Visualisation de l'intra et inter structure des groupes en classification non supervisée

Guénaël Cabanes*, Younès Bennani*

*LIPN-CNRS, UMR 7030
99 Avenue J-B. Clément, 93430 Villetaneuse, France
cabanes@lipn.univ-paris13.fr

Résumé. La croissance exponentielle des données engendre des volumétries de bases de données très importantes. Une solution couramment envisagée est l'utilisation d'une description condensée des propriétés et de la structure des données. De ce fait, il devient crucial de disposer d'outils de visualisation capables de représenter la structure des données, non pas à partir des données elles mêmes, mais à partir de ces descriptions condensées. Nous proposons une méthode de description des données à partir de prototypes enrichis puis segmentés à l'aide d'un algorithme adapté de classification non supervisée. Nous introduisons ensuite un procédé de visualisation capable de mettre en valeur la structure intra et inter-groupes des données.

1 Introduction

La croissance exponentielle des données engendre des volumétries de bases de données très importantes. Des études montrent que la quantité des données engendrées double chaque année. Parfois, l'évolution et la masse des données sont tellement importantes qu'il est impossible de les stocker dans une base et que seule une analyse "à la volée" est possible. Une solution couramment envisagée est l'utilisation d'une description condensée des propriétés et de la structure des données (Gehrke et al., 2001; Manku et Motwani, 2002; Aggarwal et al., 2003). De ce fait, il devient crucial de disposer d'outils de visualisation capables de représenter la structure des données, non pas à partir des données elles mêmes, mais à partir de ces descriptions condensées.

Dans cet article nous proposons une méthode de description des données à partir de prototypes enrichis puis segmentés à l'aide d'un algorithme adapté de classification non supervisée, puis nous introduisons des procédés de visualisation capables de mettre en valeur la structure intra et inter groupes des données.

Le reste de cet article est organisé comme suit. La section 2 présente l'apprentissage de la structure des données sous la forme d'une description condensée. La section 3 décrit l'outil de visualisation et montre quelques exemples. Une conclusion et des perspectives sont données dans la section 4.

2 Apprentissage de la structure des données

Nous proposons ici une méthode d'apprentissage de la structure des données, basée sur l'enrichissement et la segmentation automatique d'un ensemble de prototypes représentatif des données. Nous supposons ici que ces prototypes ont été préalablement calculés à partir des données, à l'aide d'un algorithme adapté tel que Neural Gaz (NG : Martinetz et Schulten, 1991) ou Self Organizing Map (SOM : Kohonen, 2001).

2.1 Principe

La première étape est l'enrichissement des prototypes par leur structure locale. En effet, nous proposons d'apprendre un certain nombre d'informations à partir des données et de les stocker avec les prototypes. Les informations associées à chaque prototype sont :

- Les modes de densité. Il s'agit d'une mesure de la densité en données au voisinage du prototype (densité locale).
- La variabilité locale. Il s'agit d'une mesure de la variabilité des données représentées par le prototype.
- Le voisinage. Il s'agit d'une mesure du voisinage du prototype. Les prototypes voisins doivent représenter le même type de données.

La deuxième étape est la segmentation (classification) des prototypes en sous-ensembles homogènes. Nous proposons une méthode de classification qui utilise directement les informations apprises lors de la première étape.

2.2 Enrichissement des prototypes

L'algorithme d'enrichissement procède en trois étapes :

Entrées :

- La matrice $Dist(w, x)$ des distances entre les M prototypes w et les N données x.

Sorties :

- Une estimation de la densité D_i et de la variabilité locale s_i associées à chaque prototype i.
- Une estimation des valeurs de voisinage $v_{i,j}$ associées à chaque paire de prototype i et j.

Algorithme :

- **Estimation de la densité** :

$$D_i = 1/N \sum_{k=1}^{N} \frac{e^{-\frac{Dist(w_i, x^{(k)})^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}}$$

  avec σ un paramètre de lissage à choisir par l'utilisateur.

- **Estimation des valeurs de voisinage** :

– Pour chaque donnée x, trouver les deux prototypes les plus proches (BMUs : Best Match Units) $u^*(x)$ et $u^{**}(x)$:

$$u^*(x) = argmin_i(Dist(w_i, x)) \text{ et } u^{**}(x) = argmin_{i \neq u^*(x)}(Dist(w_i, x))$$

– Calculer $v_{i,j} =$ le nombre de données ayant i et j comme deux premiers BMUs.

– **Estimation de la variabilité locale** : Pour chaque prototype w, la variabilité s est la distance moyenne entre w et les L données x_w représentées par w :

$$s_i = 1/L \sum_{j=1}^{L} Dist(w_i, x_w^{(j)})$$

La méthode proposée pour estimer les modes de densité est très similaire à celle proposée par Pamudurthy et al. (2007). Il a été montré que lorsque le nombre de données tend vers l'infini, l'estimateur D converge asymptotiquement vers la vraie fonction de densité (Silverman, 1986). Le choix du paramètre σ est important pour de bons résultats. Si σ est trop grand, toutes les données vont influencer la densité de tous les prototypes, les prototypes proches sont alors associés à des densités similaires, induisant une diminution de la précision de l'estimation. Si σ est trop petit, une grande proportion des données (les plus éloignées des prototypes) n'influenceront pas la densité des prototypes, ce qui induit une perte d'information. Une heuristique qui nous semble pertinente et qui donne de bons résultats est de définir σ comme la distance moyenne entre un prototype et son voisin le plus proche.

A la fin de cette étape, à chaque prototype est associée une valeur de densité et de variabilité, et à chaque paire de prototypes est associée une valeur de voisinage. 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.

2.3 Classification automatique des prototypes

De nombreuses méthodes ont été proposées pour résoudre les problèmes de classification à partir de prototypes (Bohez, 1998; Hussin et al., 2004; Ultsch, 2005; Korkmaz, 2006). Cependant, la classification obtenue n'est jamais optimale, puisqu'une partie de l'information contenue dans les données n'est pas représentée par les prototypes. Nous proposons donc une méthode de partitionnement des prototypes qui utilise les informations de densité et de voisinage apprises au §2.2, de façon à optimiser le partitionnement.

A la fin du processus d'enrichissement, les prototypes qui sont reliés par des connexions de voisinage tels que $v > 0$ définissent des groupes bien distincts, définis par la distance. Nous utilisons alors une méthode de "Watersheds" (Vincent et Soille, 1991) sur la densité de chacun de ces groupes pour détecter les zones de faible densité à l'intérieur des groupes bien séparés, de façon à caractériser les sous-groupes définis par la densité. Nous utilisons pour chaque paire de sous-groupes adjacents un indice "densité-dépendant" (Yue et al., 2004) pour déterminer si une zone de faible densité est un indicateur fiable de la structure des données, ou si elle doit être considérée comme une fluctuation aléatoire de la densité. Cette division est très rapide en raison du faible nombre de prototypes par rapport au nombre de données. L'utilisation combinée de ces deux types de définition des groupes permet d'obtenir de bons

résultats en dépit du faible nombre de prototypes utilisés et de déterminer automatiquement le nombre de clusters (cf. Cabanes et Bennani (2008)).

L'algorithme procède en trois étapes :

Entrées :
- Les valeurs de densité D_i et de voisinage $v_{i,j}$.

Sorties :
- Des groupes de prototypes similaires (les *clusters*).

1. **Extraire tous les ensembles de prototypes connectés** : Soit $P = \{C_i\}_{i=1..L}$ les L ensembles de prototypes interconnectés :

   $$\forall m \in C_i, \exists n \in C_i \text{ tel que } v_{m,n} > seuil$$

   Dans notre cas, nous choisissons $seuil = 0$.

2. **Pour chaque groupe** $C_k \in P$ **faire** :
   - Déterminer l'ensemble $M(C_k)$ des maximums locaux de densité (les modes de densité) :

     $$M(C_k) = \{\text{prototype } i \in C_k \mid D_i \geq D_j, \forall j \text{ connecté à} i\}$$

   - Calculer la matrice des seuils :

     $$S = [S(i,j)]_{i,j=1...|M(C_k)|}$$

     avec

     $$S(i,j) = \left(\frac{1}{D_i} + \frac{1}{D_j}\right)^{-1}$$

   - Pour chaque prototype $i \in C_k$, étiqueter i avec un élément $label(i)$ de $M(C_k)$, selon un gradient ascendant de densité le long des connexions de voisinage (reliant deux prototypes m et n si $v_{m,n} > seuil$). Chaque étiquette représente un sous-groupe.
   - Pour chaque paire de prototype voisins (i,j) dans C_k, si $label(i) \neq label(j)$ et si $D_i > S(label(i), label(j))$ et $D_j > S(label(i), label(j))$ alors fusionner les deux sous-groupes.

3. **Retourner les groupes fusionnés** (les *clusters*).

2.4 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 faisons ici l'hypothèse que cette fonction peut être correctement approximée sous la forme d'un mélange de noyaux gaussiens. Chaque noyau est centré sur un prototype.

Chaque noyau est de la forme :

$$G_i(x) = e^{-\frac{Dist(w_i,x)^2}{2{\lambda_i}^2}}$$

La variable λ_i contrôle l'étendue de la contribution de G_i à la densité au voisinage du prototype w_i. Elle dépend donc non seulement de la variabilité s_i des données représentées par w_i, mais aussi de la variabilité de son voisinage.

Ainsi :

$$\lambda_i = s_i + \frac{\sum_{j=1}^{M} v_{i,j} s_j}{\sum_{j=1}^{M} v_{i,j}}$$

Une approximation de la fonction de densité s'écrit alors :

$$Fd(x) = \sum_{i=1}^{M} \alpha_i G_i(x)$$

Puisque la densité D au niveau des prototypes w est connue ($Fd(w_i) = D_i$), on peut déterminer les pondérations α_i. Ces pondérations sont solution du système d'équations linéaires suivant :

$$D = \sum_{i=1}^{M} \alpha_i G_i(w)$$

avec

$$D = [D_j]_{j=1}^{M} \text{ et } w = [w_j]_{j=1}^{M}$$

Cependant, il existe une infinité de solutions à cette équation, ce qui rend impossible toute résolution matricielle basée sur une inversion de matrice. De plus, la solution obtenue par calcul de la pseudo-inverse (Ben-Israel et Greville, 2003) n'est souvent pas satisfaisante, en particulier parce qu'elle peut contenir des valeurs de α négatives qui ne garantissent plus la contrainte : $\forall x, Fd(x) > 0$. Nous utilisons donc pour résoudre ce système une méthode très simple de descente de gradient. Les α_i sont initialisés par les valeurs de D_i, puis voient leur valeur adaptée progressivement (avec une valeur minimum de 0) jusqu'à satisfaire au mieux $D = \sum_{i=1}^{M} \alpha_i G_i(w)$. Ainsi les valeurs de α restent en moyenne proportionnelles aux valeurs de D_i, ce qui satisfait l'hypothèse que chaque densité D_i est générée principalement par le prototype w_i. Pour cela, nous optimisons le critère suivant :

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

Algorithme :

1. **Initialisation** : $\forall i, \alpha_i = D_i$
2. **Calcul de l'écart** : $\forall i, ecart(i) = \sum_{j=1}^{M} \alpha_j G_j(w_i) - D_i$
3. **Mise à jours des coefficients** : $\forall i, \alpha_i(t) = max\,[0\,;\, \alpha_i(t\text{-}1) - \epsilon * ecart(i)]$, avec ϵ le pas du gradient. Nous utilisons ici $\epsilon = 0.1$.
4. **Tant que** $moyenne(|ecart|) > seuil$: retourner en 2, sinon retourner les α_i. Le seuil est choisi par l'utilisateur, nous choisissons ici 1% de la densité moyenne.

3 Visualisation

3.1 Description du procédé de visualisation

La classification est accompagnée d'un ensemble d'informations qui peut être utilisé pour compléter l'analyse des données, telles que la matrice des distances entre prototypes et la matrice de densité, mais aussi les valeurs des connexions qui peuvent être utilisées pour déterminer l'importance relative de chaque prototype pour la représentation des données. Il est possible de représenter toutes ces informations en une seule image permettant une analyse fine de la structure de chaque groupe et de leurs relations :

- Les prototypes sont projetés dans un espace à deux dimensions (éventuellement trois) à l'aide d'une projection de Sammon, qui conserve au mieux les distances initiales entre prototypes (Sammon Jr., 1969).
- La taille des disques représentant les prototypes est proportionnelle à la densité associée à chaque prototype.
- La couleur de chaque prototype dépend du cluster auquel il est associé.
- Les connexions de voisinage (topologie locale) sont représentées par un segment reliant les prototypes voisins.
- Les valeurs locales de densité et de variabilité nous permettent d'estimer les variations de densité dans l'espace de représentation. Ces variations sont représentés sous la forme de courbes de niveaux. La projection des courbes de niveaux dans le plan est effectuée par une projection des Gaussiennes du mélange dans l'espace de représentation.

Cette visualisation permet d'obtenir des informations à la fois sur la structure inter groupes (nombre de groupes, similarités entre les groupes) mais aussi la structure intra groupe (topologie locale, densité locale et variation de densité au sein du groupe, variabilité des données représentés).

3.2 Applications

Nous avons appliqué ce procédé à huit bases de données artificielles et réelles, en nous appuyant sur une Carte Auto-Organisatrice ou Self-Organizing Map (SOM : Kohonen, 2001) pour l'apprentissage des prototypes. Une SOM se compose d'un ensemble de neurones artificiels qui représentent la structure des données. Ces neurones sont connectés avec leurs voisins selon des connexions topologiques (où connexions de voisinage). L'ensemble de données à analyser est utilisé pour organiser la SOM sous des contraintes topologiques de l'espace d'entrée. Ainsi, une correspondance entre l'espace d'entrée et l'espace de la carte est construit. Deux observations proches dans l'espace d'entrée doivent activer le même neurone ou deux neurones voisins de la SOM. Chaque neurone est associé à un prototype et, pour vérifier les contraintes de voisinage, les neurones voisins du neurone le plus représentatif d'une donnée mettent à jour leur prototype pour une meilleure représentation de cette donnée. Cette mise à jour est d'autant plus importante que les neurones sont de proches voisins du meilleur neurone.

La table 1 résume les caractéristiques des bases de données utilisées.

TAB. 1 – *Description des bases de données utilisées.*

Base de données	Type	Taille	Dimension
Engytime	Artificielle	4096	2
Hepta	Artificielle	212	3
Lsun	Artificielle	400	2
Rings	Artificielle	2500	2
Spirals	Artificielle	5000	2
Iris	Réelle	150	4
Fourmis	Réelle	80	11
Enfants	Réelle	120	8

Les figures 1 à 5 montrent quelques exemples de visualisations pouvant être obtenues à partir de données artificielles de faible dimension.

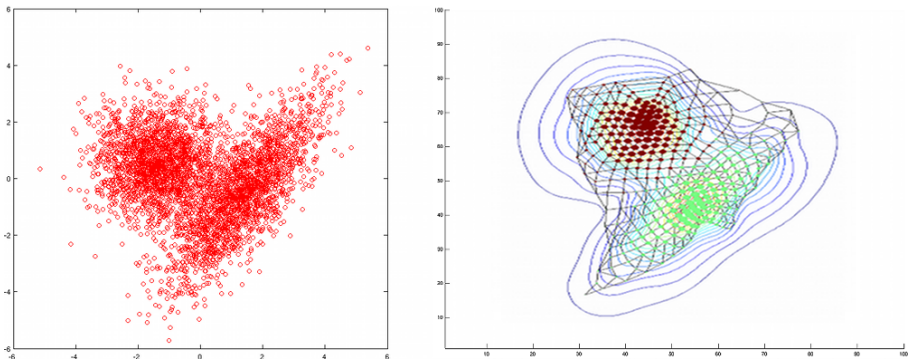

FIG. 1 – *Données "Engytime" (à gauche) et leur visualisation (à droite).*

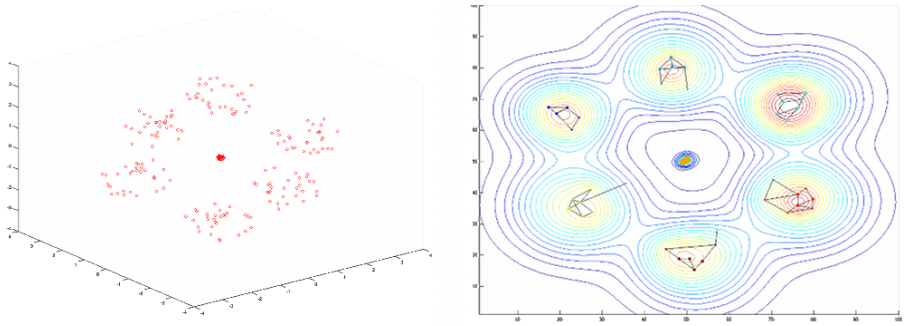

FIG. 2 – *Données "Hepta" (à gauche) et leur visualisation (à droite).*

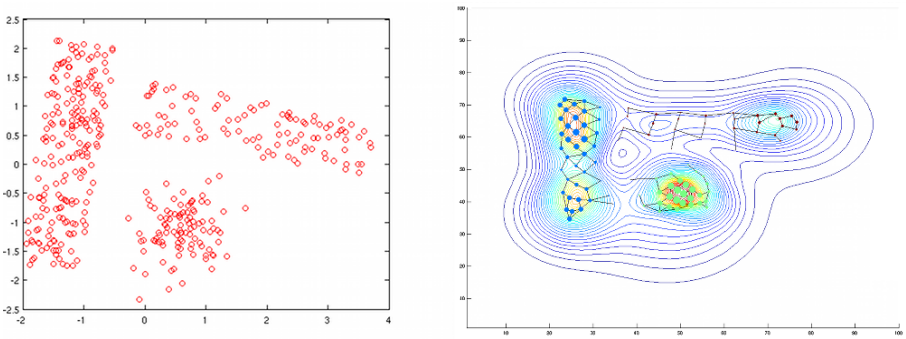

FIG. 3 – *Données "Lsun" (à gauche) et leur visualisation (à droite).*

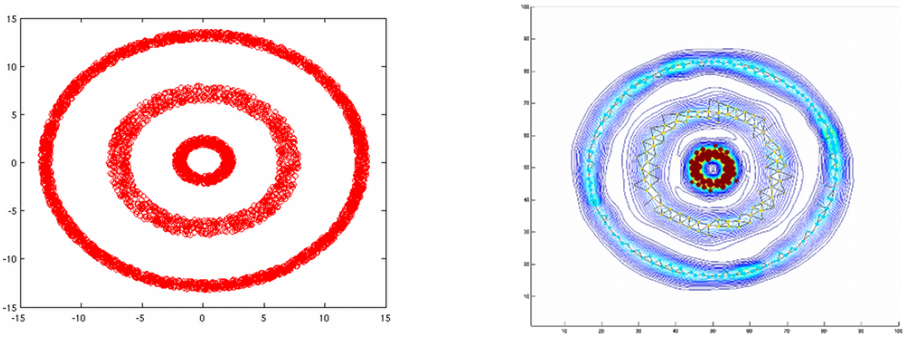

FIG. 4 – *VDonnées "Rings" (à gauche) et leur visualisation (à droite).*

On remarque que la structure des données est bien conservée par l'algorithme de quantification et de classification et qu'elle est bien représentée par le procédé de visualisation. La

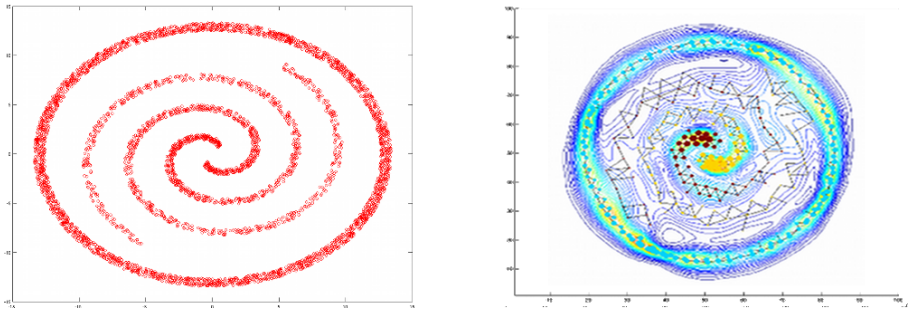

FIG. 5 – *Données "Spirals" (à gauche) et leur visualisation (à droite).*

densité des données est facilement identifiable par la taille de la représentation des prototypes et par les lignes de niveaux. Ces dernières permettent de plus de visualiser en deux dimensions la forme générale des différents clusters et leur taille relatives. La visualisation des connexions, ainsi que les différentes couleurs associées aux prototypes, autorisent une description visuelle de la segmentation des données en différents clusters. Deux clusters fortement connectés sont représentatifs de groupes de données en contact, comme dans la figure 1, alors que l'absence de connexion dénote des groupes bien séparés dans l'espace de représentation des données (par exemple dans la figure 2). De plus, la visualisation est suffisamment fine pour permettre une représentation des données de distribution complexe, comme cela est illustré par les figures 4 et 5.

Les figures 6 à 8 montrent quelques exemples de visualisations pouvant être obtenues à partir de données réelles. Les données "Iris" sont une description selon quatre modalités de fleurs de trois espèces différentes. Les données "Fourmis" décrivent l'activité de chaque individus d'une colonie de fourmis (11 variables). Enfin les données "Enfants" sont une description du temps passé dans différentes activités de jeu dans un groupe d'enfants (8 variables).

La visualisation de ces bases de données de petite taille mais de dimension supérieure à trois illustre la capacité du procédé de visualisation à projeter les informations pertinentes dans un espace à deux dimensions. Ainsi, les données "Iris" (Fig. 6) sont structurées en deux groupes bien distincts, un de ces deux groupes étant lui-même subdivisé en deux groupes très proches. Les trois groupes sont découverts automatiquement par l'algorithme de classification et correspondent à trois espèces distinctes de fleurs. Pour les données "Fourmis" (Fig. 7), chaque cluster détecté par l'algorithme correspond à un comportement et un rôle social différent au sein de la colonie (Chasseuses, nourrices, nettoyeuses, gardes, etc...). Ici, on n'observe pas de séparations nette en terme de densité entre les groupes, ce qui signifie que certains comportements intermédiaires sont possibles. L'existence de ces comportements intermédiaires sont connus en biologie, en particulier grâce à la présence de fourmis généralistes capables d'effectuer n'importe quelle tâche en fonction des besoins de la colonie (voir Holldobler et Wilson, 1990). Enfin, les données "Enfants" (Fig. 8) représentent les activités de jeux d'enfants de maternelles en récréation. Les données sont divisées en deux ensembles de densité assez bien séparés, chacun subdivisé en deux sous-ensembles. Le sous-groupe central a lui-même été subdivisés en trois clusters par l'algorithme. Il est intéressant de noter que, globalement, l'ordre

Visualisation de la structure des groupes en classification non supervisée

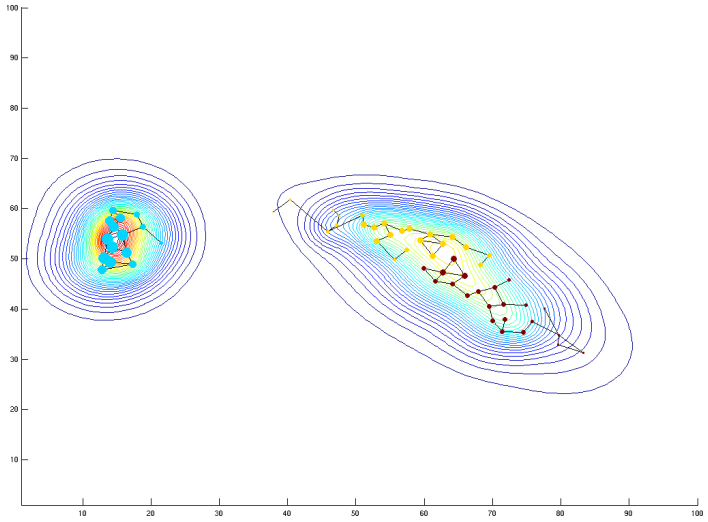

FIG. 6 – *Visualisation des données "Iris".*

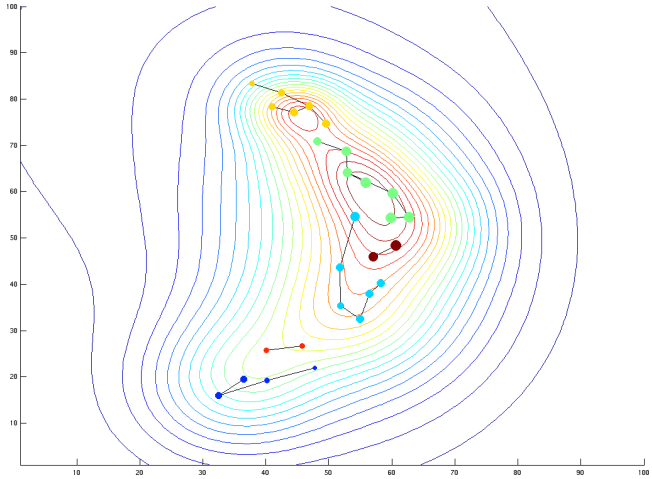

FIG. 7 – *Visualisation des données "Fourmis".*

des groupes de haut en bas correspond à une augmentation de l'âge des enfant et une augmentation de la complexités des activités de jeux. Le groupe jaune est pratiquement uniquement composé d'enfant en première année de maternelle, alors que la grande majorité des enfant en dernière année sont dans le groupe marron. La subdivision des deux années intermédiaires en quatre clusters dénote des différences individuelles dans la dynamique du développement des enfants. La baisse de densité entre le groupe bleu clair et le groupe vert sépare les enfant passant le plus clair de leur temps en jeux sociaux (avec leurs congénères) des enfants jouant

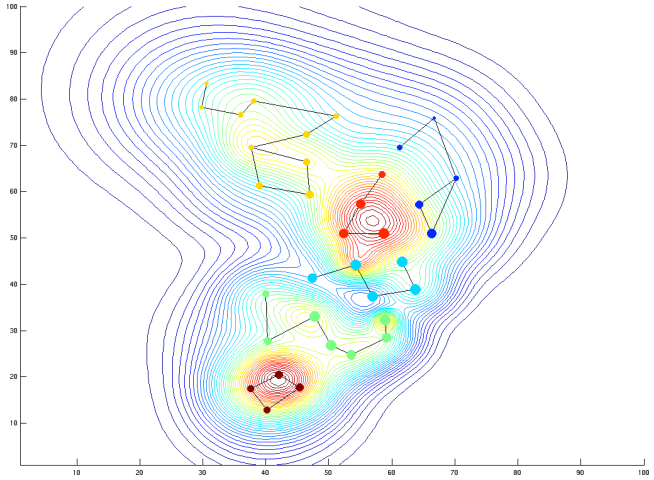

FIG. 8 – *Visualisation des données "Enfants".*

le plus souvent seul. Cela indique qu'un enfant qui commence à jouer avec ses congénères ne reviendra plus, ou rarement, à des jeux solitaires. L'ensemble de ces informations semblent en accords avec les connaissances du domaine (cf. Fromberg et Bergen, 2006).

4 Conclusion

Dans cet article, nous avons proposé une nouvelle méthode de modélisation, à base de prototypes, de la structure des données, ainsi qu'un procédé de visualisation de cette structure, capable de mettre en valeur la structure intra et inter-groupes des données. Nous avons montré sur quelques exemples artificiels et réels la pertinence de la visualisation proposée.

La poursuite de nos travaux portera sur le data mining interactif. En effet, permettre à l'utilisateur d'interagir avec les visualisations proposées pourrait aboutir à une analyse exploratoire plus fine de la structure des données. L'ajout d'informations par l'intermédiaire d'un affichage d'étiquettes textuelles sur les prototypes est aussi envisagée.

Références

Aggarwal, C. C., J. Han, J. Wang, et P. S. Yu (2003). A framework for clustering evolving data streams. In *Very Large Data Base*, pp. 81–92.

Ben-Israel, A. et T. N. E. Greville (2003). *Generalized Inverse : Theory and Applications.* New York : Springer Verlag.

Bohez, E. L. J. (1998). Two level cluster analysis based on fractal dimension and iteratedfunction systems (ifs) for speech signal recognition. *IEEE Asia-Pacific Conference on Circuits and Systems*, 291–294.

Cabanes, G. et Y. Bennani (2008). A local density-based simultaneous two-level algorithm for topographic clustering. In *Proceeding of the International Joint Conference on Neural Networks*, pp. 1176–1182.

Fromberg, D. P. et D. Bergen (2006). *Play from birth to Twelve : Contexts, perspectives, and Meanings*. New York : Routledge.

Gehrke, J., F. Korn, et D. Srivastava (2001). On computing correlated aggregates over continual data streams. In *Special Interest Group on Management of Data Conference*, pp. 13–24.

Holldobler, B. et E. Wilson (1990). *The ants*. Cambridge, MA : Harvard University Press.

Hussin, M. F., M. S. Kamel, et M. H. Nagi (2004). An efficient two-level SOMART document clustering through dimensionality reduction. In *ICONIP*, pp. 158–165.

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

Korkmaz, E. E. (2006). A two-level clustering method using linear linkage encoding. In *International Conference on Parallel Problem Solving From Nature, Lecture Notes in Computer Science*, Volume 4193, pp. 681–690. Springer-Verlag.

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

Martinetz, T. M. et K. J. Schulten (1991). A "neural-gas" network learns topologies. In T. Kohonen, K. Mäkisara, O. Simula, et J. Kangas (Eds.), *Artificial Neural Networks*, pp. 397–402. Amsterdam : Elsevier Science Publishers.

Pamudurthy, S. R., S. Chandrakala, et C. C. Sakhar (2007). Local density estimation based clustering. *Prodeeding of International Joint Conference on Neural Networks*, 1338–1343.

Sammon Jr., J. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computer 18*(5), 401–409.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC.

Ultsch, A. (2005). Clustering with SOM : U*C. In *Procedings of the Workshop on Self-Organizing Maps*, pp. 75–82.

Vincent, L. et P. Soille (1991). Watersheds in digital spaces : An efficient algorithm based on immersion simulation. *IEEE Trans. Pattern Anal. Mach. Intell. 13*, 583–598.

Yue, S.-H., P. Li, J.-D. Guo, et S.-G. Zhou (2004). Using greedy algorithm : DBSCAN revisited II. *Journal of Zhejiang University SCIENCE 5*(11), 1405–1412.

Summary

The exponential growth of data generates terabytes of very large databases. One solution commonly proposed is the use of a condensed description of the properties and structure of data. Thus, it becomes crucial to have visualization tools capable of representing the data structure, not from the data themselves, but from these condensed descriptions. We propose here a method of describing data from enriched and segmented prototypes using a clustering algorithm. We then introduce a visualization tool that can enhance the structure within and between groups of data.