STOCHASTIC NETWORK EMBEDDING (T-SNE)

Julie A Dickerson
For BCB 570
Spring 2020

DIMENSION REDUCTION

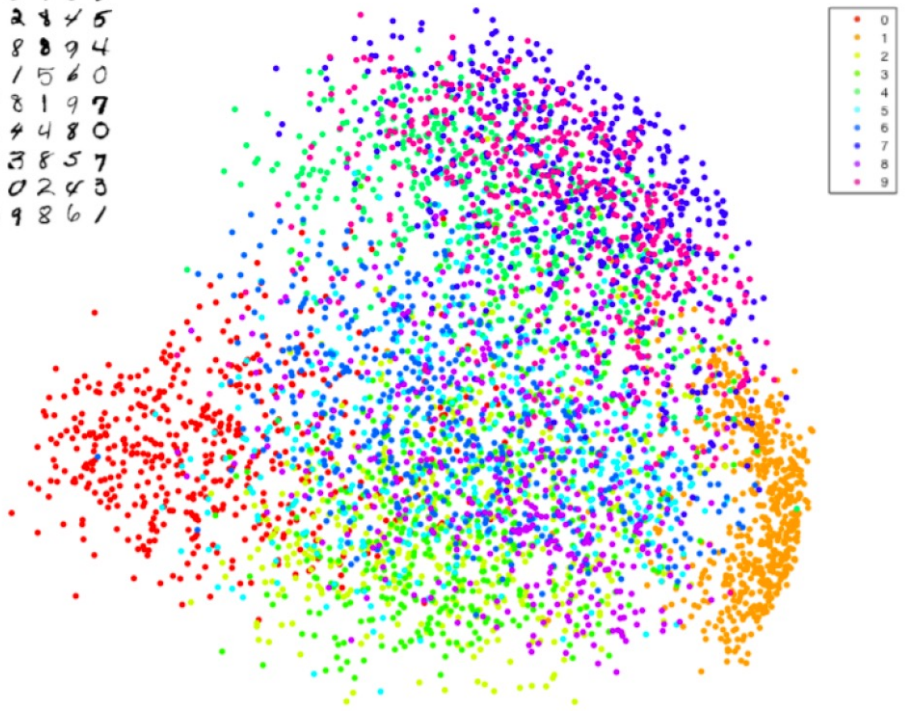
- PCA tries to find a global structure
  - Low dimensional subspace
  - Can lead to local inconsistencies
  - Far away points can become nearest neighbors
  - Linearly seperable
- t-SNE is an alternative dimensionality reduction algorithm.
- t-SNE tries to preserve local structure
  - Low dimensional neighborhood should be the same as original neighborhood.
- Unlike PCA almost only used for visualization
  - No easy way to embed new points

WHAT IS A GOOD VISUALIZATION?

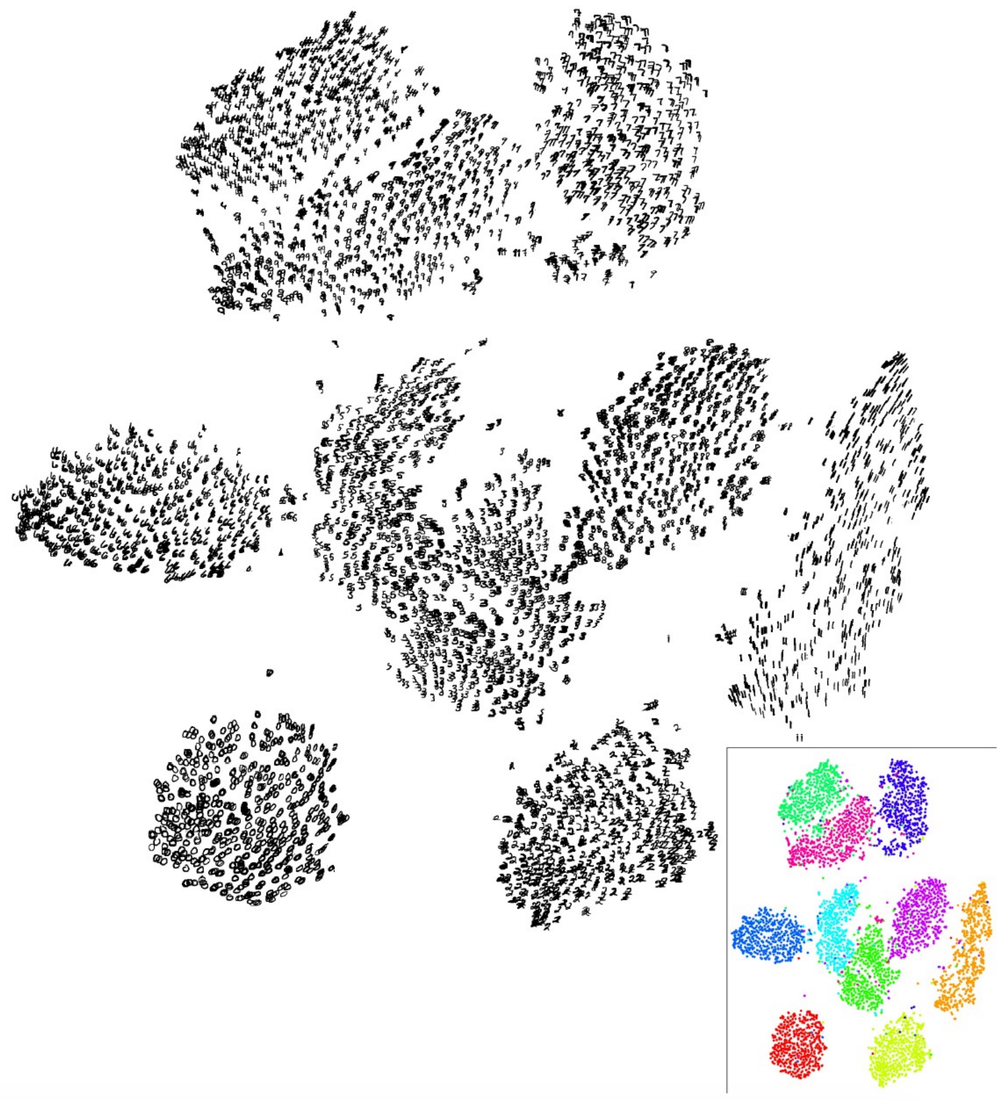
Technical requirements:

- Visualizability: (usually 2D or 3D).
- Fidelity: Relationship among points are preserved (i.e. similar points remain distinct; distinct points remain distinct).
- Scalability: Can deal with large, high-dimensional data sets.

PCA WITH MNIST DATA SET (FOR ZIP CODES)



MNIST WITH T-SNE



STOCHASTIC NEIGHBOR EMBEDDING (SNE)

Basic idea:

- "Encode" high dimensional neighborhood information as a distribution
- Intuition: Random walk between data points.
  - High probability to jump to a close point
- Find low dimensional points such that their neighborhood distribution is similar.
- How do you measure distance between distributions?
  - Most common measure: KL divergence

- G.E. Hinton and S.T. Roweis. Stochastic Neighbor Embedding. NIPS2002

NEIGHBORHOOD DISTRIBUTIONS

- Consider the neighborhood around an input data point, $x_i \in \mathbb{R}^d$
- Assume a Gaussian distribution centered around x_i
- The probability that x_i chooses some other datapoint x_j as its neighbor is in proportion with the density under the Gaussian
- A point closer to x_i in Euclidean distance will be more likely than one further away

PROBABILITIES P_{IJ}

- The i -> j probability, is the probability that point x_i chooses x_j as its neighbor

$$P_{j|i} = \frac{\exp\left(-||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-||\mathbf{x}^{(i)} - \mathbf{x}^{(k)}||^2/2\sigma_i^2\right)}$$

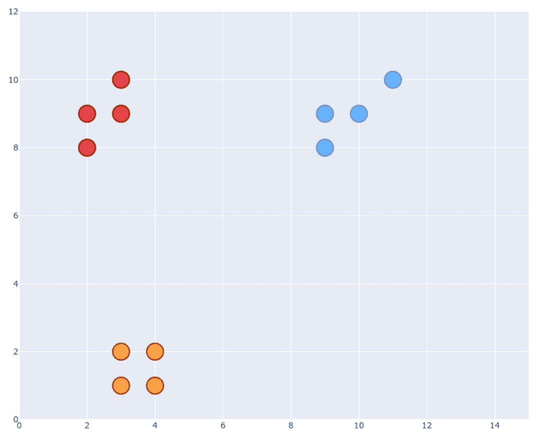
- The parameter σ_i sets the size of the neighborhood
  - Very low σ_i - all the probability is in the nearest neighbor.
  - Very high σ_i - Uniform weights.
- σ_i is set differently for each data point
- Results depend heavily on σ_i - it defines the local neighborhoods.
- Final distribution over pairs is symmetrized: $P_{ij} = (P_{i|j} + P_{j|i})/2N$
  - Pick i uniformly and then "jump" to j according to $P_{j|i}$

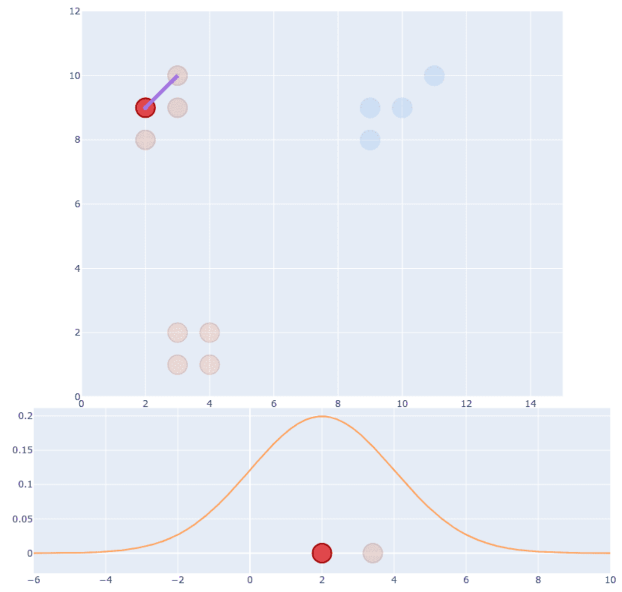
https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

SNE ILLUSTRATED

$$P_{j|i} = \frac{\exp\left(-||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-||\mathbf{x}^{(i)} - \mathbf{x}^{(k)}||^2 / 2\sigma_i^2\right)}$$

“ similarity of datapoint x_j **to datapoint** x_i **is the conditional probability** p_{j|i}, **that** x_i **would pick** x_j **as its neighbor “.**

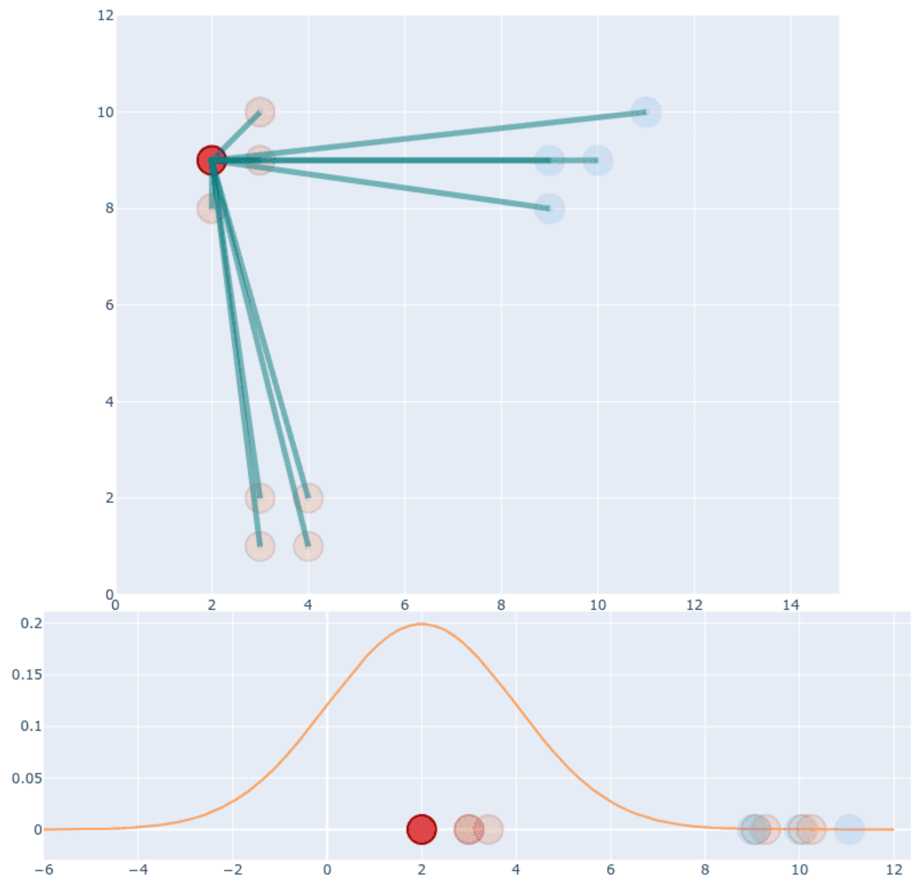




https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

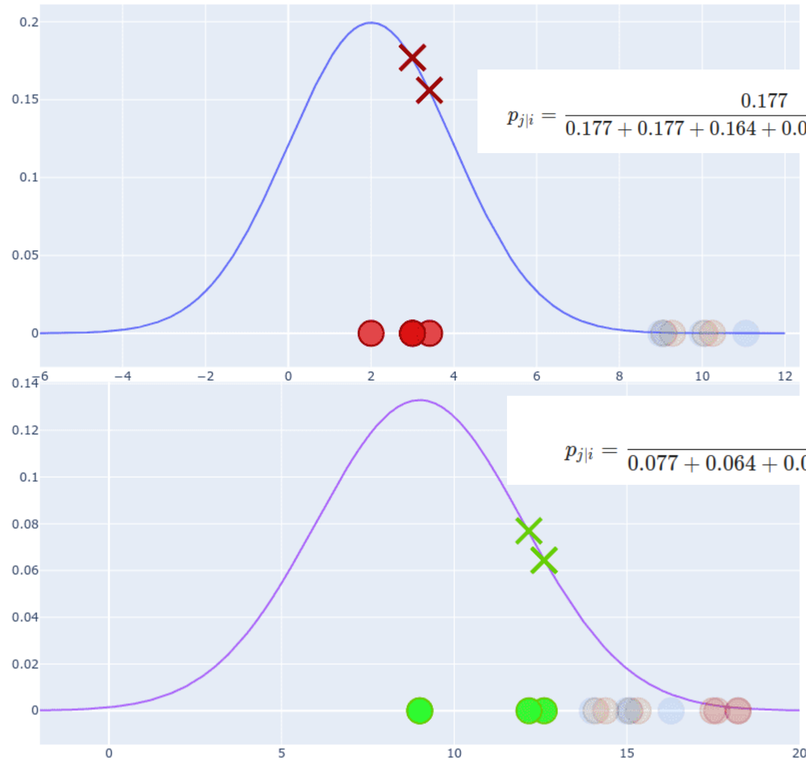
SNE ILLUSTRATED

$$P_{j|i} = \frac{\exp\left(-||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-||\mathbf{x}^{(i)} - \mathbf{x}^{(k)}||^2/2\sigma_i^2\right)}$$



https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

SNE ILLUSTRATED



$$p_{j|i} = \frac{0.177}{0.177 + 0.177 + 0.164 + 0.0014 + 0.0013 + \ldots} \approx 0.34$$

$$p_{j|i} = \frac{0.077}{0.077 + 0.064 + 0.064 + 0.032 + 0.031 + \ldots} \approx 0.27$$

Normalize by dividing the current projection value by the sum of the projections.

$$p_{j|i} = \frac{g(|x_i - x_j|)}{\sum_{k \neq i} g(|x_i - x_k|)}$$

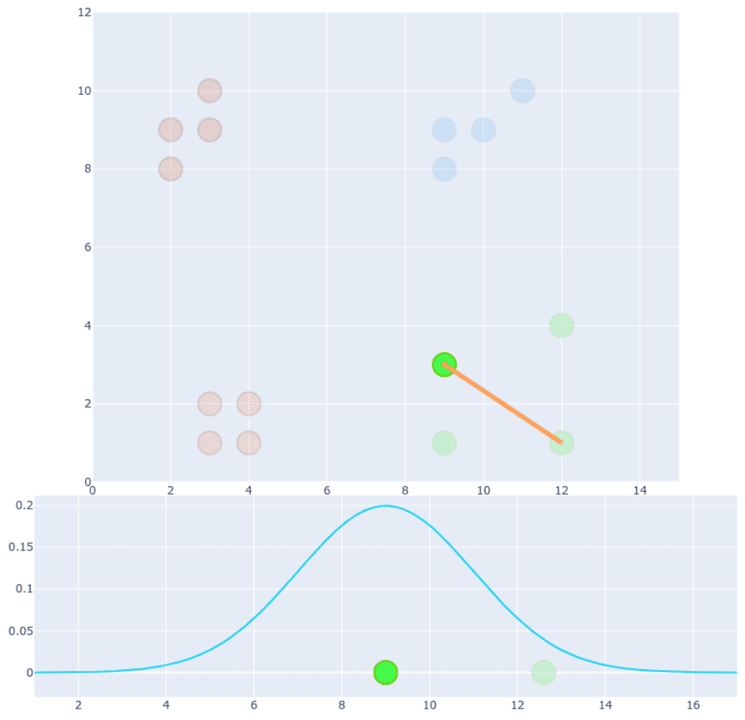
This scales all values to have a sum equal to 1.

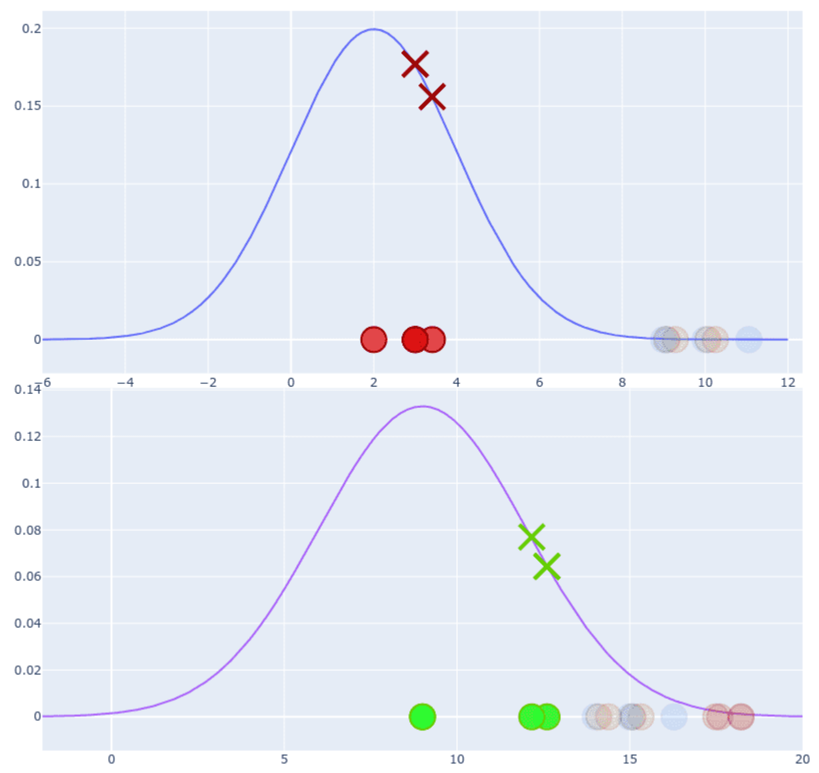
$$\sum_j p_{j|i} = 1$$

Also, p_{i|i} is set to be equal to 0, not 1.

https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

SNE ILLUSTRATED



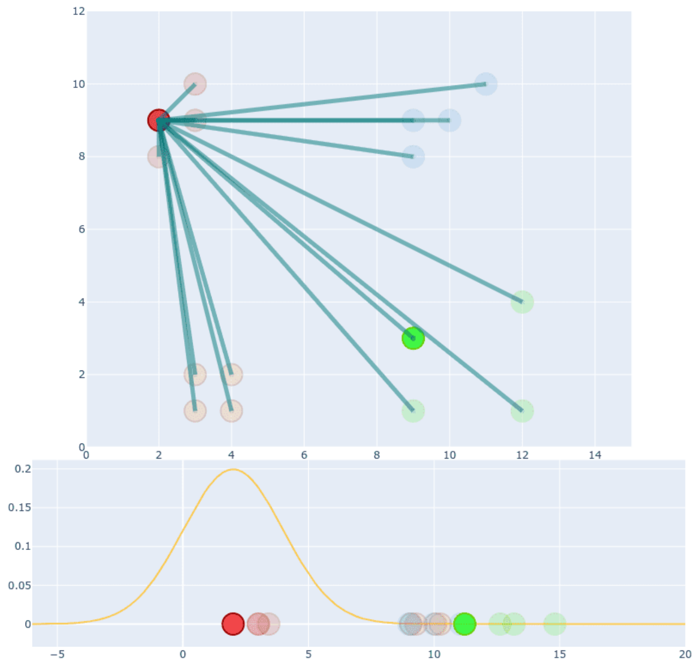


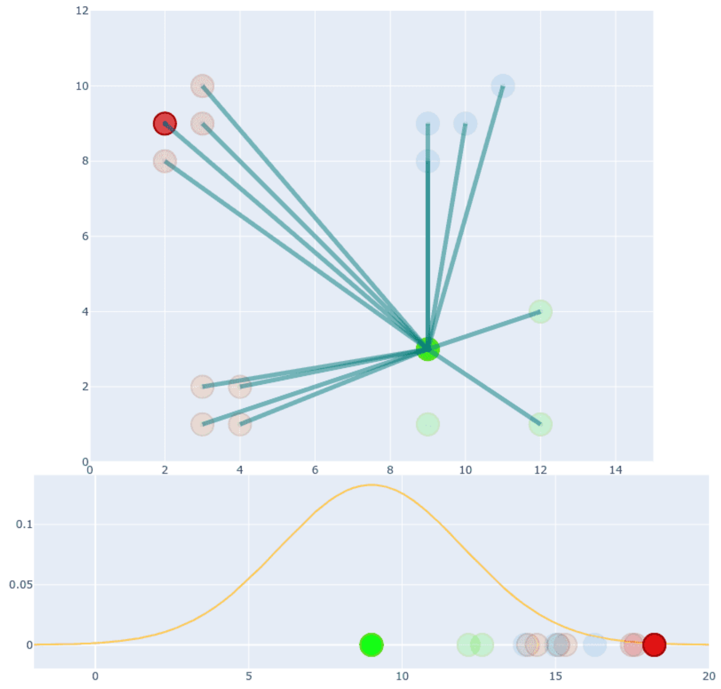
https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

SNE ILLUSTRATED SYMMETRY

values of p_{i|j} and p_{j|i} are different

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

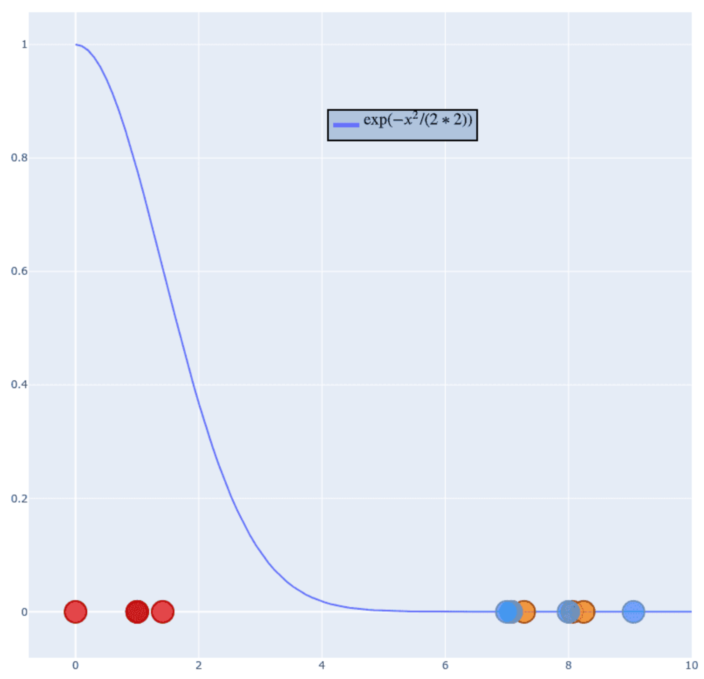


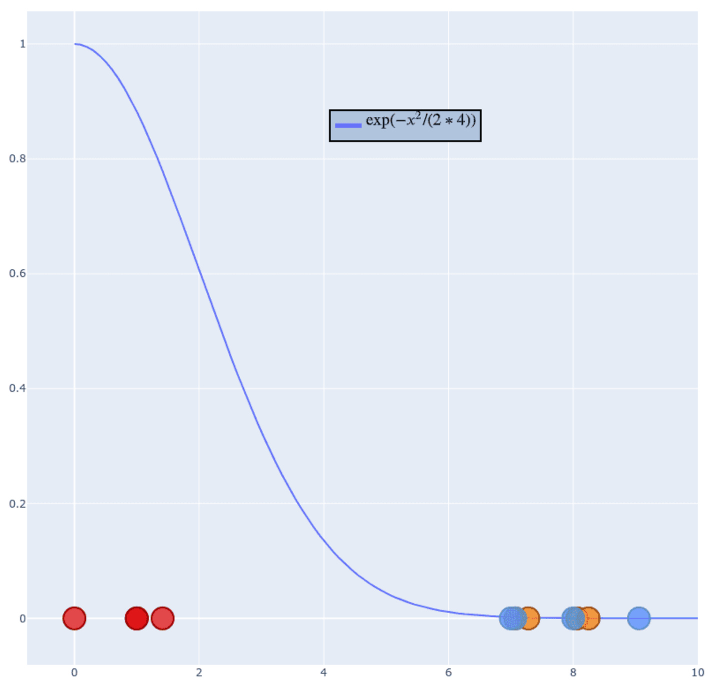


https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

SNE ILLUSTRATED SYMMETRY

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





PERPLEXITY

Perplexity is a measure for information that is defined as 2 to the power of the Shannon entropy.

The perplexity of a fair die with k sides is equal to k.

In t-SNE, the perplexity may be viewed as a knob that sets the number of effective nearest neighbors.

It is comparable with the number of nearest neighbors k that is employed in many manifold learners.

Important parameter - different perplexity values capture different scales in the data

PERPLEXITY

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

σ^2 is a key parameter

Perplexity is more or less a target number of neighbors for our central point. Basically, the higher the perplexity is, the higher value variance has.

SNE performs a binary search for sigma that produces a probability distribution with a fixed perplexity specified by the user

Where

$$Perp(P_i) = 2^{-\sum p_{j|i} log_2 p_{j|i}}$$

$$-\sum p_{j|i} log_2 p_{j|i}$$

PERPLEXITY AND $P_{J||}$(KEY HYPERPARAMETER)

For each distribution $P_{j|i}$ (depends on σ_i) we define the perplexity

- $perp(P_{j|i}) = 2^{H(P_{j|i})}$ where $H(P) = - \Sigma_i P_i \log(P_i)$ is the entropy.

If P is uniform over k elements - perplexity is k.

- Smooth version of k in kNN
- Low perplexity = small σ^2
- High perplexity = large σ^2

Define the desired perplexity and set σ_i to get that (bisection method)

Values between 5-50 usually work well

SNE OBJECTIVE

- Given a set of D dimensional data points $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)} \in \mathbb{R}^D$, define the distribution P_{ij}
- **Goal:** Find a "good" embedding, $\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(N)} \in \mathbb{R}^d$ for $d < D$.
- For $\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(N)} \in \mathbb{R}^d$, we define a distribution Q using a local Gaussian approximator

$$Q_{ij} = \frac{\exp\left(-||\mathbf{y}^{(i)} - \mathbf{y}^{(j)}||^2\right)}{\sum_k \sum_{l \neq k} \exp\left(-||\mathbf{y}^{(l)} - \mathbf{y}^{(k)}||^2\right)}$$

- What is "good"?
- Optimize Q to be close to P

The embeddings $\mathbf{y}^{(1)},\ldots,\mathbf{y}^{(N)} \in \mathbb{R}^d$ are the parameters we are optimizing.

HOW TO MEASURE SIMILARITY IN DISTRIBUTION?

- Kullback-Leibler Divergence (also called **relative entropy**) is a measure of how one probability distribution is different from a second, reference probability distribution

  Measures distance between two distributions, P and Q:

$$KL(Q\|P) = \sum_{ij} Q_{ij} \log\left(\frac{Q_{ij}}{P_{ij}}\right)$$

  Not a metric function as it is not symmetric, KL(Q||P) is not equal KL(P||Q) and the triangle inequality does not hold

CODING THEORY INTUITION:

If we are transmitting information that is distributed according to P, then the optimal (lossless) compression will need to send on average H(P) bits.

What happens you expect P (and design your compression accordingly) but the actual distribution is Q?

- Send on average H(Q) + KL(Q || P)
- KL(Q||P) is the "penalty" for using wrong distribution

https://timvieira.github.io/blog/post/2014/10/06/kl-divergence-as-an-objective-function/

SNE COST FUNCTION

Uses sum of all KL divergences over all data points.

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

P_i is conditional probability over all other datapoints given point x_i
Q_i is conditional probability over all other datapoints given point y_i

Minimize using gradient descent:

$$\frac{\partial C}{\partial y_i} = 2\sum_j \left(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j}\right)\left(y_i - y_j\right)$$

SNE ALGORITHM SUMMARY

- Compute an N x N similarity matrix in the high-dimensional input space.
- Define an N x N similarity matrix in the low-dimensional embedding space.
- Define cost function - sum of KL divergence between the two probability distributions at each point.
- Iteratively learn low-dimensional embedding by minimizing the cost function using gradient descent.

PROBLEMS WITH SNE

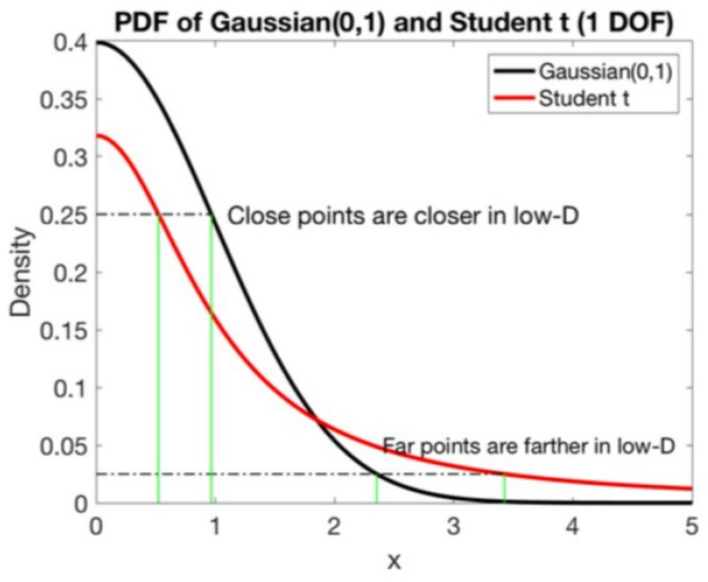
Cost function is difficult to optimize

“Crowding Problem”: The area of the 2D map that is available to accommodate moderately distant data points will not be large enough compared with the area available to accommodate nearby data points.

STUDENT-T KERNEL

The heavy tails of the normalized Student-t kernel allow dissimilar input objects x_i and x_j to be modeled by low-dimensional counterparts y_i and y_j that are too far apart because q_{ij} is not very small for two embedded points that are far apart.

Note: Since q is what to be learned, the outlier problem does not exist for the low-dimension.



http://www.cs.columbia.edu/~verma/classes/uml/lec/uml_lec8_tsne_slides.pdf

https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a

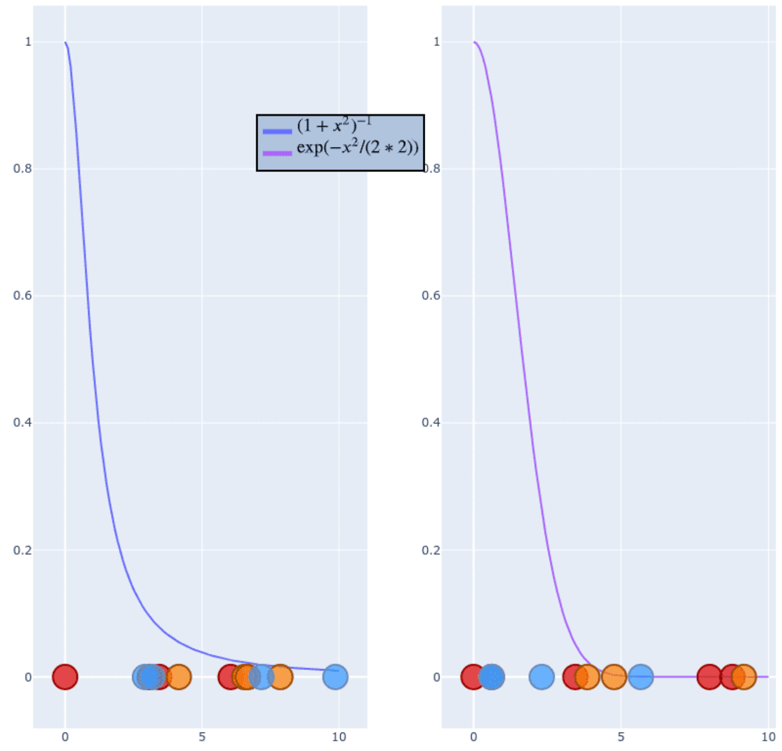
T-SNE LOW DIMENSIONAL MAPPING

Map to low-dimensional space with the same number of points as in the original space

Goal: find similar probability distribution in low-dimensional space. Original SNE used Gaussian which has problems with crowding of points, Student t-distribution with a single degree of freedom

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

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



T-SNE (T-DISTRIBUTED STOCHASTIC NEIGHBOR EMBEDDING)

L.J.P. van der Maaten and G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. JMLR2008

A symmetrized version of the SNE cost function with simpler gradients. Uses joint distribution instead of conditional (becomes symmetric):

$$C = \sum_i KL\left(P_i \| Q_i\right) = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right) \qquad p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

A Student-t distribution rather than a Gaussian to compute the similarity in the low-dimensional space to alleviate the crowding problem and the optimization problems of SNE.

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

LEARNING THE MAPPING

Use gradient descent to find this mapping using KL-divergence

$$C = D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right)$$

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

http://www.cs.columbia.edu/~verma/classes/uml/lec/uml_lec8_tsne_slides.pdf

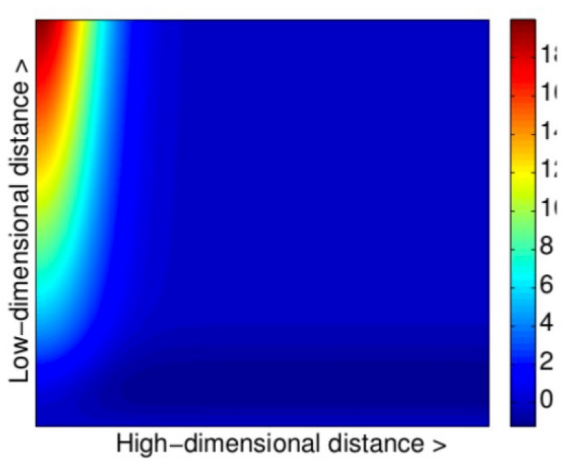
GRADIENT IS SIMPLER

The gradient of the cost function is:

$$\begin{aligned}\frac{dC}{dy_i} &= 4\sum_{j=1,j\neq i}^{n}(p_{ij}-q_{ij})(1+||y_i-y_j||^2)^{-1}(y_i-y_j)\\ &= 4\sum_{j=1,j\neq i}^{n}(p_{ij}-q_{ij})q_{ij}Z(y_i-y_j)\\ &= 4\Big(\sum_{j\neq i}p_{ij}q_{ij}Z(y_i-y_j)-\sum_{j\neq i}q_{ij}^2Z(y_i-y_j)\Big)\\ &= 4(F_{attraction}+F_{repulsion})\end{aligned}$$

where $Z=\sum_{l,s=1,l\neq s}^{n}(1+||y_l-y_s||^2)^{-1}$. The derivation can be found in the appendix of the t-SNE paper.

PROFILE REFLECTS MORE WHAT WE WANT



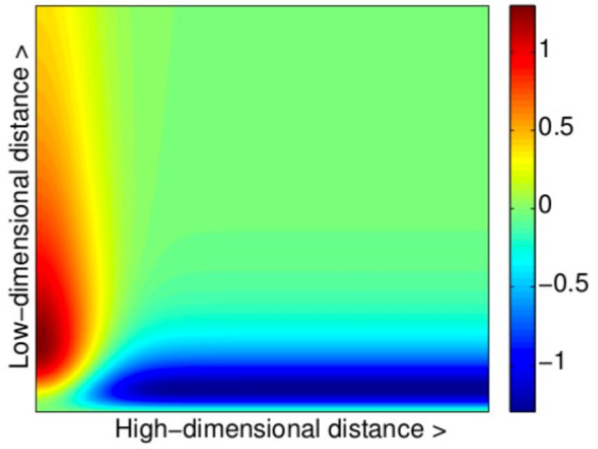


Figure : Gradient of SNE and t-SNE

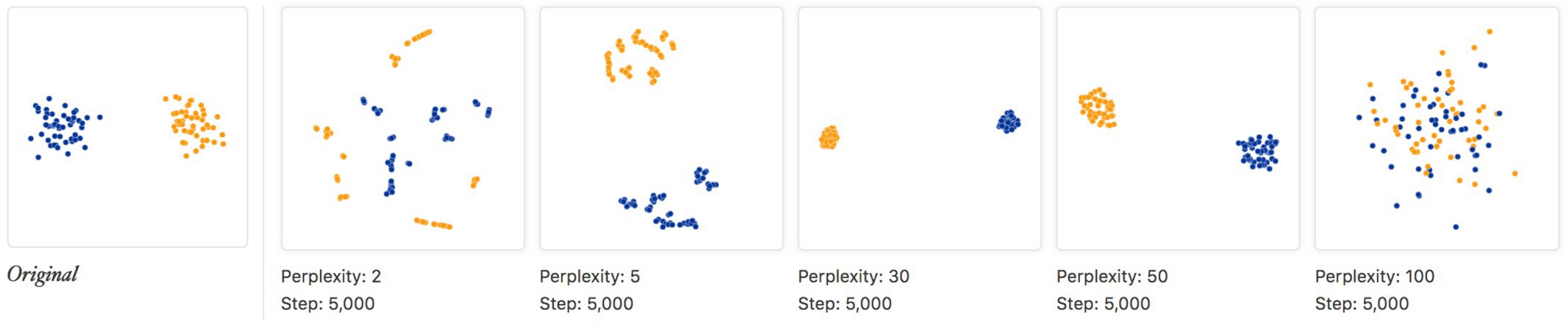
T-SNE ALGORITHM

Algorithm 1: Simple version of t-Distributed Stochastic Neighbor Embedding.

```
Data: data set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations $T$, learning rate $\eta$, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, \ldots, y_n\}$.
begin
    compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using Equation 1)
    set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
    sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, \ldots, y_n\}$ from $\mathcal{N}(0, 10^{-4} I)$
    for $t$=1 to $T$ do
        compute low-dimensional affinities $q_{ij}$ (using Equation 4)
        compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)
        set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t) \left( \mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)} \right)$
    end
end
```

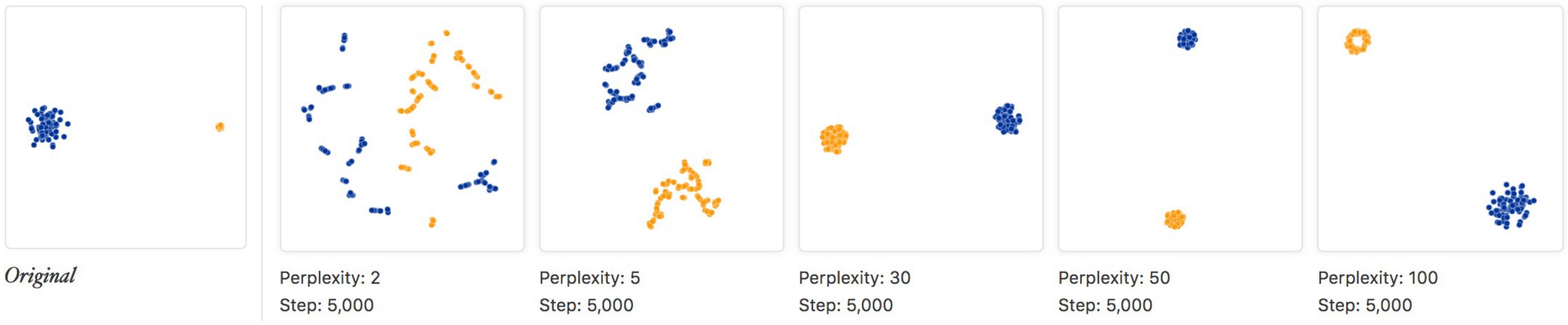
HOW TO USE T-SNE EFFECTIVELY

- Awesome resource https://distill.pub/2016/misread-tsne/
- Perplexity needs to be chosen carefully. also, keep learning until output is stable!



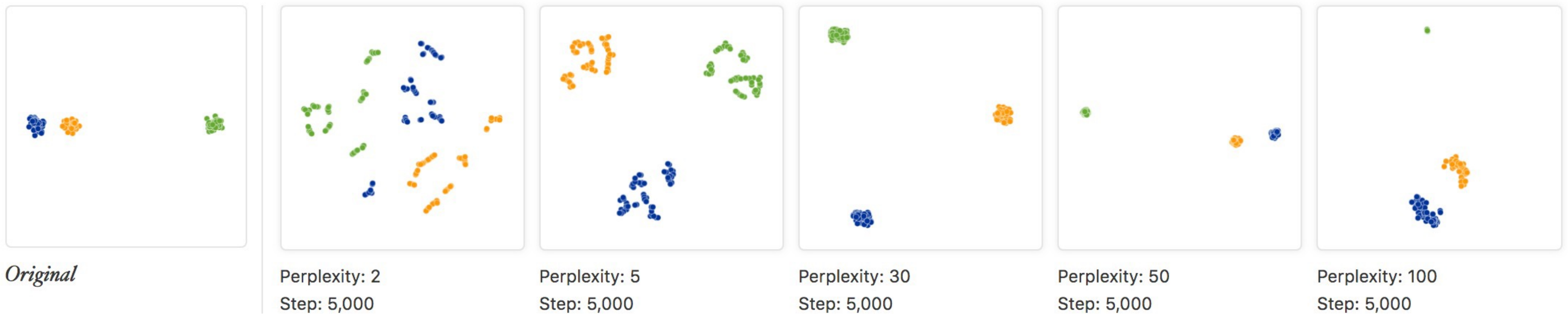
RELATIVE CLUSTER SIZE

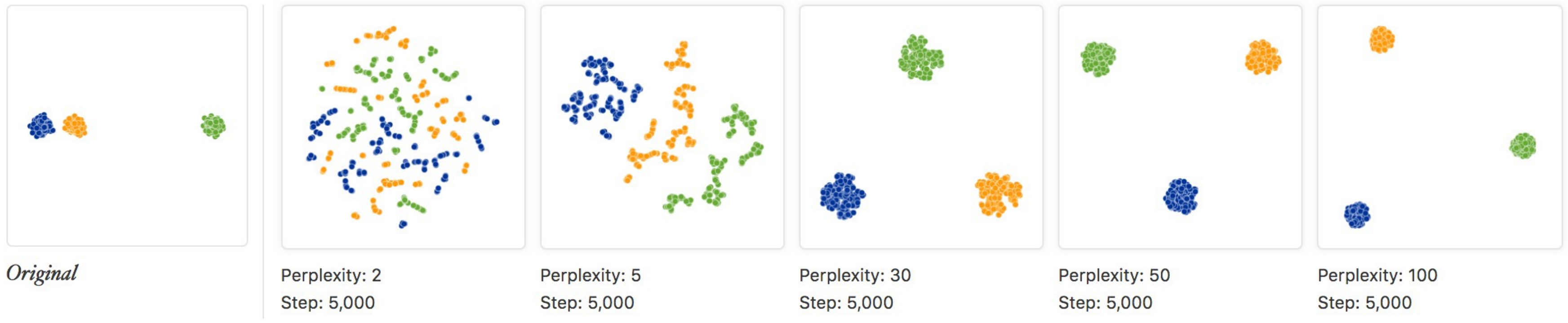
Usually not meaningful in visualization.



GLOBAL STRUCTURES

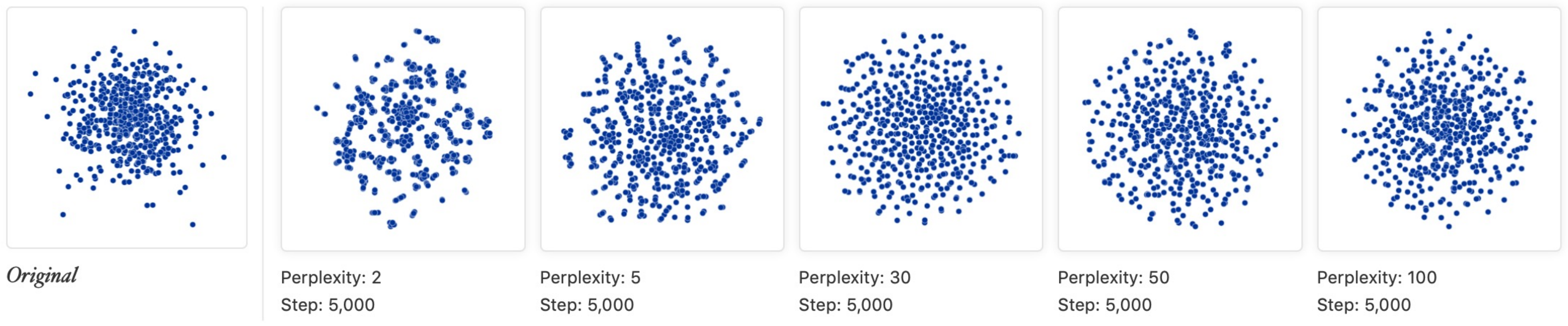
Distances between clusters not always preserved





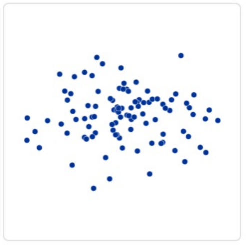
RANDOM NOISE DOESN'T ALWAYS LOOK RANDOM

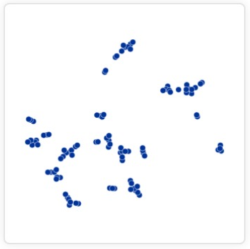
drawn from a unit Gaussian distribution in 100 dimensions

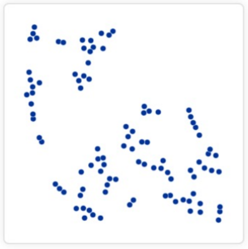


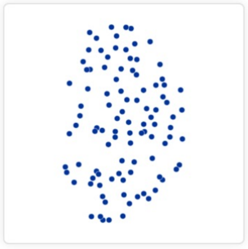
SHAPES COME AND GO

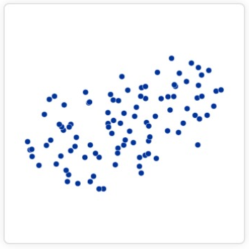
Top should look like a longish ellipse

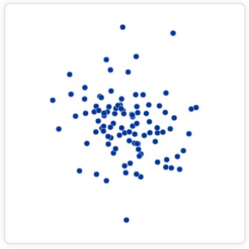

Original

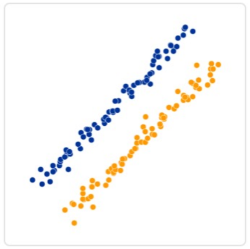

Perplexity: 2
Step: 5,000

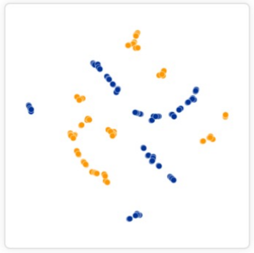

Perplexity: 5
Step: 5,000

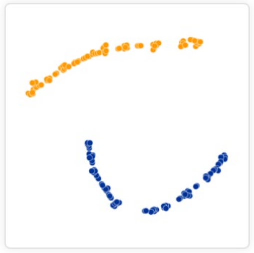

Perplexity: 30
Step: 5,000

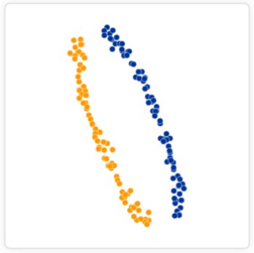

Perplexity: 50
Step: 5,000

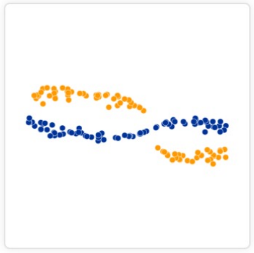

Perplexity: 100
Step: 5,000

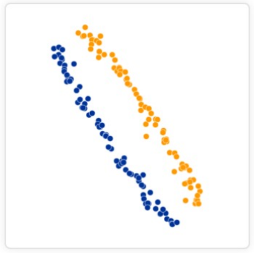

Original


Perplexity: 2
Step: 5,000


Perplexity: 5
Step: 5,000


Perplexity: 30
Step: 5,000


Perplexity: 50
Step: 5,000


Perplexity: 100
Step: 5,000

LIMITATIONS

- Dimensionality reduction for other problems(due to the heavy tail of the t-distribution, it does not preserve the local structure as well if the embedded dimension is larger, say 100).
- Curse of dimensionality(t-SNE employs Euclidean distances between near neighbors so it implicitly depends on the local linearity on the manifold).
- $O(N^2)$ computational complexity(the evaluation of the joint distributions involve $N(N - 1)$ pairs of objects.).
- Perplexity number, number of iterations, the magnitude of early exaggeration parameter have to be manually chosen.

http://www.cs.columbia.edu/~verma/classes/uml/lec/uml_lec8_tsne_slides.pdf

T-SNE CAVEATS (VAN DER MAATEN)

How can I asses the quality of the visualizations that t-SNE constructed?

Preferably, just look at them! Notice that t-SNE does not retain distances but probabilities, so measuring some error between the Euclidean distances in high-D and low-D is useless. However, if you use the same data and perplexity, you can compare the Kullback-Leibler divergences that t-SNE reports. It is perfectly fine to run t-SNE ten times, and select the solution with the lowest KL divergence.

When I run t-SNE, I get a strange 'ball' with uniformly distributed points?

This usually indicates you set your perplexity way too high. All points now want to be equidistant. The result you got is the closest you can get to equidistant points as is possible in two dimensions.

T-SNE: ACCENT STRONG CLUSTERS FIRST

- In the initial stage, multiply p_{ij} by a cofficient $\alpha > 1$. This is called "Early Exaggeration".
- This focuses on modeling the large p_{ij} by fairly large q_{ij}.
- The natural result is to form tight widely separated clusters in the map and thus makes it easier for the clusters to move around relative to each other in order to find a global organization.

COMPUTATIONAL ACCELERATION

L.J.P. van der Maaten. Accelerating t-SNE using Tree-Based Algorithms. JMLR2014

- Observations: Many of the pairwise interactions between points are very similar.
- Idea: Approximate similar interactions by a single interaction using a metric tree that has O(uN) non-zero values.
- Result: Reduce complexity to O(N log N) via Barnes-Hut-SNE (tree-based) algorithm. The method can deal with up to tens of millions data points.

http://www.cs.columbia.edu/~verma/classes/uml/lec/uml_lec8_tsne_slides.pdf

FAST T-SNE FIT-SNE

Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data. Linderman, G.C., Rachh, M., Hoskins, J.G. *et al*. Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data. *Nat Methods* **16**, 243–245 (2019). https://doi.org/10.1038/s41592-018-0308-4

Uses fast Fourier transform (FFT)-accelerated interpolation-based t-SNE (FIt-SNE), an algorithm for rapid computation of one-dimensional (1D) and 2D t-SNE based on polynomial interpolation and further accelerated using the FFT.

EXAMPLES

Arts Classification https://artsexperiments.withgoogle.com/tsnemap

- Large database of pictures, grouped by similiarity in three-dimensions
- Requires fast internet connection

REFERENCES

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. **Accelerating t-SNE using Tree-Based Algorithms.** *Journal of Machine Learning Research* 15(Oct):3221-3245, 2014.

Google Tech Talk

https://www.youtube.com/watch?v=RJVL80Gg3lA&list=UUtXKDgv1AVoG88PLl8nGXmw

How to Use t-SNE Effectively https://distill.pub/2016/misread-tsne/

http://www.cs.columbia.edu/~verma/classes/uml/lec/uml_lec8_tsne_slides.pdf