	**27.6**

4 Conclusion

Dans cet article, nous avons proposé une nouvelle méthode de modélisation de la structure des données, basée sur l'apprentissage d'une SOM, ainsi qu'une mesure de dissimilarité entre modèles. Les avantages de cette méthode sont, d'une part, une grande rapidité de calcul (mise à jour « en ligne » des estimations) et une faible quantité d'information à stocker pour chaque modèle, mais aussi une grande précision dans la modélisation obtenue. Ces propriétés rendent possible l'analyse de grandes bases de données, y compris de grands flux de données, qui nécessitent à la fois vitesse et économie de ressources.

Références

Aggarwal, C. et P. Yu (2007). A Survey of Synopsis Construction Methods in Data Streams. In C. Aggarwal (Ed.), *Data Streams : Models and Algorithms*, pp. 169–207. Springer.

Cabanes, G. et Y. Bennani (2008). A local density-based simultaneous two-level algorithm for topographic clustering. In *IJCNN*, pp. 1176–1182.

Cao, F., M. Ester, W. Qian, et A. Zhou (2006). Density-based clustering over an evolving data stream with noise. In *2006 Siam Conference on Data Mining*, pp. 328–339.

Dempster, A. P., N. M. Laird, et D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B 39*, 1–38.

Dunn, J. C. (1974). Well separated clusters and optimal fuzzy partitions. *J. Cybern. 4*, 95–104.

Gehrke, J., F. Korn, et D. Srivastava (2001). On computing correlated aggregates over continual data streams. In *SIGMOD Conference*, pp. 13–24.

Hershey, J. R. et P. A. Olsen (2007). Approximating the Kullback Leibler divergence between Gaussian mixture models. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Volume 4, pp. 317–320.

Kohonen, T. (2001). *Self-Organizing Maps*. Berlin : Springer-Verlag.

Manku, G. S. et R. Motwani (2002). Approximate frequency counts over data streams. In *VLDB*, pp. 346–357.

Sain, S., K. Baggerly, et D. Scott (1994). Cross-Validation of Multivariate Densities. *Journal of the American Statistical Association 89*, 807–817.

Vesanto, J., J. Himberg, E. Alhoniemi, et J. Parhankangas (1999). Self-Organizing Map in Matlab : the SOM Toolbox. *Proceedings of the Matlab DSP Conference*, 35–40.

Summary

In data mining, the problem of measuring similarities between different subsets is an important issue which has been little investigated up to now. In this paper, a novel method is proposed based on unsupervised learning. Different subsets of a dataset are characterized by means of a prototypes based model. Differences between models are detected using a similarity measure based on data density. Experiments over synthetic and real datasets illustrate the effectiveness, efficiency, and insights provided by our approach.