t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm

Mathilde Boissel
UMR 1283 (INSERM) UMR 8199 (CNRS) (Université de Lille / Institut Pasteur de Lille / CHU Lille)
(Épi)génomique Fonctionnelle et Physiologie Moléculaire du Diabète et Maladies Associées

E.G.I.D - FR 3508 (CNRS / Université de Lille 2 / Institut Pasteur de Lille / CHRU)
European Genomics Institute for Diabetes

E-mail: mathilde.boissel@univ-lille.fr / mathilde.boissel@cnrs.fr

06/04/2020

Concept : t-SNE.

- "… is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets." *L. Van Der Maaten* *https://lvdmaaten.github.io/tsne/*

The aim of dimensionality reduction is to preserve as much of the significant structure of the high-dimensional data as possible in the low-dimensional map.

high-dimensional data set → two or three-dimensional data

$$\mathcal{X} = \{x_1, x_2, \ldots, x_n\} \longrightarrow \mathcal{Y} = \{y_1, y_2, \ldots, y_n\}$$

Concept : t-SNE.

- "… is a technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets." *L. Van Der Maaten https://lvdmaaten.github.io/tsne/*

The aim of dimensionality reduction is to preserve as much of the significant structure of the high-dimensional data as possible in the low-dimensional map.

≠ from Principal Components Analysis (PCA; Hotelling, 1933) and multidimensional scaling (MDS; Torgerson, 1952, also known as Principal Coordinates Analysis "PCoA") which are linear techniques that focus on keeping the low-dimensional representations of dissimilar datapoints far apart. (the variance is maximized)

"For high-dimensional data that lies on or near a low-dimensional, non-linear manifold it is usually more important to keep the low-dimensional representations of very similar datapoints close together, which is typically not possible with a linear mapping."

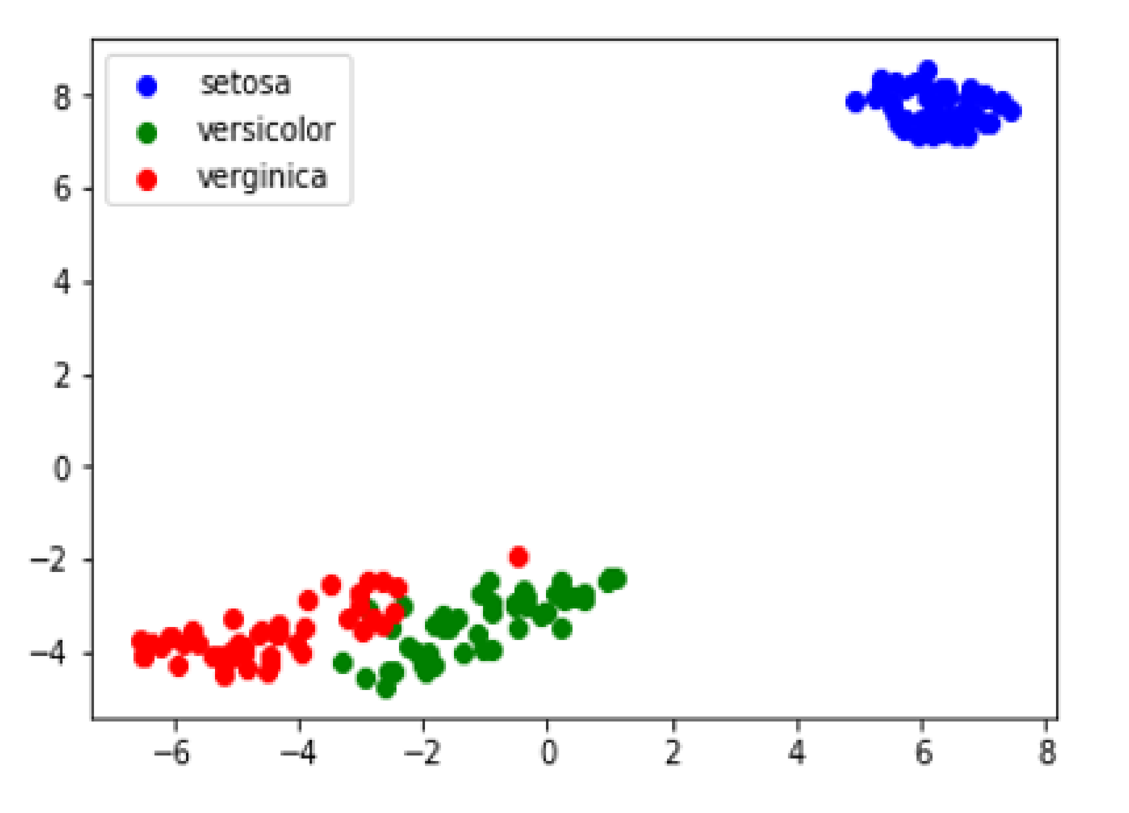
Concept : t-SNE.

➢ " also revealing global structure such as the presence of clusters at several scales. "

L. Van Der Maaten https://lvdmaaten.github.io/tsne/

The aim of clustering is to group a set of objects in the same group (called a cluster) given similarities measures.

So it belongs to the "unsupervised learning" algorithm that looks for underlying patterns in a data set with no pre-existing labels



Stochastic Neighbor Embedding (SNE)

Minimize an objective function that measures the discrepancy between similarities in the data and similarities in the map

1/ Converts the high-dimensional Euclidean distances between datapoints into conditional probabilities that represent similarities. where σ is the variance of the Gaussian that is centered on datapoint x_i

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

2/ Defines a similar conditional probability for the low-dimensional counterparts

$$q_{j|i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k \neq i} \exp\left(-\|y_i - y_k\|^2\right)}$$

If the map points y_i and y_j correctly model the similarity between the high-dimensional datapoints x_i and x_j, the conditional probabilities *p* and *q* will be equal.

3/ The cost function C => minimizes the sum of Kullback-Leibler divergences over all datapoints using a gradient descent method.

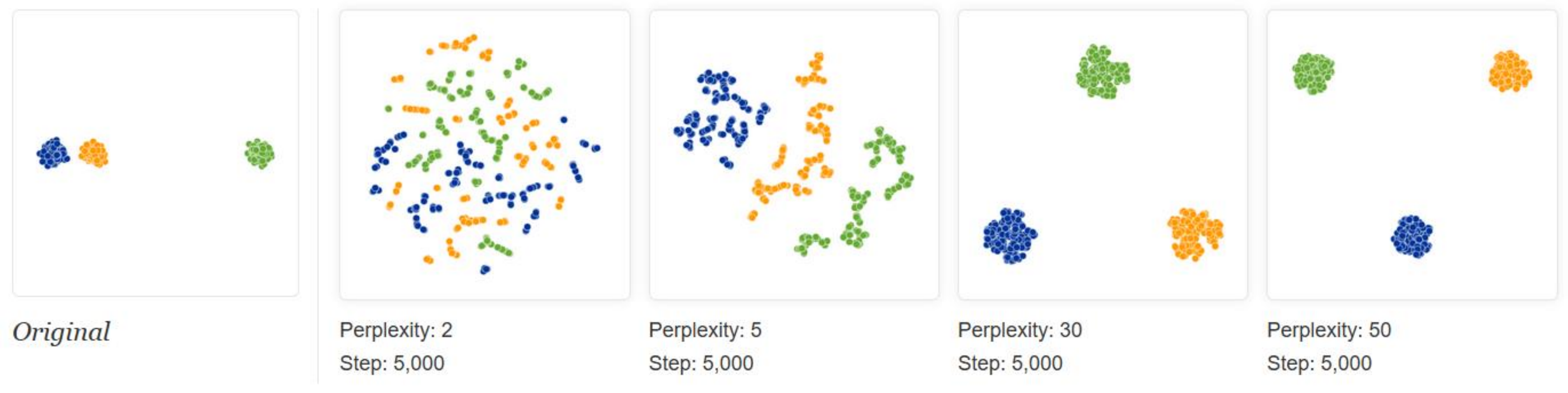
the Kullback-Leibler divergence

$$C = \sum_i KL(P_i || Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

Stochastic Neighbor Embedding (SNE)

" Any particular value of σ_i induces a probability distribution, *Pi*, over all of the other datapoints.
This distribution has an entropy which increases as σ_i increases.
SNE performs a binary search for the value of σ_i that produces a *Pi* with a fixed perplexity that is specified by the user.
The perplexity $Perp(P_i) = 2^{H(P_i)}$ where *H*(*Pi*) is the Shannon entropy of *Pi* measured $H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}$

The performance of SNE is fairly robust to changes in the perplexity, and typical values are between 5 and 50. "



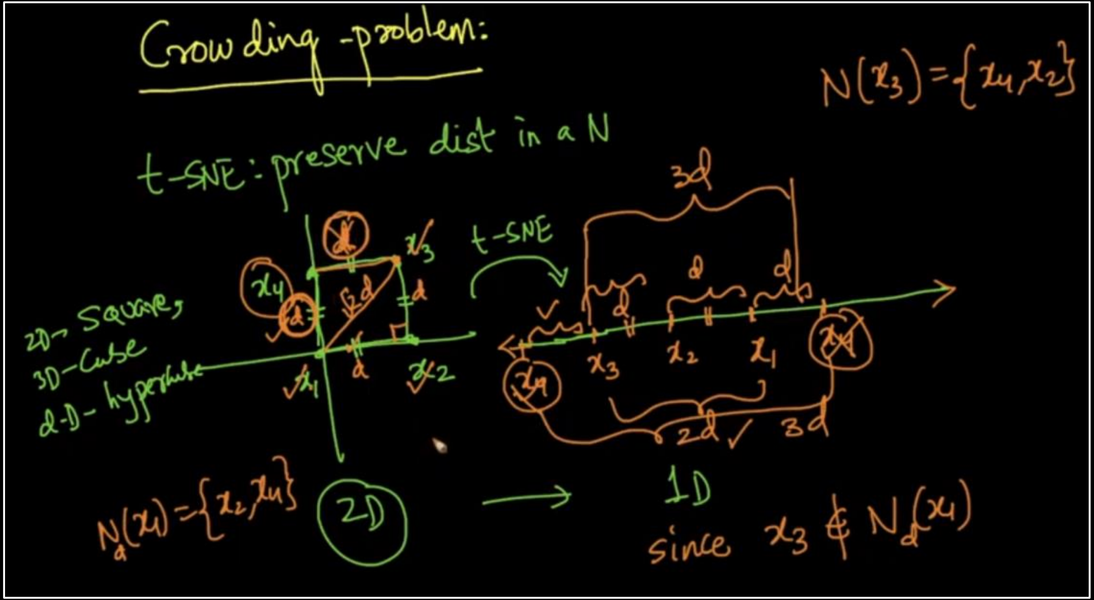
Perplexity : can be interpreted as a smooth measure of the effective number of neighbors
(information) Entropy : measure of disorder or uncertainty in a system

Problems

➢ " <u>Crowding</u> problem " :
"the area of the two-dimensional map that is available to accommodate moderately distant datapoints will not be nearly large enough compared with the area available to accommodate nearby datapoints."

Points tend to "crowd" together in the center of the map

"it is impossible to preserve distances in all the neighborhoods"



Watch this video for simple example:
https://www.youtube.com/watch?v=hMUrZ708PFk

➢ Kullback-Leibler divergence is asymmetric :
" Because the Kullback-Leibler divergence is not symmetric, different types of error in the pairwise distances in the low-dimensional map are not weighted equally "

t-SNE : How is different from SNE ?

(1) it uses a symmetrized version of the SNE cost function with simpler gradients

➔ " [Symmetric SNE …] allows a moderate distance in the high-dimensional space to be faithfully modeled by a much larger distance in the map and, as a result, it eliminates the unwanted attractive forces between map point "

➔ " Natural way of alleviating the crowding problem "

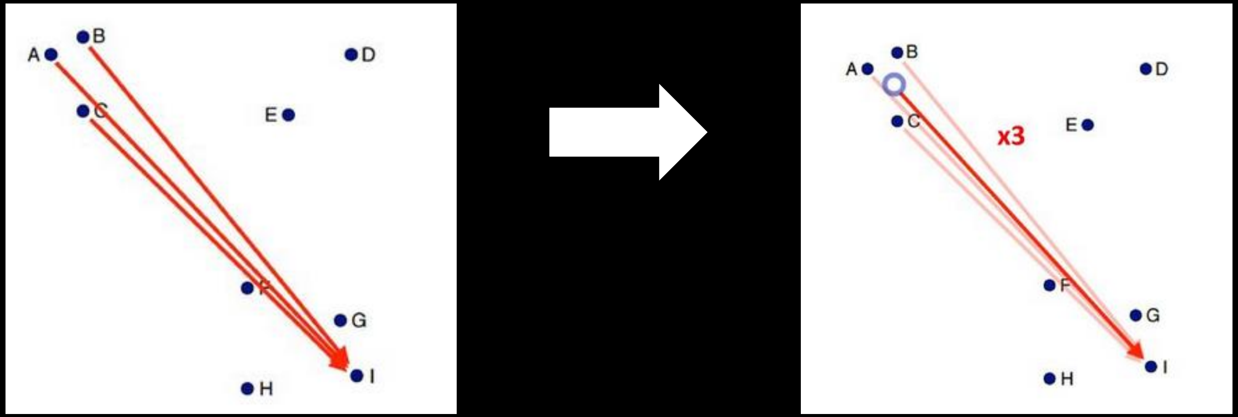
Van der Maaten & Hinton, 2008, Journal of Machine Learning Research

➔ Scalability optimization of the descending gradient with the Barnes-Hut Approximation

Van der Maaten, 2014,
Journal of Machine Learning Research

Many of the paiwise interactions between points are very similar

Approximate similar interactions by single interaction



https://slideplayer.com/slide/12695684/

t-SNE : How is different from SNE ?

(2) it uses a Student-t distribution rather than a Gaussian to compute the similarity between two points *in the low-dimensional space*.

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

“ A computationally convenient property is that it is much faster to evaluate the density of a point under a Student t-distribution than under a Gaussian because it does not involve an exponential. ”

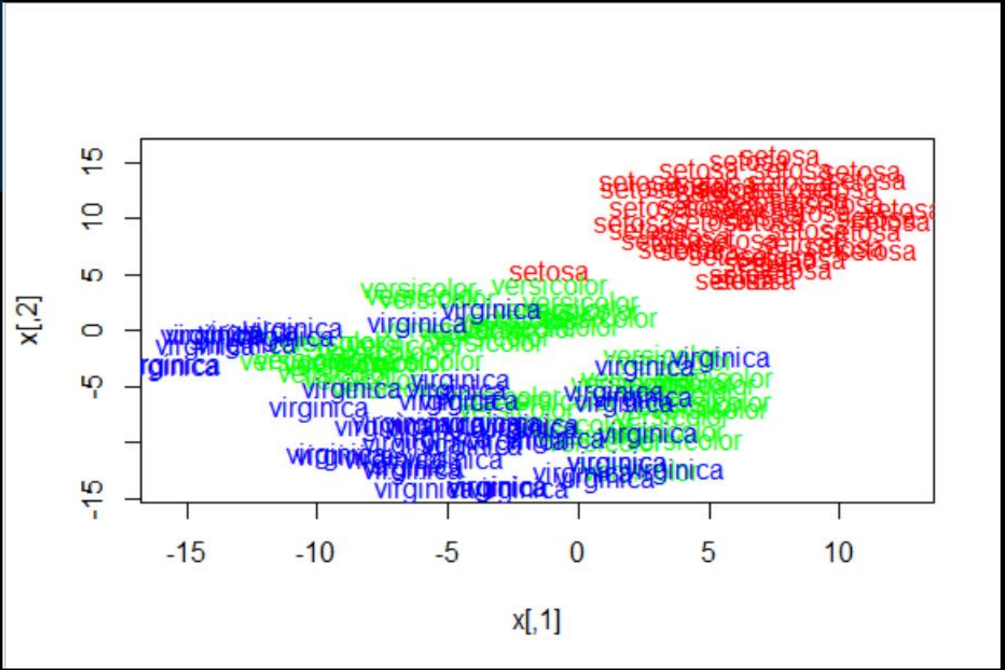
Van der Maaten & Hinton, 2008, Journal of Machine Learning Research

R implementation.

- **tsne** (published in 2016-07-15, Version: 0.1-3)
  A "pure R" implementation of the t-SNE algorithm.

```r
# install.packages("tsne")
library("tsne")

colors = rainbow(length(unique(iris$Species)))
names(colors) = unique(iris$Species)
ecb = function(x, y) {
  plot(x, t = 'n')
  text(x, labels = iris$Species, col = colors[iris$Species])
}
tsne_iris = tsne(iris[,1:4], epoch_callback = ecb, perplexity = 30)
```



https://scwn.wordpress.com/2010/02/20/new-t-sne-package-for-r/
https://cran.r-project.org/web/packages/tsne/index.html

R implementation.

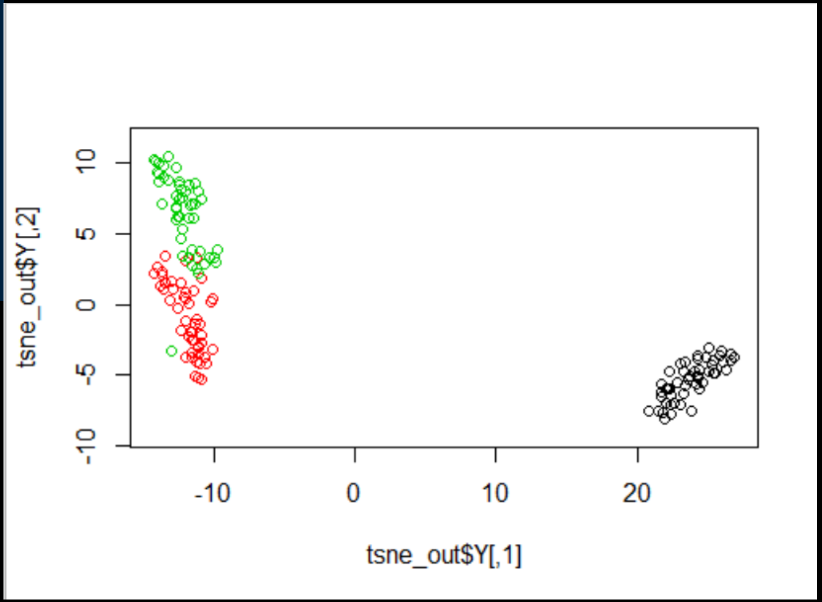
- **Rtsne** (published in 2018-11-10, Version: 0.15)
  Wrapper for the C++ implementation of Barnes-Hut t-Distributed Stochastic Neighbor Embedding. t-SNE is a method for constructing a low dimensional embedding of high-dimensional data, distances or similarities. Exact t-SNE can be computed by setting theta = 0.0.

```r
# install.packages("Rtsne")
library(Rtsne)

iris_unique <- unique(iris) # Remove duplicates
iris_matrix <- as.matrix(iris_unique[,1:4])

# Run t-SNE
tsne_out <- Rtsne(X = iris_matrix, pca = FALSE, perplexity = 30, theta = 0.0)

# Show the objects in the 2D tsne representation
plot(tsne_out$Y, col = iris_unique$Species, asp = 1)
```



https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/

R implementation.

- Rtsne

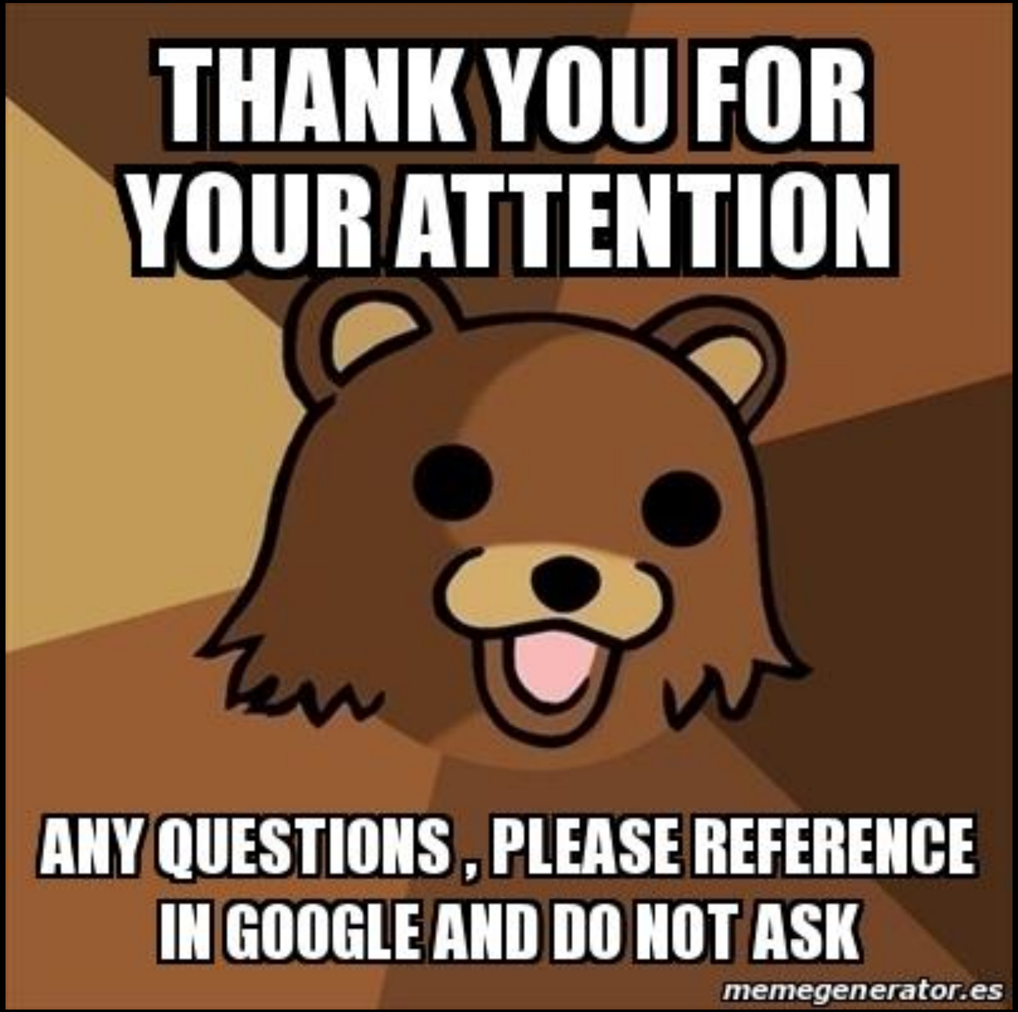
Dims	They are the number of dimensions the data must be reduced to.
Perplexity	It can be interpreted as a smooth measure of the effective number of neighbors. The performance of SNE is fairly robust to changes in the perplexity, and typical values are between 5 and 50.
Max_iter	Maximum iterations
Theta	numeric; speed / accuracy trade-off (increase for less accuracy), set to 0.0 for exact TSNE (default: 0.5)
Check_duplicates	logical; Checks whether duplicates are present. It is best to make sure there are no duplicates present and set this option to FALSE, especially for large datasets (default: TRUE)
Pca	logical; Whether an initial PCA step should be performed (default: TRUE)
Max_iter	integer; Number of iterations (default: 1000)
Verbose	logical; Whether progress updates should be printed (default: FALSE)
Is_distance	logical; Indicate whether X is a distance matrix (experimental, default: FALSE)
Y_init	matrix; Initial locations of the objects. If NULL, random initialization will be used (default: NULL). Note that when using this, the initial stage with exaggerated perplexity values and a larger momentum term will be skipped.

https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/

Sources.

- https://lvdmaaten.github.io/tsne/
- L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research* 9(Nov):2579-2605, 2008.
- L.J.P. van der Maaten. Learning a Parametric Embedding by Preserving Local Structure. In *Proceedings of the Twelfth International Conference on Artificial Intelligence & Statistics (AI-STATS), JMLR W&CP* 5:384-391, 2009.
- L.J.P. van der Maaten and G.E. Hinton. Visualizing Non-Metric Similarities in Multiple Maps. *Machine Learning* 87(1):33-55, 2012.
- L.J.P. van der Maaten. Accelerating t-SNE using Tree-Based Algorithms. *Journal of Machine Learning Research* 15(Oct):3221-3245, 2014.

Images :
https://blog.paperspace.com/dimension-reduction-with-t-sne/
https://distill.pub/2016/misread-tsne/
https://www.youtube.com/watch?v=hMUrZ708PFk
https://www.analyticsvidhya.com/blog/2017/01/t-sne-implementation-r-python/
https://slideplayer.com/slide/12695684/


THANK YOU FOR
YOUR ATTENTION
ANY QUESTIONS , PLEASE REFERENCE
IN GOOGLE AND DO NOT ASK
memegenerator.es