11/08/2020

1. t-SNE (t-distributed Stochastic Neighbour Embedding)

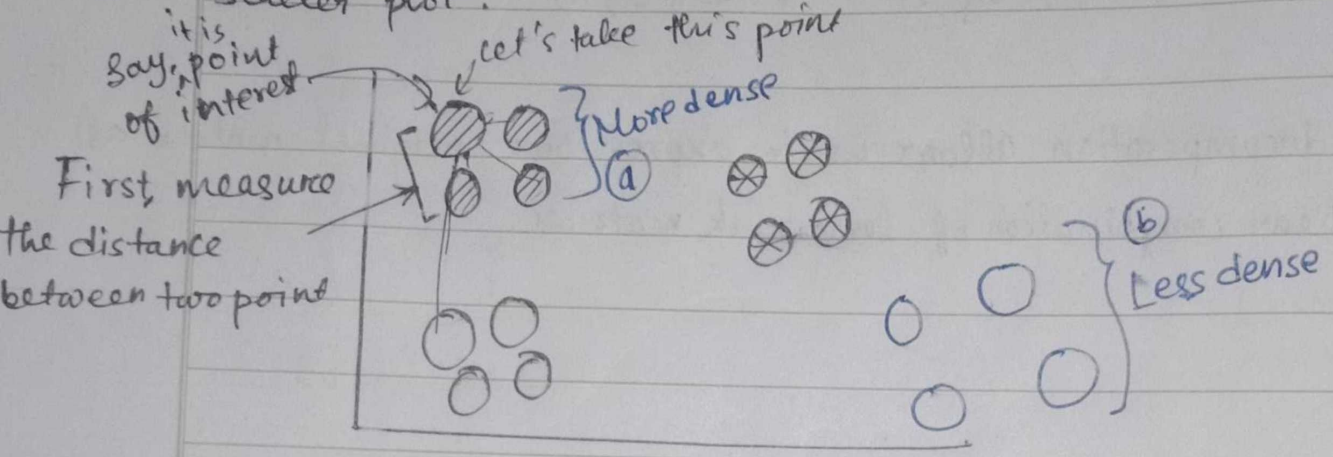
Takes a high dimensional dataset and returns a low dimensional graph.

[It models the probability distribution of neighbours around each point.]

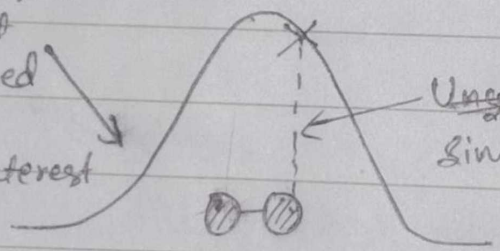
Neighbors → set of points which are closest to each point

How t-SNE does what it does?

→ **Step 1:** Determine the 'similarity' of all the points in the scatter plot.

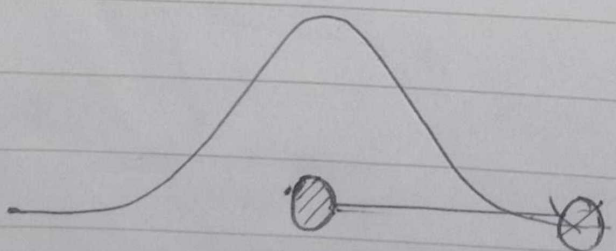


Then plot that distance on a normal curve that is centered on the point of interest



(Close points have high similarity values)

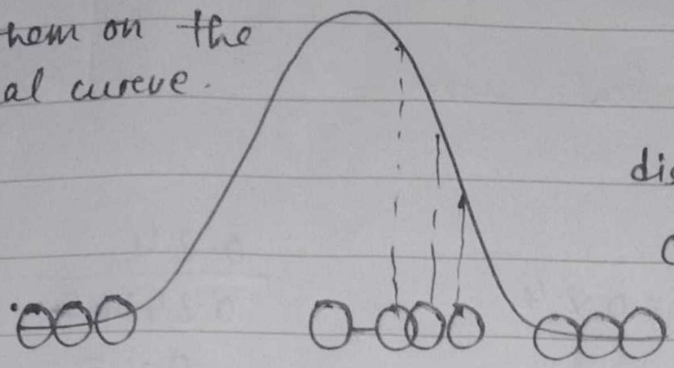
Then calculate unscaled similarity for every pair of points



Using a normal distribution means that distant points have very low similarity values

Ultimately, we measure the distances between all of the points and the point of interest

Plot them on the normal curve.

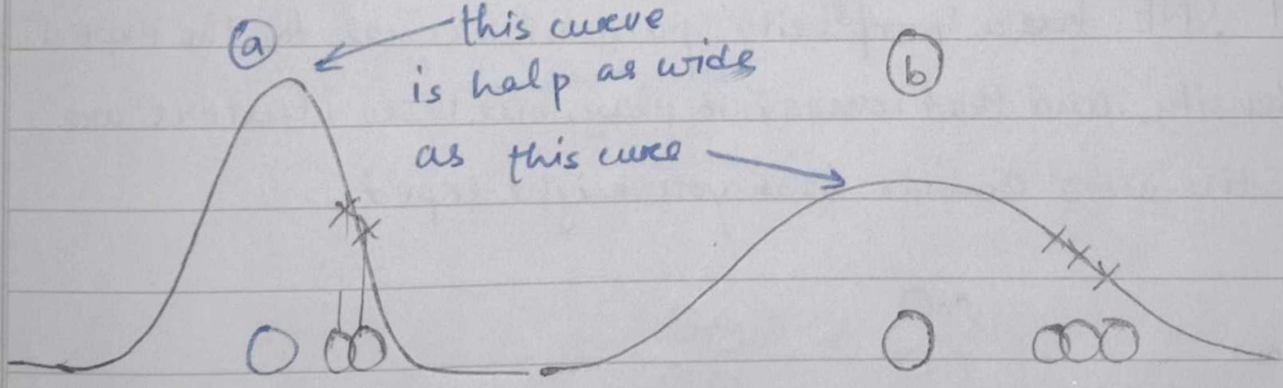


... and then measure the distances from the points to the curve to get the unscaled similarity scores with respect to the point of interest.

The next step to scale the uncaled similarities so that they add up to 1.

Umm... Why do the similarity scores need to add up to 1?

The width of normal curve depends on the density of data near the point interest

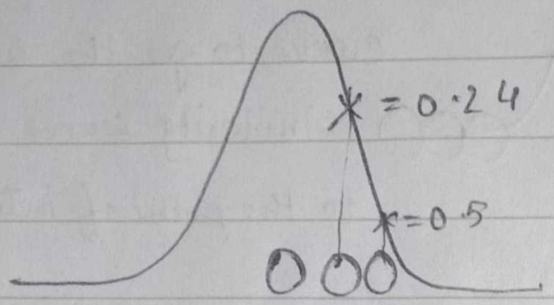


... then scaling the similarity scores will make them the same for both clusters.

Example:

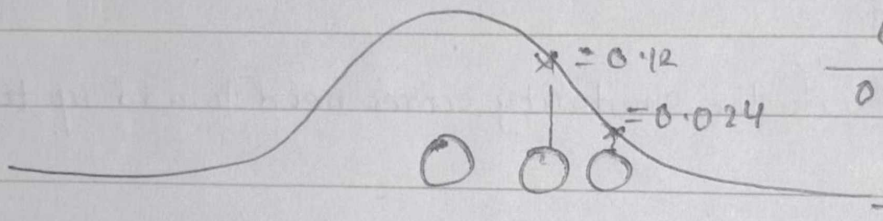
To scale the similarity scores so they sum to 1

$$\frac{\text{Score}}{\text{Sum of all scores}} = \text{Scaled score}$$



$$\frac{0.24}{0.24+0.5} = 0.82$$

$$\frac{0.05}{0.24+0.5} = 0.18.$$

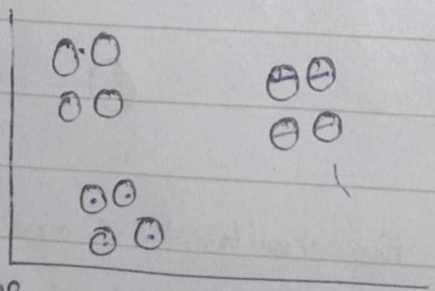


$$\frac{0.12}{0.12+0.024} = 0.82$$

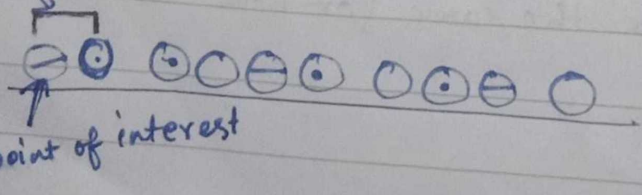
$$\frac{0.024}{0.12+0.024} = 0.18$$

t-SNE has a 'perplexity' parameter equal to the expected density, and that comes into play, but those clusters are still more similar than you might exped.

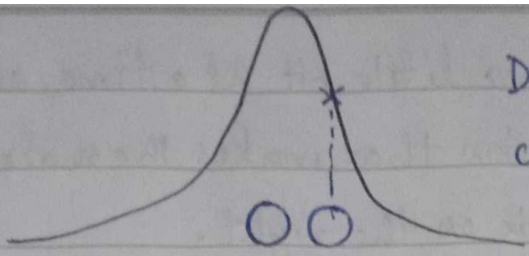
Now we randomly project the data onto the number line



Measuring a distance



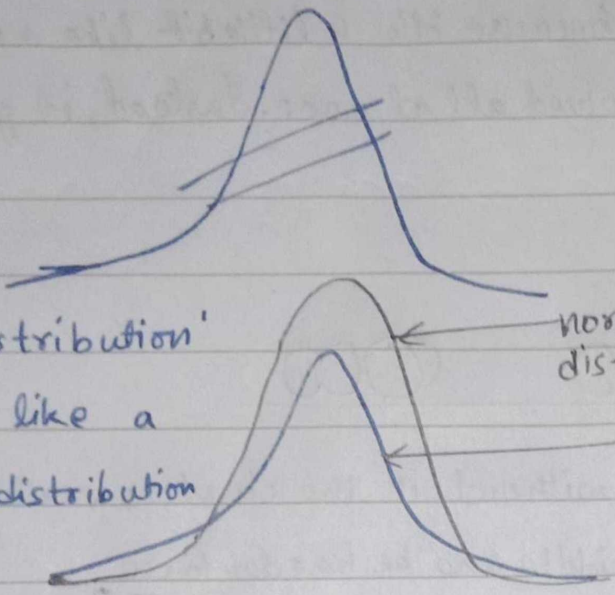
and calculate the similarity scores for the points on the number line.



Draw a line from the point to a curve. However, this time we're using a "t-distribution"

In high-dimensional space, this is modeled as a Gaussian distribution.

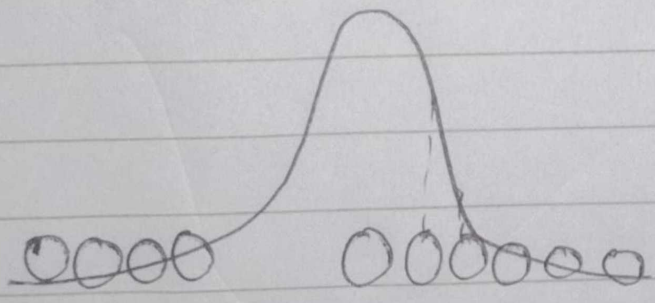
In 2-dimensional output space this is modeled as a ~~Gaussian dist~~ t-distribution.



A 't-distribution' is a lot like a normal distribution

except the t isn't as ~~smart~~ tall in the middle and the tails are taller on the ends

The 't-distribution' is the "t" in t-SNE

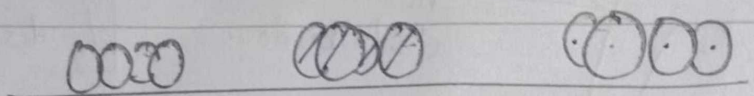


So, using a t-distribution, we calculate "unscaled" similarity scores for all the points and then scale them like before

Like before, we end up with a matrix of similarity scores, but this matrix is a ~~mess~~ mess

t-SNE moves the points a little bit at a time, and each step it chooses a direction that makes the matrix on the left more like the matrix on the right.

It uses small steps, because it's a little bit like a ~~coh~~ chess game and can't be solved all at once. Instead, it goes one move at a time.



distribution (t-SNE) is used as without it the clusters would all clump up in the middle and be harder to see.