# t-SNE

Mannem Charan AI21BTECH11019

**Abstract**

This report consists of my basic understanding of one of the modern dimensionality-reduction technique known as ”t-SNE”.

## 1 t-SNE

t-SNE is a unsupervised learning algorithm used for dimensionality reduction but in general well-suited for visualising the high-dimensional data in low-dimensional space.The full form of t-SNE is t-distributed Stochastic Neighbour Embedding.The idea behind t-SNE is that we will try to find **similarity scores** in the high-dimensional space between each data point and we will store these values.After that we will try to calculate the similarity scores in low-dimensional space between each data point and then we will compare these set of values with high dimensional space similarity scores.And we will adjust the data points in low-dimensional space such a way these set of scores are as **similar** as possible.

## 2 Understanding t-SNE

So first we will plot the scatter plot of high-dimensional data.For the sake of simplicity let us take 2-D plot as shown in fig 2.1, As you can there are three clusters and we want to convert this 2-D data into a 1-D plot which retains these characteristics(clusters)as shown in figure 2.2, So finding this line in this context is our job. Simply projecting all points on one of the axes will make mess of this data.So to find the line which best represents the data we first need set up some similarity b/w the data points in the orginal data set so that we can use it later.So for that,t-SNE assumes a **Gaussian distribution** around the target datapoint and the datapoints near to the target datapoint have high likelihood and the datapoints far away has minimum likelihood. And we will renormalize all the likelihood values w.r.t a target datapoint.The reason behind this that sometimes it may happen that the data points in the same cluster(in case of blue dots) have less density(the bell curve will be more spread) which leads to low likelihood values in b/w the cluster members. So the values after normalization are what we are calling as **similarity** values.Mathematically this can be wriiten as,

$$p\left\{x_j|x_i\right\} = \frac{\exp\left(\frac{-\left(x_j-x_i\right)^2}{2\sigma_i^2}\right)}{\sum_{k\neq i}\exp\left(\frac{-(x_k-x_i)^2}{2\sigma_k^2}\right)} \tag{2.1}$$

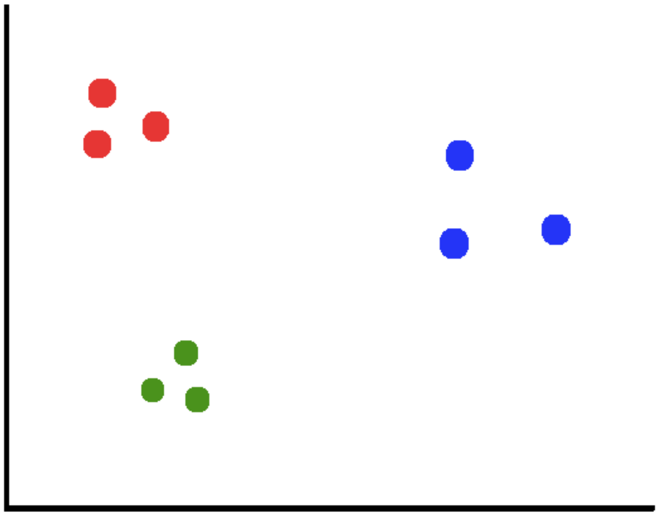

Fig. 2.1: An example of 2-D dataset

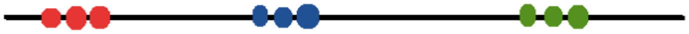

Fig. 2.2: 1-D plot which maintain clear boundaries b/w clusters

where, $p\left\{x_j|x_i\right\}$ is the conditional probability that $x_i$ would pick $x_j$ as its neighbour and this value will be the similarity score.
Now we will try to find similarity scores of data points in 1-D plot, for that let us take a random line and we will project these data points on the line as shown in the fig 2.3, In this case to find similarity scores we will not use gaussian distribution rather we use a yet another popular distribution known as t-distribution.It is similar to the gaussian distribution but it has heavier tails compared to gaussian.It is employed with this job to make the data look more clear.Similar to earlier we will find the conditional probabilities let say $q\left\{x'_j|x'_i\right\}$ where $x'_i$ and $x'_j$ are lower-dimensional counter parts of $x_i$ and $x_j$.
Now to convert the plot 2.3 to 2.2 we will compare these both distributions using something called Kullback-Liebler divergence($KL$).Our goal is to make these similarity scores as close as possible for every pair of data point so that they both map structures the same,to do that we will introduce a cost function which is the sum of these KL

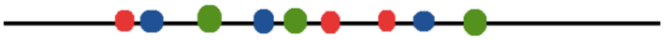

Fig. 2.3: Mess

divergence over all datapoints.

$$\sum_i KL(P_i||Q_i) = \sum_i \sum_j p\left\{x_j|x_i\right\} \log \frac{p\left\{x_j|x_i\right\}}{q\left\{x'_j|x'_i\right\}} \tag{2.2}$$

By minimising the cost function SNE retains the local structure (cluster) of the data in the map.It can be done by using gradient-descent method. The gradient of this cost function looks like,

$$\frac{\partial J}{\partial x'_i} = 2\sum_j \left(p\left\{x_j|x_i\right\} - q\left\{x'_j|x'_i\right\} + p\left\{x_i|x_j\right\} - q\left\{x'_i|x'_j\right\}\right) \tag{2.3}$$

And the last thing to discuss that the parameter $\sigma_i$ of the gaussian distribution is selected based on how the close the data points in each cluster.So depending on the density at regions we should choose the $\sigma_i$.To find these $\sigma_i$s we will use some called perplexity parameter.This will evaluate the standard deviation for each data point.

## 3 Questions

1) What is the full form of t-SNE?
2) What is *t* in t-SNE?
3) How the similarity score is calculated in higher dimensional space in t-SNE?
4) What is the role of perplexity parameter in SNE?
5) What is the main use of t-SNE?

## 4 Answers

1) The full form of t-SNE is t-distributed Stochastic Neighbourhood Embedding.
2) The *t* in t-SNE tells that low-dimensional data takes t-distribution.
3) Using eq (2.1) we can calculate the similarity score in high dimensional space.
4) The perpelexity parameter controls the number of neighbours the datapoint can have and it is a tuneable parameter.

5) t-SNE is used to understand the high dimensional data and project it into low dimensional data.It is more used data visualisation rather as a data preprocessing step.