Méthodes incrémentales de réduction de dimensions

Yoann Didry, Olivier Parisot, Philippe Pinheiro, Thomas Tamisier

Centre de Recherche Public - Gabriel Lippmann
41, rue du Brill, L-4422 Belvaux, Luxembourg
didry@lippmann.lu

Résumé. Les méthodes de réduction de dimensions sont très populaires en fouille de données, car elles permettent de faciliter la visualisation et la compréhension des données. Cependant, elles ne peuvent pas être appliquées telles quelles sur des flux de données. Dans ce papier, nous proposons un bref état de l'art des principales techniques utilisées pour la réduction de dimensions, dans le but de les utiliser avec des flux.

1 Introduction

De nombreuses méthodes existent pour projeter des données multidimensionnelles vers un espace réduit. Si l'on souhaite visualiser ces données dans le but d'y extraire de la connaissance, le plus souvent on choisira une dimension de 2 ou 3. Il existe de nombreuses méthodes adaptées aussi bien aux cas supervisés ou non-supervisés (Bennani et Guérif, 2008). Les problématiques actuelles des données venant en flux rendent difficile l'application directe de ces méthodes, car généralement la visualisation de ces données n'est pas stable *topologiquement* (i.e. l'ajout d'un simple point peut changer totalement les coordonnées des données existantes dans l'espace réduit). L'utilisateur aura donc des difficultés à exploiter la visualisation de ces projections successives. De plus, le calcul de chaque nouvelle projection est très coûteux car il est nécessaire de refaire les calculs pour l'ensemble des points, ce qui est à proscrire dans la pratique.

Dans ce travail, nous explorons principalement trois techniques standard de réduction de données : MDS (Multidimensional scaling), Isomap et LLE (Local Linear Embedding), dont on pourra trouver des détails complets dans Wickelmaier (2003); Tenenbaum (2000); Roweis (2000). Nous illustrons quelques méthodes de l'état de l'art pour permettre à ces techniques d'être utilisées avec les flux de données. (Agarwal et III, 2010; Law, 2004; Kouropteva, 2005) Le reste de l'article est composé comme suit. La première section aborde les travaux relatifs aux méthodes classiques dites *non itératives*. La seconde section décrit les méthodes *itératives* plus récentes qui peuvent être adaptées avecs les flux de données. Enfin, la dernière section évoque les différentes pistes qui peuvent être explorées pour améliorer les techniques existantes.

2 Les méthodes non-itératives

Nous détaillons MDS, Isomap et LLE qui font partie de la base de la plupart des nouveaux algorithmes de réduction de dimensions. MDS est une approche dite globale, qui permet d'observer des structures localement linéaires dans l'espace réduit. Tandis que Isomap et LLE sont des approches locales, plus adaptées aux structures non-linéaires (Silva et Joshua, 2002). D'autres méthodes non itératives à la fois adaptées aux structures linéaires et non-linéaires sont souvent étudiées dans la littérature (Deng et Lian, 2010).

2.1 MDS

La méthode classique a pour but de minimiser la fonction de stress (Kruskal, 1978) :

$$\sqrt{\frac{\sum_{i<j}\left(d_1(x_i,x_j)-d_2(r_i,r_j)\right)^2}{\sum_{i<j} d_2(r_i,r_j)^2}}$$

où $(x_1,\ldots,x_n)\in\mathbb{R}^s$ sont les vecteurs de l'espace original et $(r_1,\ldots,r_n)\in\mathbb{R}^t$ (avec $t\ll s$) sont les éléments dans l'espace réduit (appelé espace projeté). d_1 et d_2 sont des distances dans $\mathbb{R}^s$ et $\mathbb{R}^t$ respectivement.
Le plus fréquemment $t=2$ et $d_1(x,y)=\|x-y\|_2, d_2(x,y)=\|x-y\|_2$ sont des distances euclidiennes standard. Concrètement, le but est de conserver les distances de l'espace original, de telle sorte que deux points proches dans l'espace initial soient proches dans l'espace final. L'algorithme MDS standard procède ainsi :

1. Calculer la matrice des carrés des distances entre les éléments dans l'espace d'origine D^2 ;
2. Centrer cette matrice classiquement en calculant $B=-\frac{1}{2}JD^2J$ où $J=I_n-\frac{1}{n}\mathbf{1}\mathbf{1}^T$ ($\mathbf{1}$ est le vecteur de taille $n\times 1$ ne contenant que des 1 et I_n est la matrice identité) ;
3. Calculer les t plus grandes valeurs propres $\lambda_1,\ldots,\lambda_t$ de B et les vecteurs propres correspondantes $e_1,\ldots,e_t$;
4. $X=(e_1|\ldots|e_t)\text{diag}(\lambda_1,\ldots,\lambda_t)^{1/2}$ sont les coordonnées finales des données originales dans l'espace réduit.

L'approche MDS classique n'est pas adaptée pour projeter des structures qui ne seraient pas des variétés linéaires affines. Pour projeter des structures non linéaires de ce type, d'autres techniques telles que Isomap et LLE sont employées.

2.2 Isomap

Isomap reprend les idées de MDS, mais utilise une notion de distance différente entre les éléments de l'espace initial. L'idée est d'utiliser la notation de distance géodésique. La distance géodésique entre deux points de l'espace original est définie comme la longueur du plus court chemin dans le graphe de voisinage G construit en fixant $K\in\mathbb{N}^\star$ (nombre de plus proches voisins) et $\epsilon\in\mathbb{R}^{+\star}$ (rayon de voisinage). Ces distances sont calculées par l'algorithme de Dijkstra ou de Floyd-Warshall en utilisant la matrice de prédécesseurs (Law, 2006). Dans G, chaque nœud correspond à un élément du jeu de données $(x_1,\ldots,x_n)$.

Deux nœuds x_i et x_j sont connectés par un arc de poids $w = d_1(x_i, x_j)$ si la distance entre x_i et x_j est parmi les K plus petites distances inférieures à ϵ entre x_i et les autres éléments . Un exemple est présenté dans la figure 1 ci-dessous.

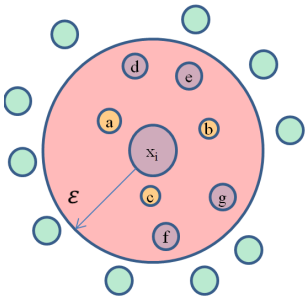

FIG. 1 – *Le point x_i est représenté dans le plan. Les points a, b et c sont les $K = 3$ plus proches voisins de x_i présents dans un cercle de rayon ϵ. Les points d, e, f et g sont dans ce même cercle, mais ne font pas partie du voisinage.*

L'utilisation de la distance géodésique pour approximer les distances dans l'espace d'origine peut être problématique dans le cas où ce dernier ne respecte pas certaines conditions de régularité (jeu de données sans outlier). Des extensions telles que EN-ISOMAP (Shao, 2005, 2012) ont vu le jour pour permettre de manipuler des jeux de données avec une structure dite imparfaite.

2.3 LLE

L'algorithme n'utilise qu'un seul paramètre utilisateur : K. En général, ce paramètre est fixé par validation croisée. La distance utilisée ici est la distance euclidienne. Choisissons n points $x_1, \ldots, x_n \in \mathbb{R}^s$ afin de les projeter dans un espace de dimension $d \ll n$. L'idée de LLE consiste à calculer chaque point x_i par une combinaison linéaire de ces K plus proches voisins afin de permettre la reconstruction le plus précisément possible dans un espace euclidien (typiquement le plan $\mathbb{R}^2$). L'algorithme repose donc sur le calcul des poids intervenant dans ces combinaisons linéaires. L'algorithme LLE fonctionne selon les étapes suivantes :

1. Calculer pour chaque point x_i ses K plus proches voisins $x_i^1, \ldots, x_i^K$;
2. Minimiser $\epsilon(\mathbf{W}) = \sum_{i=1}^n \|x_i - \sum_{j=1}^n w_{ij} x_j\|^2$;
   soumis aux contraintes $w_{ij} = 0$ si $x_j \notin \{x_i^1, \ldots, x_i^K\}$ et $\sum_{j=1}^n w_{ij} = 1$;
3. Minimiser $\delta(Y) = \sum_{i=1}^n \|y_i - \sum_{j=1}^n w_{ij} y_j\|^2$;
   soumis aux contraintes $\frac{1}{N} \sum_{i=1}^n y_i y_i^T = I$ et $\sum_{i=1}^n y_i = 0$
   Cette dernière étape est équivalente à trouver les $d + 1$ vecteurs propres associés aux $d+1$ plus petites valeurs propres de la matrice symétrique $M = (I-W)^T(I-W)$. Le premier vecteur propre composé uniquement de 1 est écarté et seulement les d suivants sont conservés ;
4. Les d vecteurs propres choisis composent alors la projection finale Y.

LLE a plusieurs avantages comparé à Isomap, son temps d'exécution est en général plus rapide car son implémentation utilise des matrices creuses. De plus, de nombreux benchmarks montrent de meilleurs résultats sur des jeux de données académiques, ou issues de la vie courante (Bernstein et Erofeev, 2012). En revanche, comme pour Isomap, une certaine régularité de la distribution des données est requise et des extensions existent dans le cas de jeux de données ayant une structure dite imparfaite (Hadid, 2003; Chang, 2006) .

2.4 Limites des méthodes non-itératives

L'approche Isomap a déjà été critiquée pour ne pas être stable topologiquement lorsque le jeu de données est altéré en rajoutant du bruit blanc gaussien aux coordonnées des données initiales (voir figure 2, issue des travaux de Mukund (2002)). De la même manière, l'algorithme MDS ne réagit pas bien à la présence de bruit ou d'outliers. Des méthodes similaires robustes ont ainsi vu le jour pour pallier ce défaut (Du et Zhao, 2012; Forero, 2012; Chang, 2005, 2006). Il en est de même lorsque les données viennent en flux, la structure évolue trop rapidement pour que l'ensemble des projections dans le temps soit interprétable visuellement par l'utilisateur.

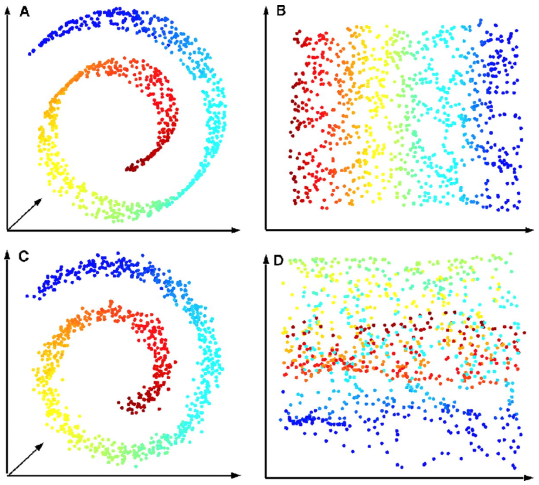

FIG. 2 – *(A) Swiss roll dataset utilisé par Tenenbaum (2000) (B) représentation de l'espace projeté par Isomap avec* $K = 7, \epsilon = 5$ *(C) Ajout aux coordonnées des points de (A)d'un bruit blanc gaussien (*$\sigma = 2\%$ *de la plus petite boite entourant les données) (D) Isomap sur les données bruitées avec* $K = 7, \epsilon = 5$*. On remarque que la structure topologique n'est plus respectée (les points bruités sont mélangés avec d'autres classes)*

Il faut noter que Isomap, LLE et MDS utilisent en général la distance euclidienne dans le sous-espace projeté, ce qui ne permet pas toujours de capturer finement les structures non-linéaires, pour ce faire, des dérivations à base de noyaux sont souvent employées (Choi, 2007; Zimmer, 2014).

3 Méthodes itératives

Nous supposons que les données arrivent en flux, sont de dimension d et que nous souhaitons les projeter dans un espace de dimension $k \ll d$. Deux approches sont généralement étudiées, la première consiste à calculer les coordonnées du point x_t en fonction des précédents $x_1, \ldots, x_{t-1}$, qui sont fixes, tandis que la seconde adapte également les coordonnées des points précédents. La première approche permet d'obtenir des visualisations de ces projections topologiquement stables, mais moins justes, tandis que la seconde permet d'obtenir des visualisation plus justes, mais topologiquement non stables. Bengio et Vincent (2003) étendent ainsi les cinq méthodes les plus classiques de réduction de dimensions (MDS, Isomap, LLE (Roweis, 2000), Eigenmaps (Belkin, 2003) et Spectral Clustering (Weiss, 1999; Ng, 2003)) avec la première approche. Nous détaillons ci-dessous trois techniques (Incremental MDS (Agarwal et III, 2010), Incremental Isomap (Law, 2004, 2006) et Incremental LLE (Kouropteva, 2005)) qui étendent les méthodes décrites dans la première partie, dans le cadre d'une utilisation avec des flux de données.

3.1 Incremental MDS

Les points $x_1, \ldots, x_{t-1} \in \mathbb{R}^k$ sont les points déjà projetés et $r_1, \ldots, r_{t-1} \in \mathbb{R}$ sont les distances d'un nouveau point arrivant à l'instant t aux points déjà projetés aux instants $1, \ldots, t-1$ dans $\mathbb{R}^d$.
L'objectif de iMDS est de calculer x_t en s'aidant des points déjà projetés $x_1, \ldots, x_{t-1}$:

$$x_t = \text{argmin}_{p \in \mathbb{R}^k} \sum_{i=1}^{t-1} \left(\|p - x_i\|_2 - r_i \right)^2$$

En d'autres termes, le nouveau point projeté x_t est choisi de telle manière à minimiser la somme des erreurs au carré entre les distances en dimension k et en dimension d.
La résolution de ce problème de minimisation est possible en utilisant la méthode proposée par Agarwal et Venkatasubramanian (2010) qui consiste à réduire l'erreur de manière itérative en utilisant une approche originale de l'algorithme min-sum, qui était déjà résolu de manière exacte ou approximative (Bose et Morin, 2003; Weiszfeld, 1938) dans la littérature.
L'algorithme proposé demande de conserver l'historique de tous les points précédents, ce qui n'est en général pas possible pour un flux de données. Cependant, des expériences démontrent que conserver seulement les $2k$ derniers points permet d'avoir des résultats satisfaisants (Agarwal et III, 2010).

3.2 Incremental Isomap

L'idée de la version incrémentale d'Isomap repose sur la mise à jour du graphe de voisinage de manière itérative, sans recalculer le graphe complet. L'algorithme s'intéresse en particulier à l'impact de l'ajout d'un nœud dans le graphe de voisinage, de manière à mettre à jour les plus courts chemins de manière optimale. De même, les coordonnées des nouveaux points sont calculés grâce à un schéma incrémentale de mise à jour des valeurs et vecteurs propres dominants, basé sur une accélération de type Rayleigh-Ritz (Golub, 1996).
La version standard incrémentale reste gourmande en mémoire ($O(n^2)$) où n est le nombre

de points, ce qui la rend inutilisable concrètement pour les grands volumes de données. Pour cette raison, une extension appelée Incremental Landmark Isomap (Law, 2006) utilise seulement $m \ll n$ points pour maintenir son graphe de voisinage. En revanche, le choix de ces m points est crucial pour obtenir des résultats corrects et il faut respecter certaines contraintes géométriques (Silva, 2003).

3.3 Incremental LLE

La version incrémentale de LLE, calcule la matrice de taille $(n+1) \times (n+1)$ (dite de coût) $M_{new} = (I - W_{new})^T (I - W_{new})$ de la même manière que LLE standard (première étape de l'algorithme). Ensuite, la méthode suppose que les valeurs propres les plus petites de M et M_{new} sont quasiment inchangées. De cette manière, les nouvelles coordonnées Y_{new} sont obtenues en résolvant le problème de minimisation :

$$Y_{new} = \text{argmin}_Y \{ Y M_{new} Y^T - \text{diag}(\lambda_1, \ldots, \lambda_d) \}$$

où $\lambda_1, \ldots, \lambda_d$ sont les d plus petites valeurs propres de M. La dernière étape de l'algorithme travaille donc sur une matrice de taille $d \times d$ au lieu d'une matrice de taille $N \times N$, ce qui améliore grandement la rapidité de l'algorithme.

4 Problèmes et perspectives

La plupart de ces méthodes présupposent que les données soient numériques et ne présentent pas de valeurs manquantes. Pourtant, il est très fréquent que les jeux de données de la vie réelle (biologie, médecine, sociologie, etc) utilisent des données à la fois mixtes et incomplètes. Une solution est d'utiliser une distance plus appropriée pour ce type de jeux de données. Par exemple Tecuanhuehue-Vera et Martínez-Trinidad (2012) étend MDS aux données mixtes et incomplètes en se basant sur la distance HEOM (Wilson, 1997). Cette distance permet de comparer des données quantitatives et qualitatives contenant des valeurs manquantes. Dans la littérature, un certain nombre d'algorithmes utilisant des données mixtes (Huang, 1997; Kim, 2004; Huang, 1998; Chan et Ching, 2004) utilisent l'appariement simple (distance de Hamming (Hamming, 1950)), qui définit la distance entre deux vecteurs de mots par $d((x_1, \ldots, x_n), (y_1, \ldots, y_n)) = \sum_{i=1}^{n} \delta_{x_i, y_i}$ avec $\delta(x_i, y_i) = \begin{cases} 0 & \text{si} \quad x_i = y_i \\ 1 & \text{si} \quad x_i \neq y_i \end{cases}$
Cette approche est problématique car elle ne prend pas en compte la sémantique et la similarité sous-jacente aux valeurs catégorielles (Hsu et Chin-Long, 2007). Des méthodes basées sur la co-occurence ont vu le jour pour pallier ce problème (Ahmad et Lipika, 2007; Shih, 2010).
Un autre problème est lié au caractère infini des flux de données, pour lequel il est impossible de stocker toute l'information. En effet, il n'est pas possible d'afficher tous les points indéfiniment dans l'espace projeté. Il est donc souvent nécessaire de faire appel à des techniques de clustering de stream, qui permettent d'oublier les données obsolètes et regroupent les données émergentes (Gaber, 2012; Wan et Dang, 2009; Tu, 2009). Combiner ces méthodes de clustering avec des techniques de projection incrémentale est donc une piste particulièrement intéressante à explorer.

5 Conclusion

De nombreuses méthodes existent pour réduire les dimensions de données. Les méthodes standard ne sont en général pas suffisantes pour gérer de grands volumes, et ne sont pas stables topologiquement à l'ajout de bruit, ou même de nouvelles données. Pour pallier le problème de résistance au bruit, des méthodes dites robustes ont vu le jour. Tandis que des méthodes itératives, n'utilisant qu'un sous-ensemble de points et réduisant la taille des matrices utilisées dans les algorithmes sous-jacents, permettent de projeter des données de grande dimension, très fréquentes dans les flux de données.

Références

Agarwal, P. et H. D. III (2010). Incremental multi-dimensional scaling. *The Learning Workshop*.

Agarwal, P. et Venkatasubramanian (2010). A unified algorithmic framework for multi-dimensional scaling. *KDD*.

Ahmad, A. et Lipika (2007). A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering 63*(2), 503–527.

Belkin, N. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation 15*, 1373–1396.

Bengio, P. et Vincent (2003). Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *Johns Hopkins University Press*.

Bennani, V. et Guérif (2008). Réduction des dimensions des données en apprentissage artificiel.

Bernstein, B. et Erofeev (2012). Comparative study of nonlinear methods for manifold learning. In *Proc. of the conf. "Information Technologies and Systems"*, pp. 85–91.

Bose, M. et Morin (2003). Fast approximations for sums of distances, clustering and the Fermat Weber problem. *Computational Geometry Theory and Application 24*, 135–146.

Chan, E. et Ching (2004). An optimization algorithm for clustering using weighted dissimilarity measures. *Pattern recognition 37*(5), 943–952.

Chang, Y. (2005). Robust locally linear embedding. *Pattern Recognition 39*, 1053–1065.

Chang, Hong, Y. (2006). Robust locally linear embedding. *Pattern recognition 39*(6), 1053–1065.

Choi (2007). Robust kernel isomap. *Pattern Recognition 40*(3), 853–862.

Deng, Z. et Lian (2010). Projection vector machine: one-stage learning algorithm from high-dimension small-sample data. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pp. 1–8. IEEE.

Du, Zhou, S. et Zhao (2012). Robust isomap based on neighbor ranking metric. *Lecture Notes in Computer Science 7389*, 221–229.

Forero (2012). Sensitivity and robustness in MDS configurations for mixed-type data : a study of the economic crisis impact on socially vulnerable spanish people. *Signal Processing,*

IEEE Transactions 60, 4118–4134.

Gaber, M. (2012). Advances in data stream mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2*(1), 79–85.

Golub, L. (1996). Matrix computations. *Johns Hopkins University Press*.

Hadid, P. (2003). Efficient locally linear embeddings of imperfect manifolds. In *Machine learning and data mining in pattern recognition*, pp. 188–201. Springer.

Hamming, R. (1950). Error detecting and error correcting codes. *Bell System technical journal 29*(2), 147–160.

Hsu, C.-C. et Chin-Long (2007). Hierarchical clustering of mixed data based on distance hierarchy. *Information Sciences 177*(20), 4474–4492.

Huang, Z. (1997). Clustering large data sets with mixed numeric and categorical values. In *Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining,(PAKDD)*, pp. 21–34. Singapore.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery 2*(3), 283–304.

Kim, L. (2004). Fuzzy clustering of categorical data using fuzzy centroids. *Pattern Recognition Letters 25*(11), 1263–1271.

Kouropteva, Okun, P. (2005). Incremental locally linear embedding. *Pattern Recognition 38*, 1764–1767.

Kruskal, W. (1978). Multidimensional scaling. *Sage University Paper series on Quantitative Application in the Social Sciences*, 07–011.

Law, J. (2006). Incremental nonlinear dimensionality reduction by manifold learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 377–391.

Law, Zhang, J. (2004). Nonlinear manifold learning for data stream. *SIAM International Conference for Data Mining*, 33–44.

Mukund (2002). The isomap algorithm and topological stability. *Science 295*, 7a.

Ng, Jordan, W. (2003). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems 14*.

Roweis, S. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 2323–2326.

Shao, H. (2005). Improvement of data visualization based on isomap. In *MICAI 2005: Advances in Artificial Intelligence*, pp. 534–543. Springer.

Shao, H. (2012). Extension of isomap for imperfect manifolds. *Journal of Computers 7*(7), 1780–1785.

Shih, Jheng, L. (2010). A two-step method for clustering mixed categroical and numeric data. *Tamkang Journal of science and Engineering 13*(1), 11–19.

Silva, T. (2003). Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems 15*, 705–712.

Silva, T. et Joshua (2002). Global versus local methods in nonlinear dimensionality reduction. In *Advances in neural information processing systems*, pp. 705–712.

Tecuanhuehue-Vera, C.-O. et Martínez-Trinidad (2012). Genetic algorithm for multidimensional scaling over mixed and incomplete data. In *Pattern Recognition*, pp. 226–235. Springer.

Tenenbaum, Silva, L. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 2319,2323.

Tu, C. (2009). Stream data clustering based on grid density and attraction. *ACM Transactions on Knowledge Discovery from Data (TKDD) 3*(3), 12.

Wan, N. et Dang (2009). Density-based clustering of data streams at multiple resolutions. *ACM Transactions on Knowledge Discovery from Data (TKDD) 3*(3), 14.

Weiss (1999). Segmentation using eigenvectors: a unifying view. *Proceedings IEEE International Conference on Computer Vision*, 975–982.

Weiszfeld (1938). Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Math. 3*, 19–22.

Wickelmaier (2003). An introduction to mds. Manuscript.

Wilson, Randall, M. (1997). Improved heterogeneous distance functions. *arXiv preprint cs/9701101*.

Zimmer, Lekadir, H. (2014). A framework for optimal kernel-based manifold embedding of medical image data. *Computerized Medical Imaging and Graphics*.

Summary

Dimension reduction methods are popular in data mining, because they facilitate the visualization and the understanding of data. Nevertheless, they cannot be applied *as is* on data streams. In this paper, we propose a brief state of the art of the main techniques of dimension reduction, in order to cope with data streams.