t-Distributed Stochastic Neighbor Embedding

Mario De Simone

Faculty of Artificial Intelligence
University of Florence

July 2023

Table of Contents

Introduction

t-SNE is a non-linear dimensionality reduction algorithm, to map a set of points into a lower dimension space (usually 2D or 3D embedding).
It is an improvement over the previous Stochastic Neighbor Embedding.

t-SNE Idea

The idea is simple yet effective:

- Similarity distance across point in first space (**Normal Distribution**)
- Similarity distance across point in second space (**t-Student Distribution**)

Map to preserve similarity, it is worth notice that these distance (as shown in the following), try to preserve the local structure of the data in contrast to algorithm like PCA (which is a linear technique)

t-SNE Algorithm

Algorithm 1 t-SNE

```
Require: $X$ set of observations, $maxIter$, desired dimension $dim$ and desired perplexity $perp$
Compute the similarity pairwise distance for each point in the starting space with given perplexity (P-Matrix)
while $iter < maxIter$ or stopping-criteria do
    Compute the similarity pairwise distance for each point in the space of arrive (Q-Matrix)
    Compute gradient of cost function with respect to the point embedding of the current iteration
    Compute a step of gradient descent with momentum
    $iter = iter + 1$
end while
```

Condtional P-Matrix I

To compute the P-Matrix we have to get the conditional probabilities (similarities) between each pair of points:

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum\limits_{k \neq i} \exp(-||x_k - x_l||^2/2\sigma_i^2)} \tag{1}$$

we impose $p_{i|i} = 0$
Problem: As we can see we have somehow to infer the variance σ_i^2.
Solution: Given the perplexity of the data, find the variance with similar perplexity

Conditional P-Matrix II

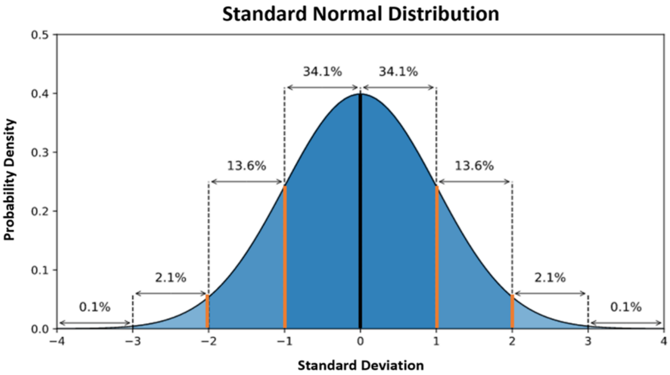


Figure: Normal Distribution

Perplexity

The perplexity is a parameter that reflects the density structure of the data:

$$perp(p) = 2^{-\sum_x p(x) \log_2 p(x)} \tag{2}$$

At this point the idea to get σ_i^2 is to perform a binary search in order to find the most appropriate variance value.
We can notice that this operation is computationally expensive but luckily it has to be executed just once (the entire P-Matrix is computed once)

P-Matrix

Finally we want the P-Matrix to be symmetric and scaled appropriately to avoid too small contributes:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \tag{3}$$

with N total number of samples.
As already mentioned the computation of the P-Matrix is very expensive in the following some ideas to improve will be discussed.

Q-Matrix I

The Q-Matrix is the biggest change with respect to the SNE algorithm, instead of compute normal distributed distances, we use the t-Student (the longer tails ensure softer penalizations with respect to the normal distribution):

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq i}(1 + ||y_k - y_l||^2)^{-1}} \tag{4}$$

again we impose $q_{ii} = 0$

Q-Matrix II

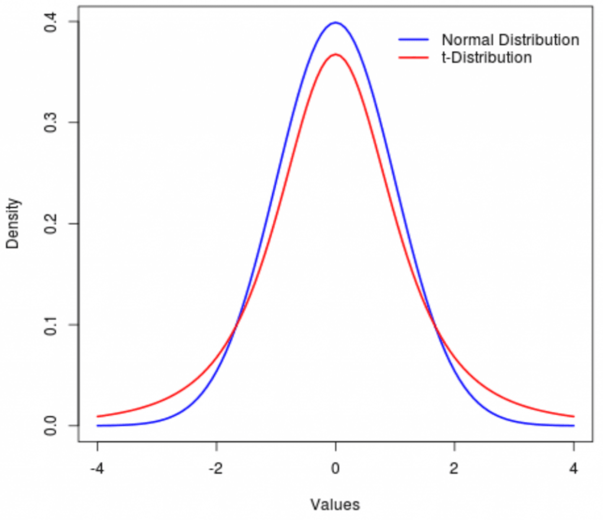


Figure: Normal vs t-Student Distribution

Cost Function

Now that we have defined the distributions, we have to choose a measure of distance between P and Q to verify and improve, using an iterative approach the similarity, we use the KL divergence:

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{5}$$

it can be proved that taking the derivatives with respect to the point of the current embedding y_i is:

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + ||y_i + y_j||^2)^{-1} \tag{6}$$

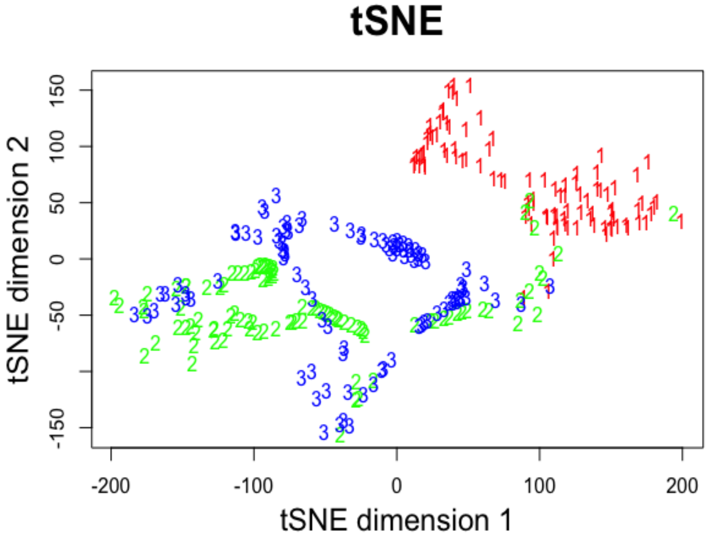
Experiments I

We tried the algorithm on 2 different datasets:

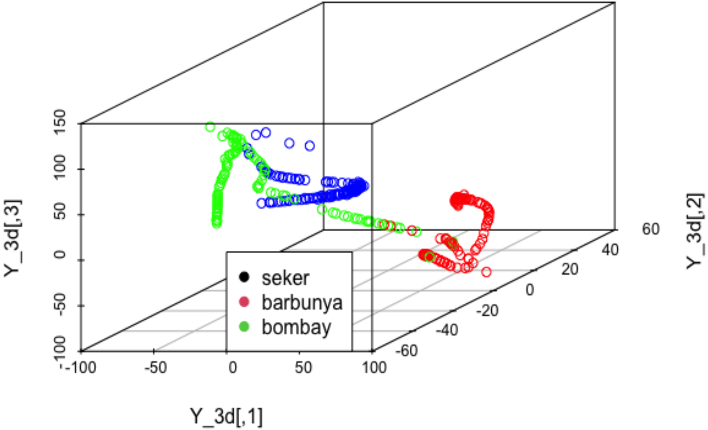
- iris: a collection of different kind of iris with their observed characteristics (150 observations of 5 variables)
- beans: a collection of different kind of beans with their observed characteristics (13611 observations of 17 variables)

It is worth noticing that due to the slow implementation and the effective dimension of the beans dataset it has been necessary to choose a subpopulation and less class on which try the algorithm.
In the following the plot 2D and 3D obtained with the algorithm.

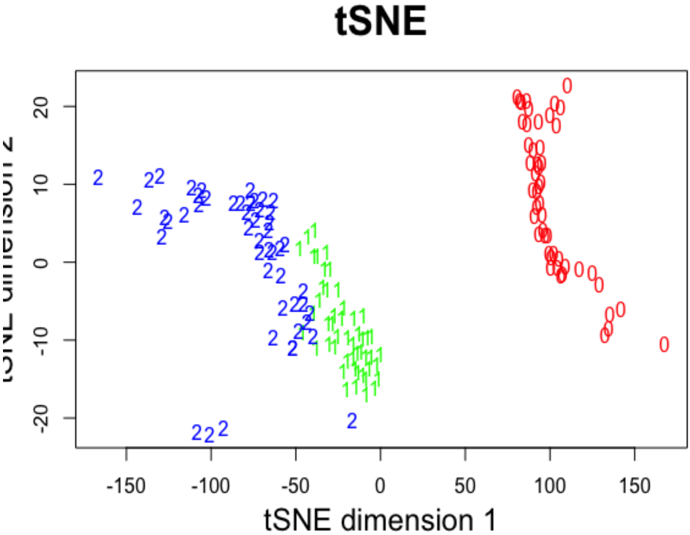
Experiments II



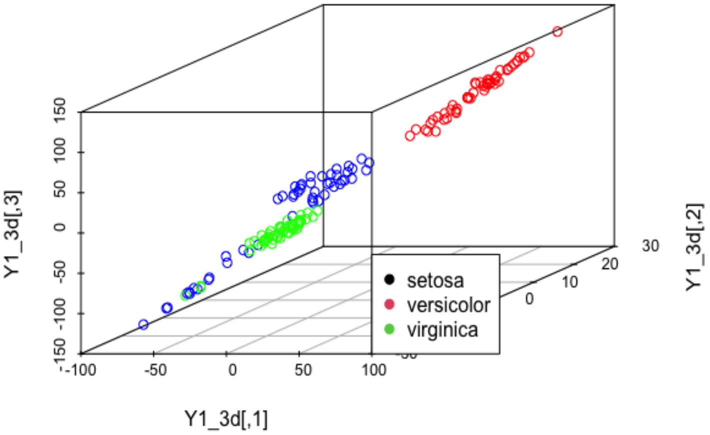
Experiments III



Experiments IV



Experiments V



Evaluation Metric Idea

In addition to cost function we introduced a simple idea to evaluate the separation of the embedding, a double measure:

- max distance between points of different clusters
- min distance between points of different clusters

this is not a precise information but it gives an hint about the separation of structeres in the data.

Complexity

The implementation made of the algorithm is really slow, but the algorithm is computationally expensive.
The complexity become worse with the increment of the number of samples and the dimension of the points, there are different ways to improve its time complexity:

- choose a stopping criteria different from the maxIter one
- choose arbitrarily σ_i^2
- choose a sub-sample population on which compute the pairwise distances
- perform PCA before the t-SNE
- make use of tensors structures

Change Distances

As shown in the previous equations the distance used is the euclidean one, it can be useful change that distance according to background knowledge of the data, in the following i show the usage of manhattan distance

Conclusion

The t-SNE achieves wonderful performance, making the data more readable, its reduction dimensionality tend to isolate clusters. Moreover the possibility to change the distance used and the adaptive search for σ_i^2 make the algorithm versatile, the only downside is its expensiveness that can be mitigated in different ways.