Visualizing Data using t-SNE - An overview

Jan Ruman

21.5.2021

Abstract

This document attempts to summarize the Visualizing Data using t-SNE [2] research paper by Maaten et al.

1 Introduction

The main goal of data visualization is to find a projection to 2 or 3 dimensions that preserves as much of the structure of data as possible. Methods attempting to visualize data should be able to cluster similar data together while keeping the clusters themselves clearly separated from each other.

In their work Maaten et al. introduce t-SNE, a new non-parametric method for high dimensional data visualization. The method is based on one of the previous works, Stochastic Neighborhood Embedding [1]. In both methods high- and low-dimensional data structure is modelled by probability distributions whose difference induces a cost function, which is then optimized using gradient descent.

At the time of publishing this method was able to outperform previous approaches to data visualization on multiple benchmark datasets.

2 t-SNE

The t-SNE algorithm is based on the Stochastic Neighborhood Embedding method. We will first describe SNE and then mention in what respects does t-SNE differ.

2.1 Stochastic neighborhood estimation

To retain the structure of the data SNE attempts to find a mapping such that for every two points the difference of distances between the points in high and low dimension is as small as possible; the metric used is one of the core concepts of the work. The distance $p_{j|i}$ from x_j to x_i, where x_i, x_j are both high-dimensional points, is defined as the value of a Gaussian with mean x_i and variance σ_i^2 at the point x_j. Formally

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||x_i - x_k||^2/2\sigma_i^2)}$$

Notice the distance is not symmetrical, i.e. $p_{i|j} \neq p_{j|i}$. The distance $q_{j|i}$ between two low-dimensional points y_j and y_i is defined similarly; the only difference is that all the Gaussians in lower dimension have the same variance $\frac{1}{\sqrt{2}}$.

Let P_i and Q_i be probability distributions over points x_i in high dimension and its mapping y_i in low dimension, respectively. An ideal mapping would minimize the sum of differences between distributions P_i and Q_i over the whole dataset. To measure the difference between distributions the authors use Kullback-Leibler divergence. We have thus obtained a cost function, which we can now optimize using gradient descent.

It remains to describe how the variances for Gaussians in high dimension are selected. To obtain a variance for P_i a binary seach is performed; a variance is selected if the distribution attains a given perplexity $Perp(P_i)$, specified by user. Formally, perplexity is given by

$$Perp(P_i) = 2^{H(P_i)}$$

where $H(P_i)$ is the Shannon entropy of P_i. Perplexity can be thought of as a smooth measure of number of the effective number of neighbors of the point the distribution is centered around. Some experimentation might be needed to find the ideal value of perplexity.

At the beginning of the training Gaussian noise is added to map points to escape poor local optima. Map points are initialized randomly from an isotropic Gaussian with small variance centered around the origin.

2.2 t-SNE

The first core difference between SNE and t-SNE is the use of symmetrical distance; the way symmetry is achieved differs depending on whether we are dealing with high- or low-dimensional data. The approach for low-dimensional data resembles joint probability distribution - when calculating the distance between two points we take into account all the distances.

$$q_{ij} = \frac{\exp(-||y_i - y_j||^2)}{\sum_{k \neq l} \exp(-||y_k - y_l||^2)}$$

This approach does not work very well for high-dimensional data. To make distances for high-dimensional data symmetrical, we simply normalize them.

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

The second major change brought by t-SNE is given by its name - the use of Student t-distribution for low dimensional points. Using Student t-distribution

Technique	Cost function parameters
t-SNE	$Perp = 40$
Sammon mapping	none
Isomap	$k = 12$
LLE	$k = 12$

Table 1: The hyperparameter setup.

with single degree of freedom, we get the following distance between two low-dimensional points:

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}$$

Now we are ready to write the gradient of the Kullback-Leibler divergence between P and Q:

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1}$$

There are a few more minor differences between SNE and t-SNE we won't be covering here.

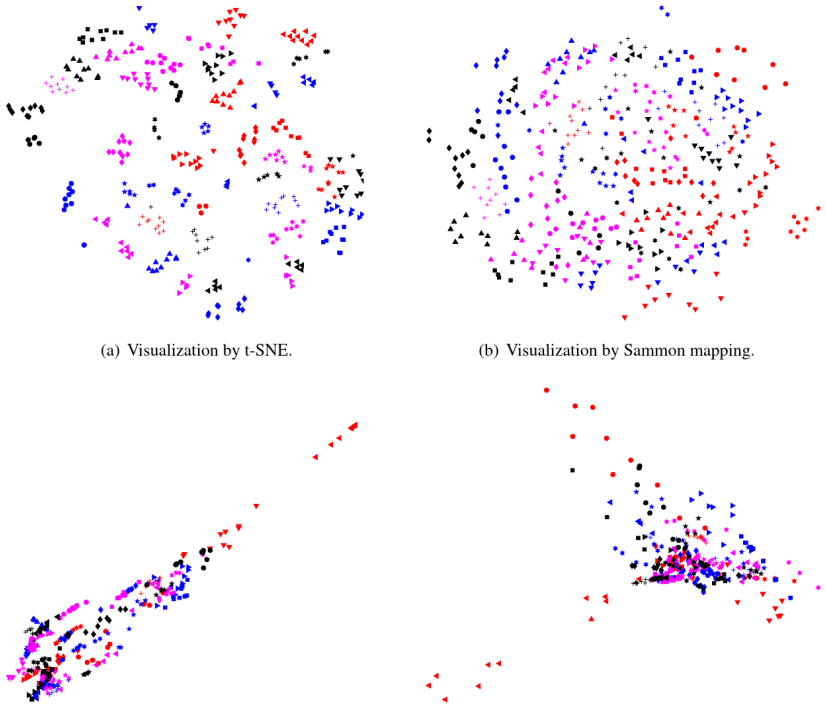
It is also worth mentioning that the t-SNE algorithm runs in quadratical time with respect to the number of examples in the dataset. For bigger datasets (10 000 and more examples) the authors thus suggest using random walks on neighborhood graphs to compute P.

3 Experiments

The authors have evaluated t-SNE against other contemporary non-parametric methods for data visualization, namely Sammon mapping, Isomap, LLE, CCA, SNE, MVU and Laplacian Eigenmaps. Showing visualizations produced by all the methods on all five datasets would be too space consuming, therefore the authors decided to show only some of the visualizations. For this summary we will use only one of the datasets and compare the performance of t-SNE, Sammon mapping Isomap and LLE. These results constitute sufficient representation of the effectiveness of individual methods for the purposes of this work. The results can be seen in Figure 1.

We shall now examine the exact experimental setup used to achieve those results. The dataset dimensionality is first reduced to 30 using PCA. The data is stripped of class information before the methods are applied to it; it is however used in the visualization itself to distinguish points belonging to different classes. Table 1 shows the hyperparameters used for invidivual methods.

As for the results themselves, it is clear that t-SNE performed the best from all the methods. Both Isomap and LLE fail to properly separate more



(a) Visualization by t-SNE.

(b) Visualization by Sammon mapping.

(c) Visualization by Isomap.

(d) Visualization by LLE.

Figure 1: Visualizations of the Olivetti faces dataset. Taken from the original t-SNE paper [2]

than several classes. Sammon mapping manages to cluster similar examples together, but it fails to separate the clusters from one another, in some cases even leading to clusters overlapping to great extent. On the other hand t-SNE is successful in both clustering similar data points together as well as separating the clusters.

4 Conclusions

The t-SNE method is a data visualization technique that manages to properly visualize high-dimensional data in two dimensions. The authors have provided experimental results showing its superiority to its predecessors.

References

[1] Geoffrey Hinton and Sam T Roweis. Stochastic neighbor embedding. In *NIPS*, volume 15, pages 833–840. Citeseer, 2002.

[2] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.