TP : Etude de la réduction non-linéaire de dimension par tSNE

Faïcel Chamroukhi

(draft)

Résumé

Dans ce projet TP nous étudions la réduction de dimension non-linéaire pour la visualisation de données de grande dimension par la méthode t-SNE et la comparons à l'ACP, comme méthode linéaire. Nous effectuons des expérimentations sur un jeu de données réel issue du domaine de la reconnaissance d'images. Nous abordons également la question du choix des paramètres de l'algorithme de t-SNE en vue de son automatisation.

1 Introduction

Dans cette étude nous intéressons à représenter des données $(x_1, \ldots, x_n)$ qui vivent dans un espace de grande dimension, par une représentation/transformation $(y_1, \ldots, y_n)$ dans un espace de dimension réduite, typiquement 2, pour des besoins essentiellement de visualisation. Nous nous focalisons sur les méthodes non-supervisés qui s'apprêtent également au clustering. Parmi les méthodes classiques permettant une telle réduction de dimension, on trouve l'Analyse en Composantes Principales (ACP) (Jolliffe, 2002). D'autres plus récentes sont basées sur le *stochastic neighborhood embedding* (SNE, Hinton and Roweis (2003)), comme t-SNE introduite par Maaten and Hinton (2008), que nous étudions dans ce travail et l'appliquerons à un jeu de données réel, tout en la comparant à l'ACP en terme de capacité de visualisation et de clustering. Le reste de ce rapport est composé comme suite. La section 2 présente les principes de la méthode t-SNE. Ensuite, dans la section 3, nous appliquerons t-SNE sur le jeu de données MNIST et comparons les résultats obtenus à celles fournies par l'ACP.

2 Présentation de t-SNE

Soit un ensemble de données $(x_1, \ldots, x_n)$ décrites par d variables que l'on souhaite représenter par des projetées $(y_1, \ldots, y_n)$ dans $\mathbb{R}^m$ où la dimension m de l'espace projeté est réduite ($m \ll q$), typiquement $m = 2$. Alors que l'espace projeté de l'ACP est défini à partir des composantes principales qui sont de combinaisons linéaires des variables initiales, maximisant la variance dans l'espace projeté, celui de tSNE (et aussi de SNE) résulte d'une transformation non-linéaire, minimisant la divergence entre la distribution des similarités des données dans l'espace d'origine, et celle dans l'espace projeté.
Plus précisément, le principe dont (t)SNE cherche à préserver la topologie des données d'origine dans l'espace projeté (pour que des données "similaires" dans l'espace d'origine doivent le rester dans l'espace projeté), tSNE se base sur le Stochastic Neighborhood Embedding (Hinton and Roweis, 2003) selon lequel la mesure de similarité (dans SNE) est effectuée via des probabilités conditionnelles, à travers la conversion des distances euclidiennes par un noyau Gaussien. Ainsi, dans l'espace d'entrée (resp. l'espace projecté) les distances euclidiennes $||x_i - x_j||_2$ entre les données $x_{i,j}$ (resp. les distances $||y_i - y_j||_2$ entre les données projetées $y_{i,j}$) sont converties en des probabilités conditionnelles $p_{j|i}$ (resp. $q_{j|i}$), à travers un noyau Gaussien (comme défini par l'équation (1) et (1)-bis dans Maaten and Hinton (2008)).Ainsi, afin de préserver la topologie des données de l'espace d'origine dans l'espace projeté : les données proches dans l'espace d'origine (au sens qu'elles ont des valeurs proches de distributions conditionnelles p), doivent le rester dans l'espace projeté, i.e seront représentées par des projetés ayant des valeurs proches de distributions conditionnelles q. Afin que la projection préserve cette topologie de l'espace de

données d'entrée, le principe de SNE consiste alors à minimiser la somme des divergences de Kull-Back Leibler notées

$$\mathrm{KL}(P_i||Q_i) = \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}, \tag{1}$$

entre les distributions $p_{j|i}$ des similarités dans l'espace d'origine, et celles $q_{j|i}$ des similarités des données projetées $y_{i,j}$. Cette fonction coût somme est notée ici C_{SNE} et est définie par :

$$C_{\mathrm{SNE}} = \sum_i \mathrm{KL}(P_i||Q_i). \tag{2}$$

Ce principe de SNE présente cependant deux inconvénients : (*i*) comme la divergence de KL n'est pas une fonction symétrique (i.e $KL(p||q)$ n'est pas nécessaire égal à $KL(q||p)$), cela fait que, d'après (1), des projetés très éloignées pour représenter des données proches (i.e faible $q_{j|i}$ pour modéliser un grand $p_{j|i}$) donnera une valeur élevée du coût, mais des projetés proches pour représenter des données très éloignés donnera un coût faible. Or on devrait avoir un coût similaire. Ainsi SNE favorise surtout le maintien de la structure locale des données lors de la projection. (*ii*) pour modéliser les similarités dans l'espace projeté, l'utilisation du noyau Gaussien pour convertir les distances euclidiennes peut être limitée lorsque il s'agit d'accommoder des représentations de données assez éloignées dans l'espace d'origine, versus des données très éloignées (le problème du *crowding*). Cela ne permet pas d'éviter que des représentations de données suffisament éloignées dans l'espace d'origine, se trouvent mergées dans l'espace de projection, car la gaussienne ne permet pas de couvrir une plage plus large de similarités.
Les apports essentiels de t-SNE concernent principalement ces deux points. (*i*) Tout d'abord, tSNE se base sur des probabilités jointes pour mesurer les similarités dans l'espace d'origine, et ce en additionnant (et normalisant) les probabilités conditionnelles (principe de SNE) par

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n},$$

et en modélisant directement les probabilités jointes q_{ij}.Cette astuce permet de rendre symétrique la mesure de similarité d'un espace à un autre. Ensuite, plutôt que de minimiser une somme de divergences de KL (comme pour SNE dans (3)), t-SNE minimise une seule divergence de KL entre les distributions jointes des similarités d'origines P et celle des projetées Q : La divergence de KL minimisées est alors

$$C_{t\mathrm{SNE}} = \mathrm{KL}(P||Q), \text{ avec } \mathrm{KL}(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{3}$$

(*ii*) Pour le deuxième point, la modélisation des similarités dans l'espace projeté est réalisé à travers un noyau t-Student (à un degré de liberté) normalisé, au lieu du noyau Gaussien, pour calculer directement les probabilités jointes q_{ii}. Ceci permet d'atténuer l'effet du problème de *crowding*, car la loi student a une queue plus lourde que celle de la gaussienne. De plus, cette modélisation amène à une forme plus simple du gradient de la fonction coût, ce qui facilité son optimisation.
Apprentissage : La fonction coût (3) ne peut pas être optimisée de façon exacte ; la fonctionnelle qui régie la projection entre l'espace des données et l'espace projeté de t-SNE, n'a pas de forme analytique paramétrique, elle est non-paramétrique. Elle est de plus potentiellement multimodale surtout en grande dimension. La fonction coût est alors minimisée itérativement par une descente de gradient. Cette descente est initialisée aléatoirement à partir d'une loi Gaussienne centrée de petite variance.
Il est ainsi pertinent, tout comme pour tout problème d'optimisation locale par descente de gradient, d'effectuer plusieurs descentes à partir de différentes initialisations (voire avec différents pas de descente), ce qui permet de visiter différents minimas-locaux de la fonction coût, afin de n'en garder que le meilleur (celui donnant la plus petite divergence de Kull-Back ici)

3 t-SNE versus PCA

Dans cette partie on étude les capacités de réduction de dimension de t-SNE Maaten and Hinton (2008) et on la compare à l'Analyse en Composantes Principales (ACP). Dans un premier temps, on considère

pour t-SNE une paramétrisation de choix dans laquelle la perplexité `perp` et le nombre d'itérations de l'algorithme du descente de gradient sont fixés sur la base d'une inspection visuelle des résultats, notamment la pertinence de la représentation fournie et à quel point elle pourrait accommoder l'objectif d'une classification non-supervisée (clustering). Ainsi, la perplexité dans une plage de valeur de 5 à 50, ce qui est une suggestion de Maaten and Hinton (2008), qui semble raisonnable pour les données analysées. Le nombre d'itérations est quand à lui choisi "suffisamment" grand (de 1000 jusqu'à 5000 itérations), puisque, s'agissant d'une descente de gradient, le nombre d'itérations avant la convergence reste un problème difficile à évaluer de façon optimale, contrairement à la perplexité qui reste un paramètre intrinsèque à la modélisation et, malgré que cela semble rester une question de recherche ouverte, des approches récentes ont été proposées dans la littérature, comme dans Cao and Wang (2017) et Belkina et al. (2019)). Nous détaillerons cela dans la deuxième partie.
Lorsque nous effectuons la représentation des données dans l'espace t-SNE, nous considérons les données originales représentées dans l'espace de l'ACP en gardant le nombre de composantes principales optimales, au sens que celles-ci préservent un pourcentage fixé (ici 95%) de la variance cumulée dans l'espace projeté. Bien entendu, ceci est fait par soucis de diminuer les temps de calcul, le calcul de l'ACP peut toujours être laissé à se faire au sien de la méthode t-SNE elle-même.
Dans cet exemple, les données utilisées sont le jeu de données de test de la base MNIST (LeCun and Cortes, 2010). Ce jeu de données comprend $n = 10.000$ exemples de dimension $p = 784$ et couvre $K = 10$ classes (images en niveau de gris (28x28 pixels) de chiffres manuscrits (0 à 9). La figure 1 montre le nombre de composantes principales `nb_PC` retenues pour ce jeu de données. La figure 2

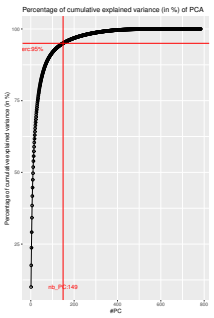

FIGURE 1 – Choosing optimal number of principal components

montre la visualisation des données dans l'espace de l'ACP défini par les deux premières composantes principales (graphique de gauche) et dans l'espace projeté 2-d t-SNE (graphique de droite). On peut constater que ce dernier met plus en évidence la nature hétérogène des données à travers la préservation de la formation de groupes pour les données. Cependant, pour l'ACP, les données qui dans l'espace d'origine exhibent bien une notion de clustering car présentent par nature plusieurs groupes, la ne préserve pas (assez) cette dispersion. Pour ce jeux de données nous avons utilisés tSNE avec une valeur

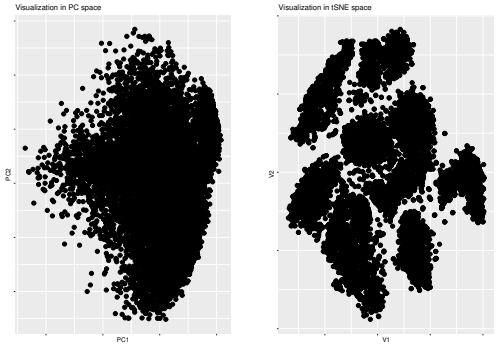

FIGURE 2 – Data embedded into the PCA space (left) and the t-SNE space (right)

de 30 pour la perplexité et 2000 itérations pour l'algorithme de descente de gradient.
La figure 3 représente les données dans l'espace des composantes principales (graphique de gauche) et dans celui de t-SNE (graphique de droite) selon la densité locale des points (en utilisant la fonction

densCols). On peut constater que alors que l'ACP fait apparaître que la densité des données est concentrée autour de principalement trois modes de densités locales, t-SNE met clairement en évidence dix modes de densités, ce qui correspond bien aux nombres de classes des données d'origine. Ainsi, la structure latente de ces données hétérogènes dans l'espace d'origine est mise plus en évidence par t-SNE que par l'ACP. t-SNE s'apprête ainsi plus au clustering de telles données.

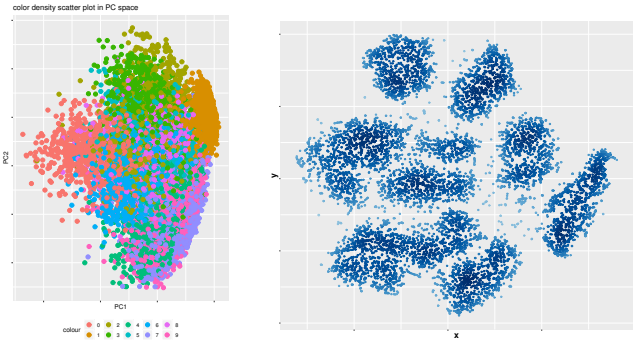

FIGURE 3 – Scatter data density visualized on the PC space (left) and on the t-SNE space (right)

Ceci peut être vu plus clairement lorsque l'on colore les données selon leur vraie appartenance aux classes comme le montre les deux graphiques de la figure 4

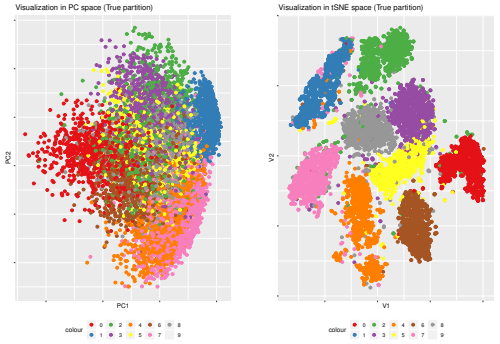

FIGURE 4 – True data partition visualized on the PC space (left) and on the t-SNE space (right)

Ainsi, la figure 5 montre que la structure en classes sous-jascente est bien préservée et mise en évidence par tSNE, alors que pour l'ACP la plupart des classes sont clairement inséparables (se chevauchent beaucoup). Dans ce graphique, le clustering a été effectué dans l'espace de l'ACP en sélection le nombre de composantes principales qui préserve 95% du pourcentage de la variance expliquée comme montré dans le graphique 1.
La figure 5 (respectivement figure 6) montre la partition des composantes principales (réduites) (respectivement. partition des données d'origines) obtenue par l'algorithme K-means (redavec $K = 10$) visualisée dans l'espace de l'ACP (graphique de gauche) et dans celui de t-SNE (graphe de droite). On peut y voir que la structure en classes de données est plus conforme à la nature des données (i.e mise en évidence des K classes) pour t-SNE alors que pour l'ACP l'hétérogénéité n'a pas été préservée par la projection.

Références

Anna C Belkina, Christopher O Ciccolella, Rina Anno, Richard Halpert, Josef Spidlen, and Jennifer E Snyder-Cappione. Automated optimized parameters for t-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nature communications*, 10(1) :1–12, 2019.

Yanshuai Cao and Luyu Wang. Automatic selection of t-sne perplexity. *arXiv preprint arXiv :1708.03229*, 2017.

Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pages 857–864, 2003.

Ian Jolliffe. *Principal component analysis*. Springer Verlag, New York, 2002.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

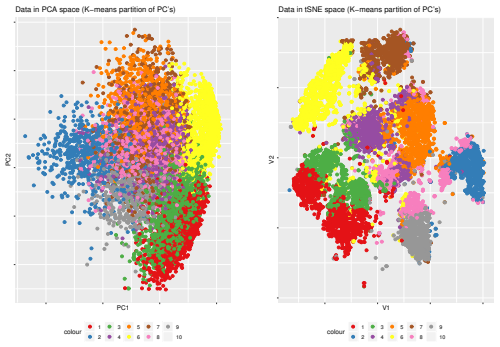

FIGURE 5 – Kmeans partition of the PCA data representation, after keeping the "optimal" number of principal components visualized on the PC space (left) and on the t-SNE space (right)

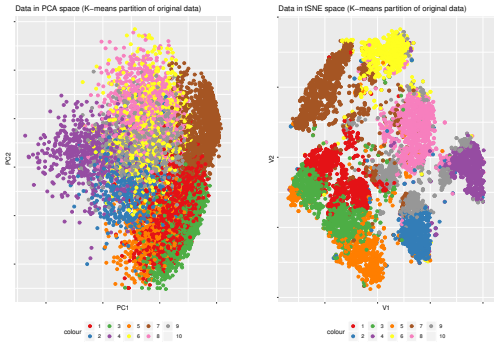

FIGURE 6 – Kmeans partition of the original data visualized on the PC space (left) and on the t-SNE space (right)

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov) :2579–2605, 2008.