TP: Study of Nonlinear Dimensionality Reduction and DataViz using t-SNE

Faïcel Chamroukhi

(draft, from an origianl french version)

Abstract

In this TP project, we study nonlinear dimensionality reduction for high-dimensional data visualization using the t-SNE method and compare it with PCA as a linear method. We perform experiments on a real dataset from the field of image recognition.

1 Introduction

In this study, we aim to represent data $(x_1, \ldots, x_n)$ that live in a high-dimensional space through a representation/transformation $(y_1, \ldots, y_n)$ in a lower-dimensional space, typically 2, primarily for visualization purposes. We focus on unsupervised methods that are also suitable for clustering.

Among classical methods for such dimensionality reduction, we find Principal Component Analysis (PCA) (Jolliffe, 2002). More recent methods are based on Stochastic Neighborhood Embedding (SNE, Hinton and Roweis (2003)), such as t-SNE introduced by Maaten and Hinton (2008), which we study in this work. We apply it to a real dataset while comparing it with PCA in terms of visualization and clustering capabilities.

2 Presentation of t-SNE

Let a dataset $(x_1, \ldots, x_n)$ described by d variables that we wish to represent through projections $(y_1, \ldots, y_n)$ in $\mathbb{R}^m$ where the dimension m of the projected space is reduced ($m \ll q$), typically $m = 2$. While the projected space of PCA is defined based on principal components, which are linear combinations of the initial variables maximizing variance in the projected space, that of tSNE (and also SNE) results from a non-linear transformation, minimizing the divergence between the similarity distribution of the data in the original space and that in the projected space.

More precisely, the principle of (t)SNE is to preserve the topology of the original data in the projected space (so that "similar" data in the original space remain similar in the projected space). tSNE is based on Stochastic Neighborhood Embedding (Hinton and Roweis, 2003), where the similarity measure (in SNE) is performed using conditional probabilities, through the conversion of Euclidean distances by a Gaussian kernel. Thus, in the input space (resp. the projected space), the Euclidean distances $||x_i - x_j||_2$ between the data $x_{i,j}$ (resp. the distances $||y_i - y_j||_2$ between the projected data $y_{i,j}$) are converted into conditional probabilities $p_{j|i}$ (resp. $q_{j|i}$), through a Gaussian kernel (as defined by equation (1) and (1)-bis in Maaten and Hinton (2008)). Thus, to preserve the topology of the original data in the projected space: data close in the original space (in the sense that they have close values of conditional distributions p), must remain so in the projected space, i.e., they will be represented by projections having close values of conditional distributions q.

To ensure that the projection preserves this topology of the input data space, the principle of SNE then consists in minimizing the sum of Kull-Back Leibler divergences noted

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

between the distributions $p_{j|i}$ of similarities in the original space, and those $q_{j|i}$ of similarities of the projected data $y_{i,j}$. This sum cost function is noted here as C_{SNE} and is defined by:

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

However, this principle of SNE presents two drawbacks: (*i*) since KL divergence is not a symmetric function (i.e., $KL(p||q)$ is not necessarily equal to $KL(q||p)$), this implies that, from (1), projections that are very distant to represent close data (i.e., low $q_{j|i}$ to model a large $p_{j|i}$) will give a high cost value, whereas projections close to represent very distant data will give a low cost value. Yet we should have a similar cost. Thus, SNE mainly favors maintaining the local structure of the data during projection. (*ii*) To model similarities in the projected space, using a Gaussian kernel to convert Euclidean distances can be limited when it comes to accommodating representations of data that are quite distant in the original space versus very distant data (the *crowding* problem). This does not prevent sufficiently distant data representations in the original space from being merged in the projection space because the Gaussian does not allow covering a wider range of similarities.

The essential contributions of t-SNE mainly concern these two points. (*i*) First, tSNE is based on joint probabilities to measure similarities in the original space, and this by adding (and normalizing) conditional probabilities (SNE principle) by

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

and by directly modeling the joint probabilities q_{ij}. This trick makes the similarity measure from one space to another symmetric. Then, instead of minimizing a sum of KL divergences (as for SNE in (3)), t-SNE minimizes a single KL divergence between the joint distributions of original similarities P and that of the projected Q: The minimized KL divergence is then

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

(*ii*) For the second point, the modeling of similarities in the projected space is performed using a normalized t-Student kernel (with one degree of freedom), instead of the Gaussian kernel, to directly compute the joint probabilities q_{ii}. This helps mitigate the effect of the *crowding* problem because the Student distribution has a heavier tail than the Gaussian. Additionally, this modeling leads to a simpler form of the cost function gradient, which facilitates its optimization.

Learning: The cost function (3) cannot be optimized exactly; the functional governing the projection between the data space and the projected space of t-SNE has no parametric analytical form, it is non-parametric. Moreover, it is potentially multimodal, especially in high dimensions.

The cost function is then minimized iteratively using gradient descent. This descent is randomly initialized from a centered Gaussian distribution with small variance.

It is therefore relevant, as for any local optimization problem by gradient descent, to perform multiple descents from different initializations (or even with different descent steps), which allows visiting different local minima of the cost function, in order to retain only the best one (the one yielding the smallest Kull-Back divergence here).

3 t-SNE versus PCA

In this part, we study the dimensionality reduction capabilities of t-SNE Maaten and Hinton (2008) and compare it to Principal Component Analysis (PCA).

Initially, we consider a specific parameterization for t-SNE, in which the perplexity `perp` and the number of iterations of the gradient descent algorithm are set based on a visual inspection of the results, particularly assessing the quality of the provided representation and how well it accommodates the goal of unsupervised classification (clustering).

Thus, perplexity is chosen within a value range of 5 to 50, as suggested by Maaten and Hinton (2008), which appears reasonable for the analyzed data.

The number of iterations, on the other hand, is chosen to be “sufficiently” large (from 1000 up to 5000 iterations), since, in the case of gradient descent, the number of iterations before convergence remains a difficult problem to evaluate optimally, unlike perplexity, which remains an intrinsic parameter to the modeling. Despite this, it seems to remain an open research question, and recent approaches have been proposed in the literature, such as in Cao and Wang (2017) and Belkina et al. (2019). We will detail this in the second part.

When performing the representation of data in the t-SNE space, we consider the original data represented in the PCA space while keeping the optimal number of principal components, in the sense that these preserve a fixed percentage (here 95%) of the cumulative variance in the projected space. Of course, this is done to reduce computation times, as PCA computation can always be performed within the t-SNE method itself.

In this example, the data used is the test dataset from the MNIST database (LeCun and Cortes, 2010). This dataset consists of $n = 10,000$ examples of dimension $p = 784$ and covers $K = 10$ classes (grayscale images (28x28 pixels) of handwritten digits (0 to 9).

The figure 1 shows the number of principal components `nb_PC` retained for this dataset.

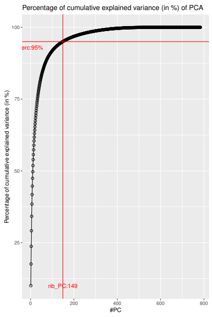

Figure 1: Choosing the optimal number of principal components

The figure 2 shows the visualization of the data in the PCA space defined by the first two principal components (left plot) and in the projected 2D t-SNE space (right plot). We can see that the latter highlights the heterogeneous nature of the data through the preservation of group formation in the data. However, for PCA, data that exhibit a clear notion of clustering in the original space, as they naturally contain multiple groups, does not preserve (enough) this dispersion.

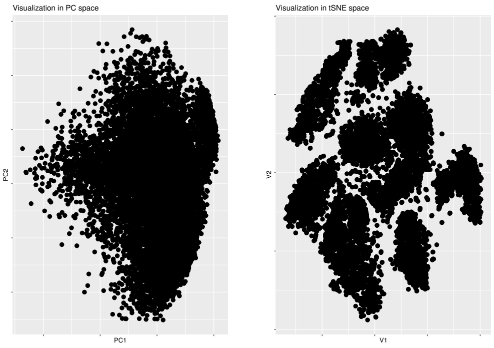

Figure 2: Data embedded into the PCA space (left) and the t-SNE space (right)

For this dataset, we used tSNE with a perplexity value of 30 and 2000 iterations for the gradient descent algorithm.

The figure 3 represents the data in the principal component space (left plot) and in the t-SNE space (right plot) according to the local density of points (using the `densCols` function).

We can observe that, while PCA reveals that the data density is concentrated around primarily $\sim$ three modes of local densities, t-SNE clearly highlights ten density modes, which precisely corresponds to the number of classes in the original data. Thus, the latent structure of these heterogeneous data in the original space is better revealed by t-SNE than by PCA. t-SNE is therefore more suited for clustering such data.

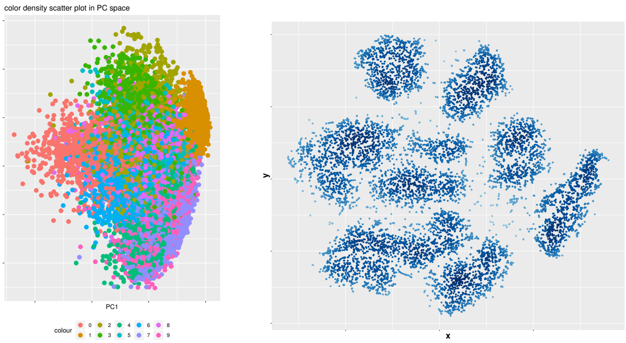

Figure 3: Scatter data density visualized on the PC space (left) and on the t-SNE space (right)

This can be seen more clearly when we color the data according to their true class membership, as shown in the two plots of figure 4.

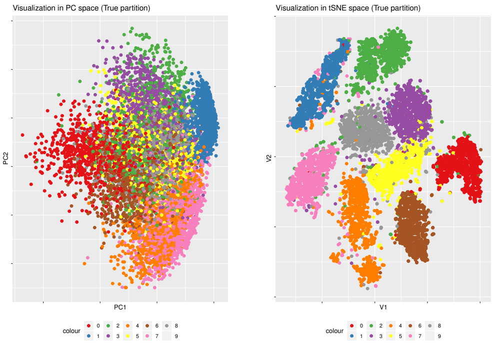

Figure 4: True data partition visualized on the PC space (left) and on the t-SNE space (right)

References

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/`.

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