PAGE No.
DATE / /

TSNE - T-distributed stochastic Neighbor embedding

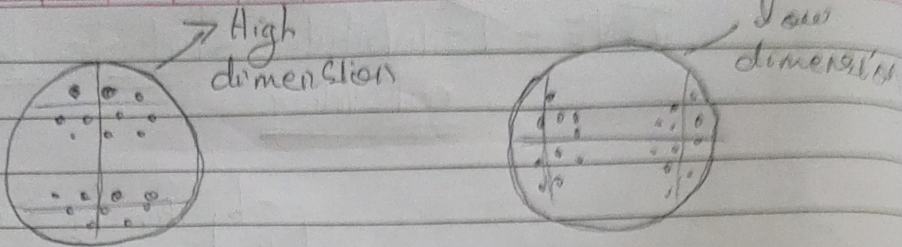
- by van der Matten and Hinton in 2008.

(Non-linear visualization technique)

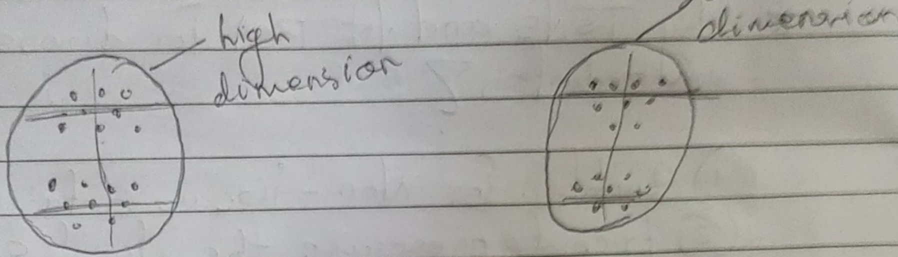
* why T-SNE and Not PCA for dimension reduction?

1) Suitable for Non-linear data.
2) tries to preserve the local structure (cluster) of data
3) can handle outlier.

* when there is 100 or 1000 Dimension to 2, to 3 colums will do TSNE to visualize data
* important thing we are working on features and Not the columns.
* PCA change orientation of the data set but whereas in T-SNE it doesnt it maintain the structure of visualization.



PCA algorithm (changing orientation)



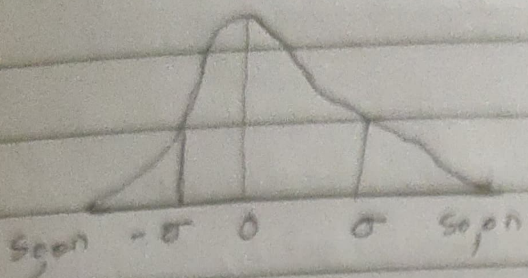
TSNE algorithms.

Note :- T-SNE is only useful for data visualization. and dimension reduction.

* PCA is really bad with non linear dataset
* TSNE can make up 3Dimension visualization
* whereas in TSNE will learn the high dimension data distribution and it will try to imitate in small dimension.

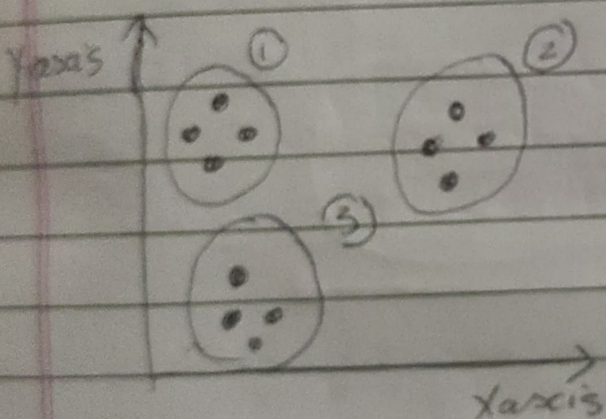
MDS, Sammon mapping, Graph based technique

★ what is Gaussian distribution
=> it's a Normal distribution



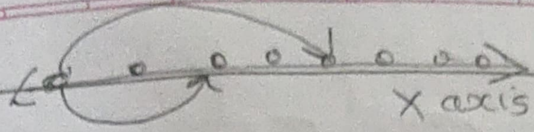
★ <u>How T-SNE work</u>

Let understand with an example.

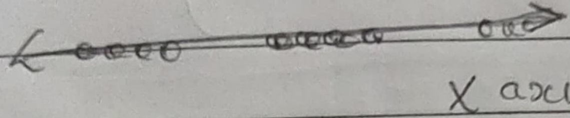


we have this dataset plotted in Scatter plot by seeing this data we can easily understand that it had 3 cluster but t-sne does is
it take 2D data and plot in 1D space.

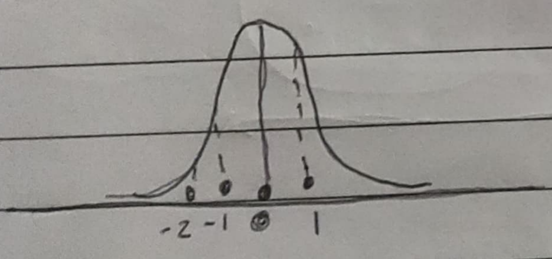
Now data is plot randomly in 1D Space.



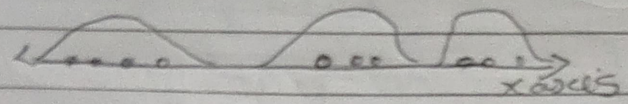
Now whoever datapoint have similarity with them it come closer and closer and eventually it form cluster among themselves



let see How actually work first tsne take one data point and measure the distance between the and plot it in T-distribution curve

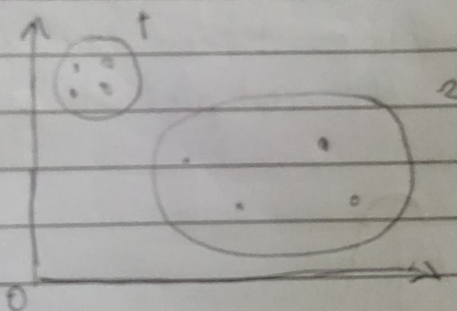


Now current datapoint will measure conditional probability with others (what is probability with current datapoint to Neighbor datapoint. but before proceeding we have to mention k value how many cluster we want, and Next tsne will calculate conditional probabilit with them thoes who have higher conditional probability it will group them.

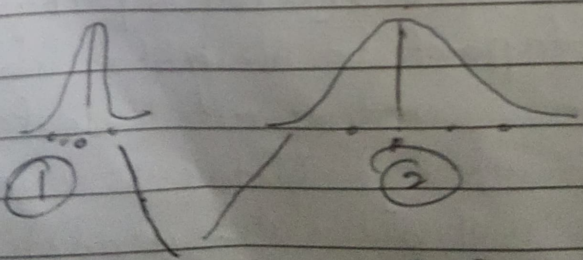


and in T-sNe the proximity measure is conditional propability

but what if cluster are more spread?



the width of the curve depend on density



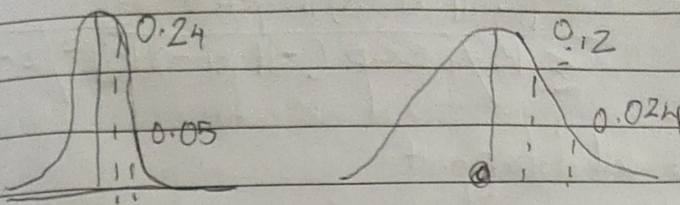
if 1 is half of the 2 then they are similarity score same

Note :- when cluster deviation is more spread we will use scalling.

~~(1) Randomly Gaussian will be selected~~

(2) Scalling.

let take one example.



$\dfrac{0.24}{0.24+0.05} = 0.82$ | $\dfrac{0.12}{0.12+0.024} = 0.82$

$\dfrac{0.05}{0.24+0.05} = 0.18$ | $\dfrac{0.024}{0.12+0.024} = 0.18$

$\overline{\quad 1 \quad}$ | $\overline{\quad 1 \quad}$

So the big and small cluster is same just a difference of deviation. in this situation when cluster is more spread we will use scalling so who ever got highest values is the best cluster. if we dont address likelyhood then those who have deviation is smaller then it will have more liklihood then those who have more deviation.

Equation for normal distribution

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}$$

PAGE No.

DATE / /

we change exponent to conditional probability

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

j = all datapoint.

k = no. cluster

i = data point.

x_i = current datapoint x_k = all datapoint

we will change exponent in normal distribution formula to $P_{j|i}$

Now we capture the essence
~~Now we will~~

create low-dimensional space

to ~~work~~ create low dimesional space we will work on

optimization

loss function (sum of square error)

is to calculate predicted value and actual value difference

For Tsne we use (KLD) (Kullback-leibler Divergence) for loss function)

(two distribution is similar or not)

$(y - \hat{y}) =$ loss function

(loss function should be minimum) is the objective

to get less difference we will use

(Stocastic gradient desent) SGD.
(optimization algorithm)
For loss Function

the objective is to get minimum value of loss function. So we use min Function

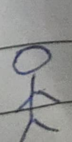
PAGE No.
DATE / /

10,000 → rows but dimension (10,000) (columns) should be 10,000 to 2 or 3. because KLD is only work with 2 or 3 dimension

points should be spread randomly on a new space the goal of this algorithm to find similar probability distribution

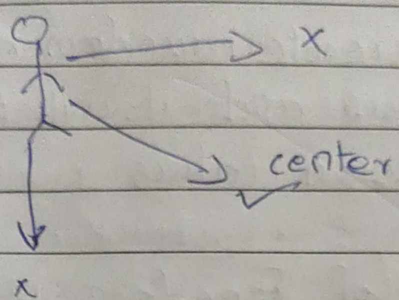
but problem with Gussian is that it has a short tail and because of it creates a crowding problem to solve this we are going to use student t- distribution with single degree of freedom

gradient decent
to understand this we will take an example
suppose we have a person who wants to go in this center

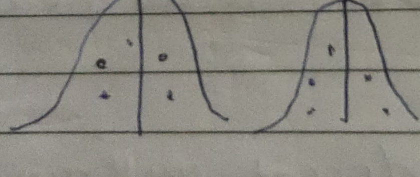


centre

to go in center gredent will try different direction when center is near



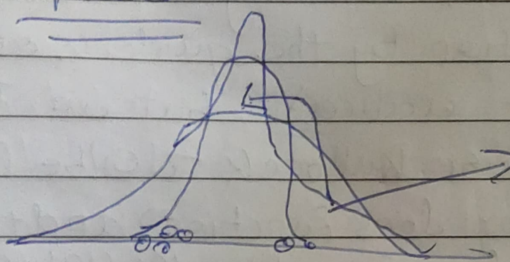
gredient decent will try to minimize the loss function by the guidence of center (mean past values). but in T-sne we will use Kullback-Leilber Divergence (algorithm for loss function) and this will do until loss function value is < 1
if value is less than one then distribution is similar and if 1 > then it is dis similar distribution



student t - distribution

it will create random distribution with New values and it will take help of high dimension values

the degree of freedom will create some space



Degree of freedom

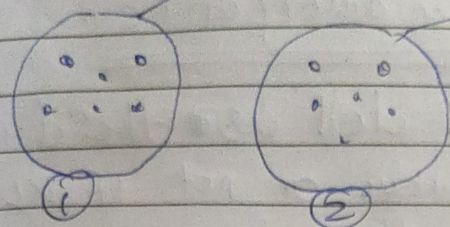
Student t - distrubution has longer tail but why

because in normal distribution it will create ~~creat~~. crowd near to the tail.

Now gredian decent will guide us to distribut values in low dimension space.

high dimension

low dimension



suppose we have distribution called (1) the conditional probability should be similar to (1) (2) space and this will help us through KlD and value of low dimension space is completely random it does not meaning anything because we are only focusing on visualization

Now we will confirm the distribution of both (1) and (2) using r divegence (KL) (KLD).

PAGE NO.
DATE / /

Kullback- leiber divergence.

this (KLD) will tell us the two distribution is similar or not if KL value is 1 or less then one it means it similar if it is greater than 1 it mean it is dissimilar. and 0 means identical.