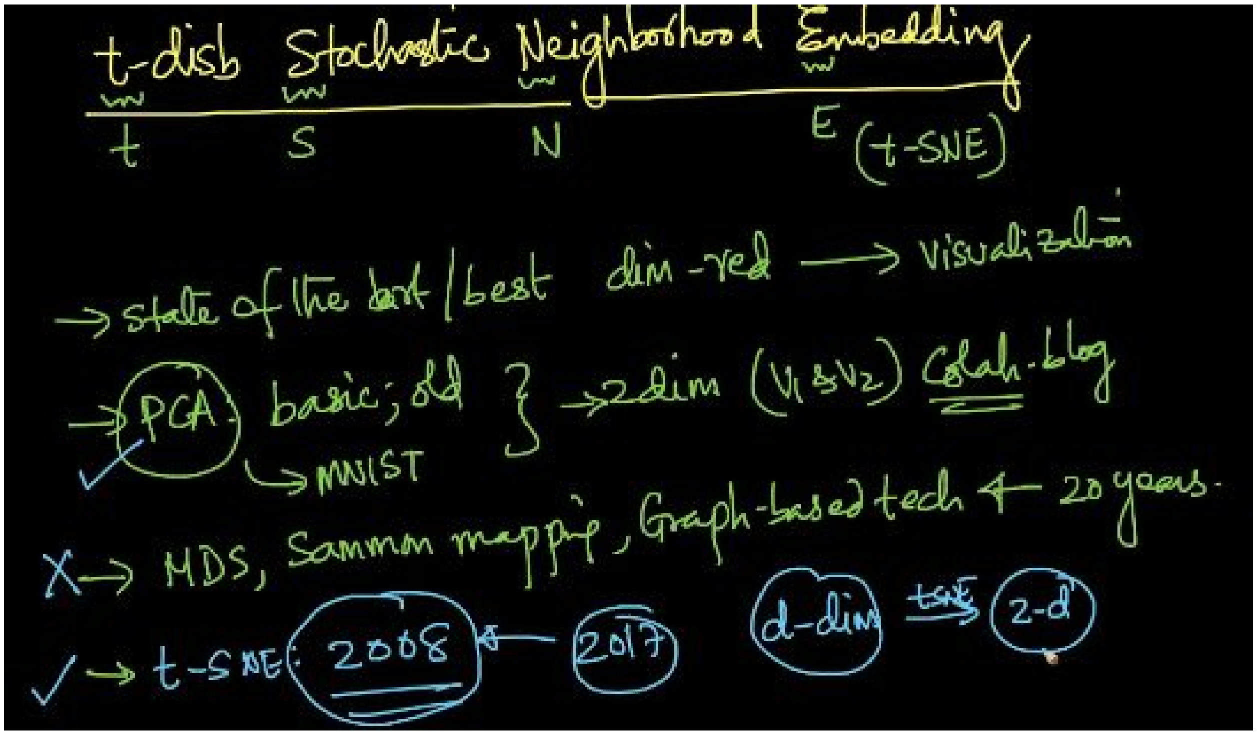
T-distributed stochastic neighbor embedding



t-disb Stochastic Neighborhood Embedding

t S N E (t-SNE)

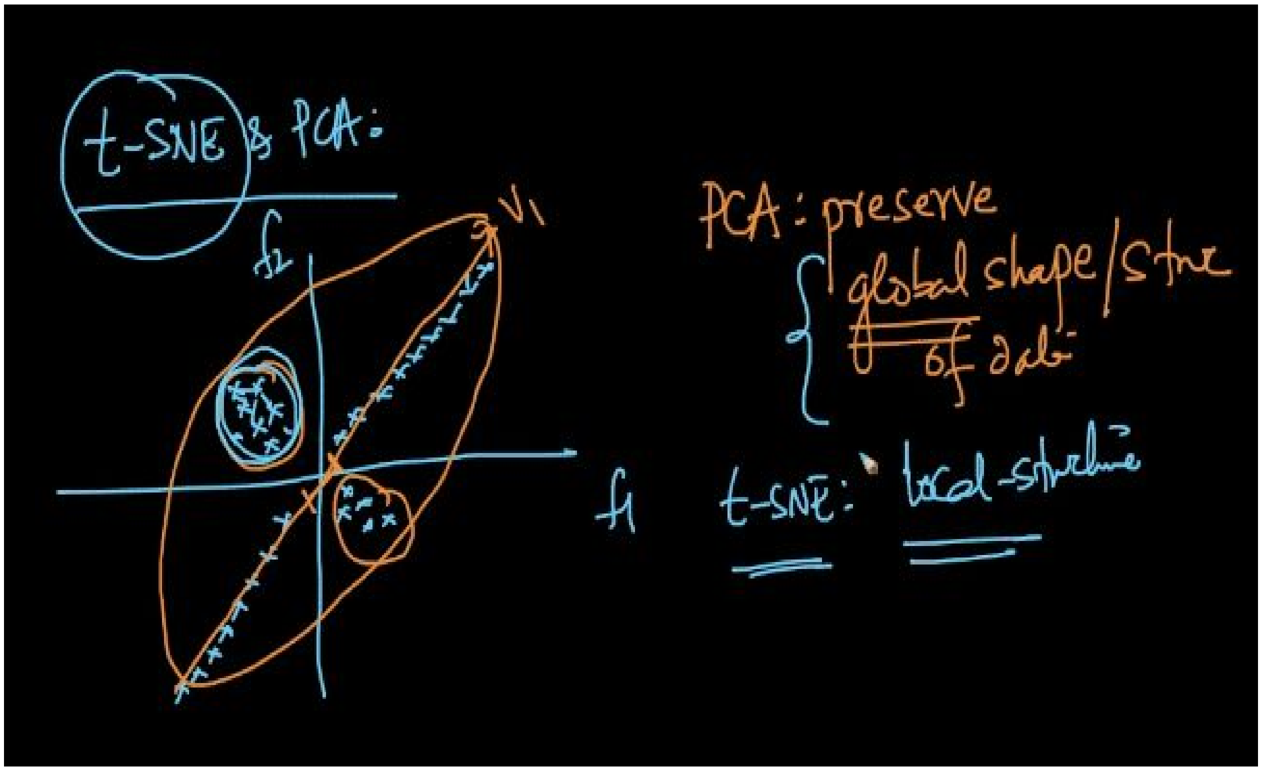
→ state of the art / best dim-red → visualization

→ PCA basic; old } → 2dim (V_1 & V_2) Colah.blog

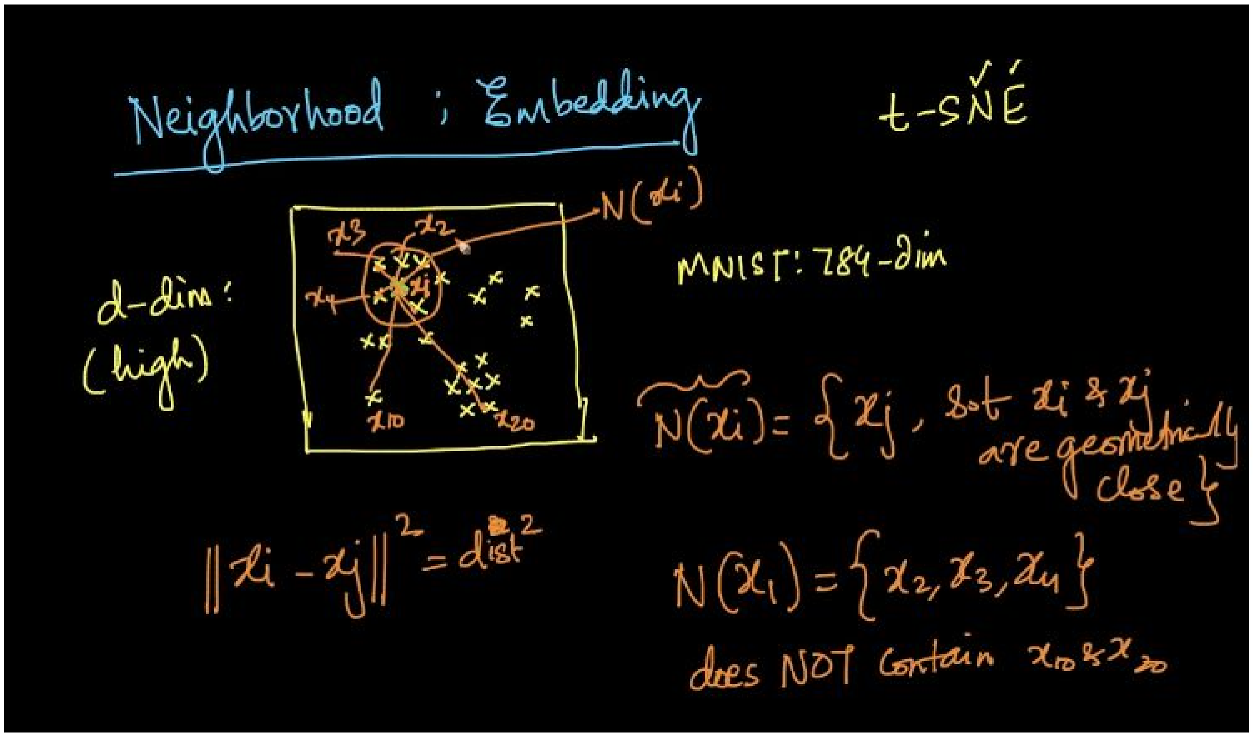
↳ MNIST

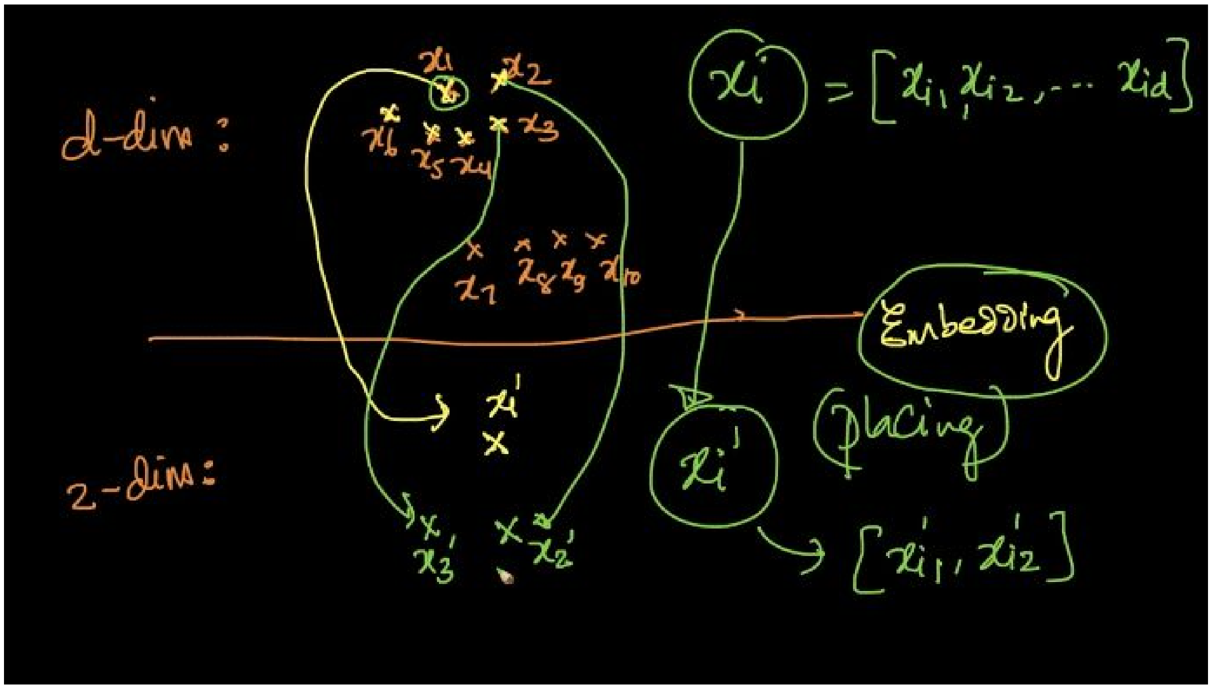
X → MDS, Sammon mapping, Graph-based tech ← 20 years.

✓ → t-SNE: 2008 ← 2017   d-dim $\xrightarrow{tSNE}$ 2-d

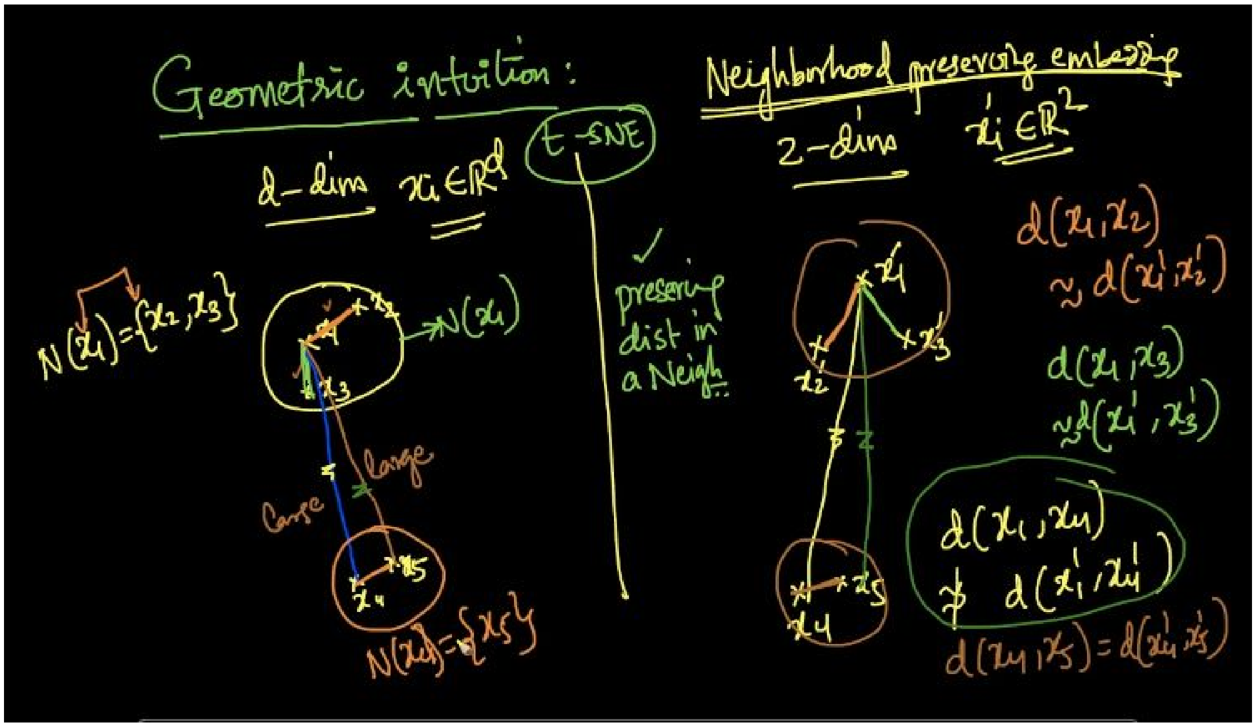


Neighborhood of a point, Embedding





Geometric intuition of t-SNE


Geometric intuition:
Neighborhood preserving embedding
t-SNE
d-dim $x_i \in \mathbb{R}^d$
2-dim $x_i' \in \mathbb{R}^2$
$N(x_1)=\{x_2,x_3\}$
$N(x_1)$
$N(x_4)=\{x_5\}$
large
large
preserving dist in a Neigh..
$d(x_1,x_2) \approx d(x_1',x_2')$
$d(x_1,x_3) \approx d(x_1',x_3')$
$d(x_1,x_4) \not\approx d(x_1',x_4')$
$d(x_4,x_5) = d(x_4',x_5')$

More Intuition: https://www.youtube.com/watch?v=NEaUSP4YerM

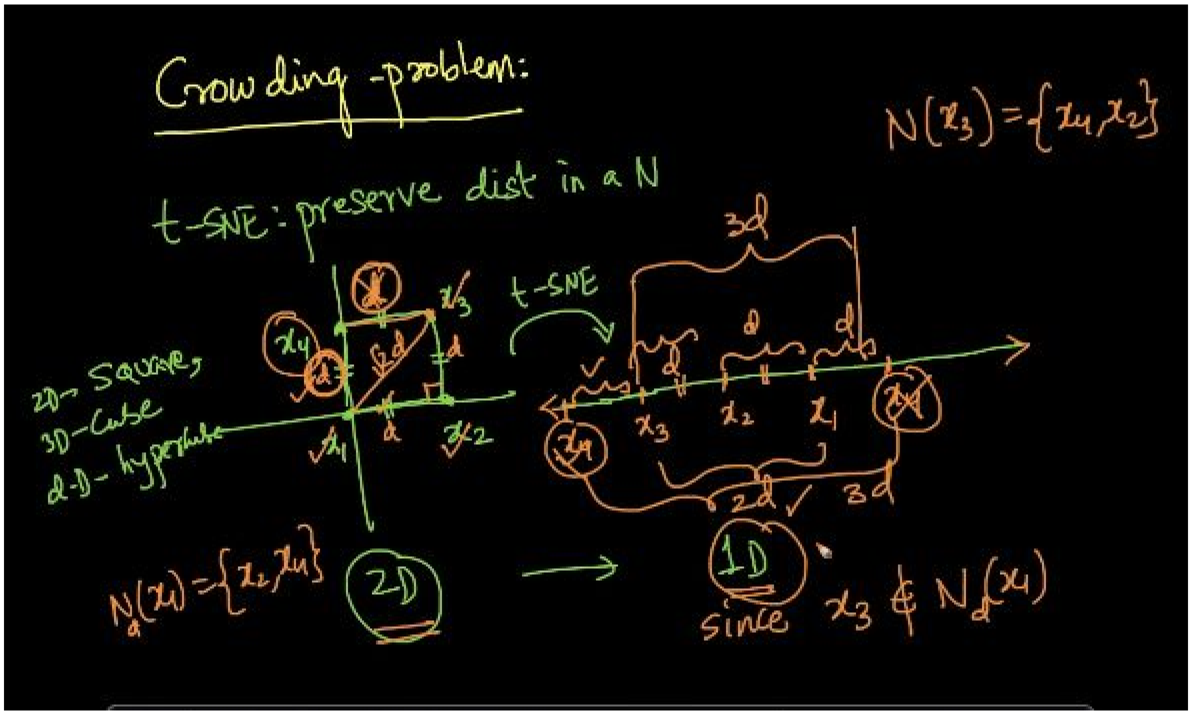
Mathematical: https://www.youtube.com/watch?v=ohQXphVSEQM

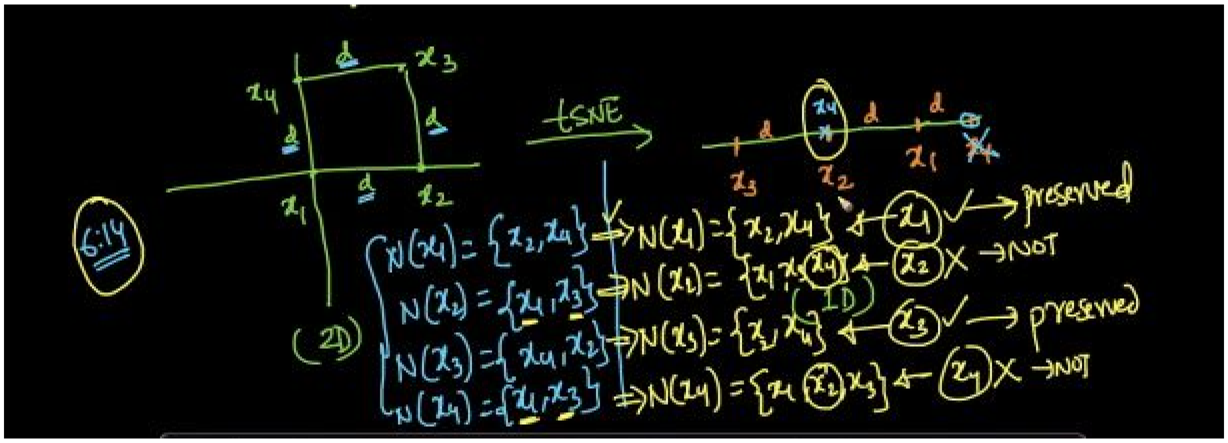
Crowding Problem

Sometimes it is impossible to preserve all the distance in Neighborhood.Such problems are called crowding problems.

Before t-SNE, there was Stochastic Neighbor Embedding, they started observing the problems of crowding. So they decided to use t-SNE

t- SNE = t distribution SNE =Student’s t distribution (Shape is slightly different from Normal /Gaussian distribution)


Crowding problem:
t-SNE: preserve dist in a N
$N(x_3) = \{x_4, x_2\}$
2D → Square, 3D → Cube, d-D → hypercube
t-SNE
3d
d
2d
x_1 x_2 x_3 x_4
$N_d(x_1) = \{x_2, x_4\}$
2D → 1D
since $x_3 \notin N_d(x_1)$


tSNE
6:14
(2D)
$N(x_1) = \{x_2, x_4\}$
$N(x_2) = \{x_1, x_3\}$
$N(x_3) = \{x_4, x_2\}$
$N(x_4) = \{x_1, x_3\}$
$N(x_1) = \{x_2, x_4\}$ ← x_1 ✓ → preserved
$N(x_2) = \{x_1, x_3, x_4\}$ ← x_2 X → NOT
$N(x_3) = \{x_2, x_4\}$ ← x_3 ✓ → preserved
$N(x_4) = \{x_1, x_2, x_3\}$ ← x_4 X → NOT
(1D)

Primary objective of t-SNE is to visualize. So we can clearly see upper configurations are not the same.

<u>How to apply t-SNE and interpret its output</u>

Reference : https://distill.pub/2016/misread-tsne/

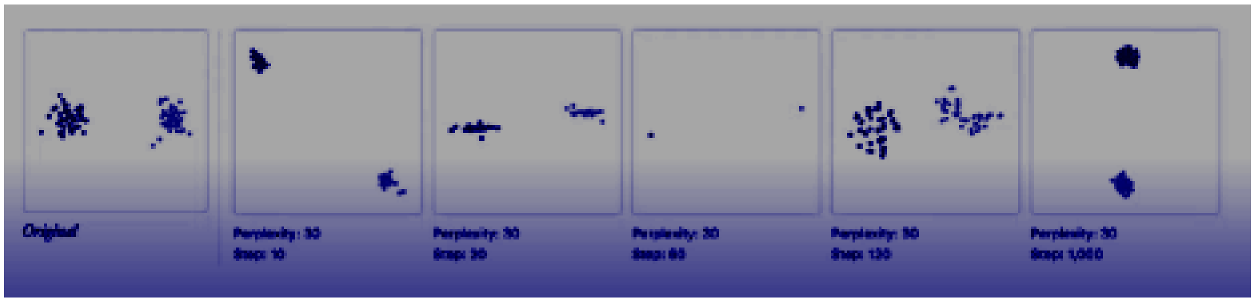
Parameters: Perplexity, step, and t-sne stochastic

Lessons :

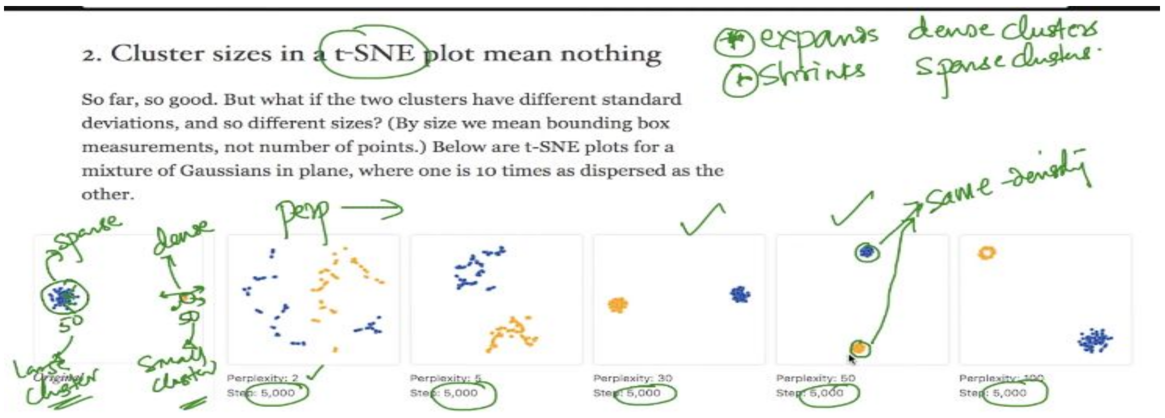
- Never run your t-SNE only once. Run with multiple perplexity values
- When perplexity is equal to the no of original data points, it will simply create a mess. Best practice is to run t-SNE with perplexity<n .

  This can be explained as When perplexity=n, it tries to protect the whole data set creating a mess
- Keeping perplexity constant, and changing iterations(steps) will bring a constant or stabilized shape.

  Figure belows clearly summarizes the stabilized shape with changeable iteration:



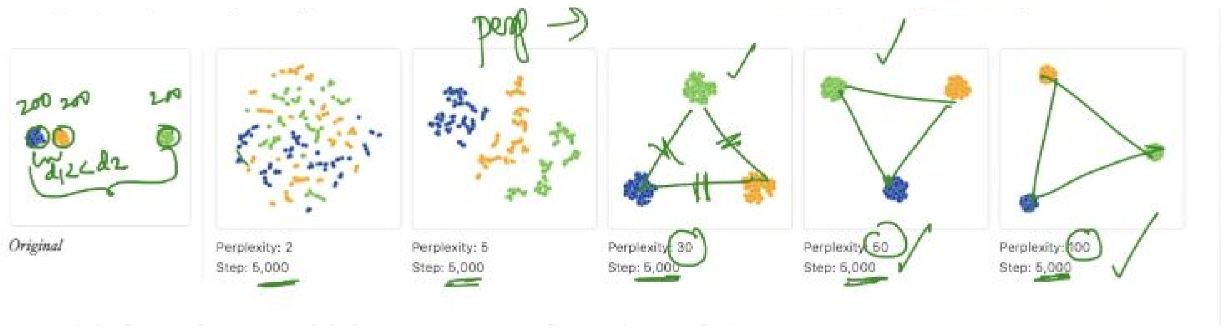
- When we run t-sne multiple times with the same parameters, we could get slightly different results because t-sne is not a deterministic algorithm. The t-distribution in t-sne clearly speaks for itself as it is a probabilistic algorithm.
- T-sne basically expands dense clusters and shrinks sparse clusters to create similar groups of clusters. ***We can't conclude cluster size from t-sne . This is a drawback of t-sne.***



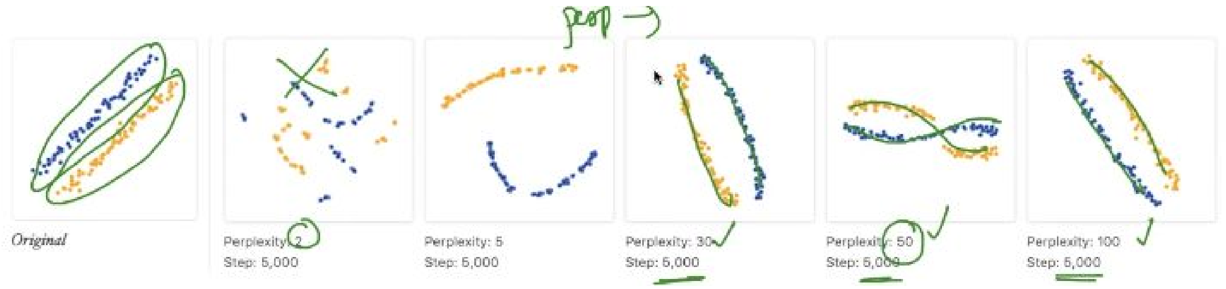
2. Cluster sizes in a t-SNE plot mean nothing

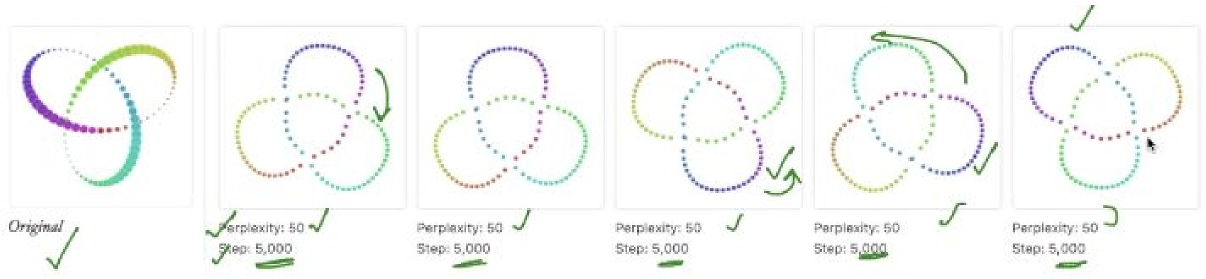
So far, so good. But what if the two clusters have different standard deviations, and so different sizes? (By size we mean bounding box measurements, not number of points.) Below are t-SNE plots for a mixture of Gaussians in plane, where one is 10 times as dispersed as the other.

- We can't conclude a valid conclusion from a distance between clusters. T-sne doesn't preserve distance between clusters.



- Never learn or conclude with small perplexity.



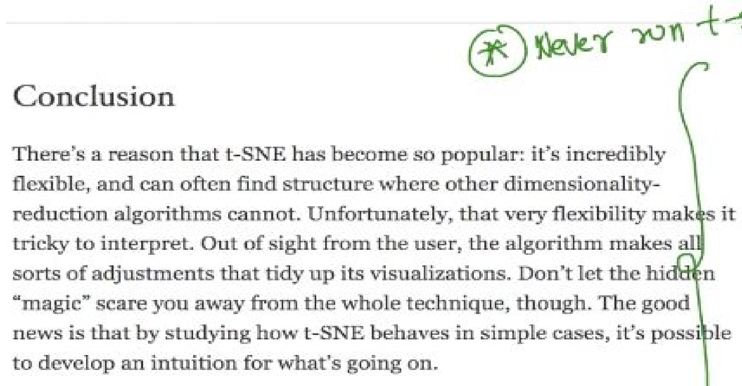


(*) Never run t-SNE just once

1) run steps/iter till shapes stabilize

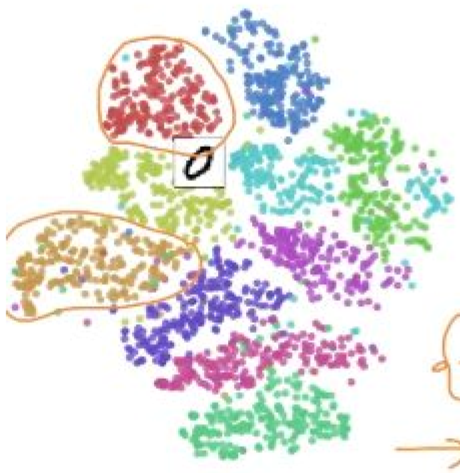
2) perplexity $\boxed{2 \le p < n}$

3) re-run t-SNE p, step ↳ stable or not



Conclusion

There's a reason that t-SNE has become so popular: it's incredibly flexible, and can often find structure where other dimensionality-reduction algorithms cannot. Unfortunately, that very flexibility makes it tricky to interpret. Out of sight from the user, the algorithm makes all sorts of adjustments that tidy up its visualizations. Don't let the hidden "magic" scare you away from the whole technique, though. The good news is that by studying how t-SNE behaves in simple cases, it's possible to develop an intuition for what's going on.


A t-SNE plot of MNIST

$$\text{✓ MNIST: 784 dim} \downarrow \text{2 dim}$$

$\rightarrow$ [illegible]

{→ Cannot interpret cluster sizes (or) intercluster distances. }

T-SNE : group points based on visual similarity