t-SNE

Aaryan
CO21BTECH11001

t – distributed stochastic neighbor gradient (t-SNE) is a technique that visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map.

Let's say we want to convert the high dimensional data $X = \{x_1, x_2, \dots, x_n\}$ into a 2 or 3 dimensional data $Y = \{y_1, y_2, \dots, y_n\}$.

First, we assume a random initial solution Y.

The aim of the algorithm is to preserve as much of the significant structure of the high-dimensional data as possible in the low-dimensional map.
To do so, we calculate the probability that the neighbor of $x^{(i)}$ is $x^{(j)}$ by fitting a Gaussian centered at $x^{(i)}$.

$$p_{ij} = \frac{\exp\left(-\frac{\left||x^{(i)} - x^{(j)}\right||^2}{2\sigma^2}\right)}{\sum_{k \neq l} \exp\left(-\frac{\left||x^{(k)} - x^{(l)}\right||^2}{2\sigma^2}\right)}$$

where σ is a parameter representing the variance, which is assumed to be constant of whole dataset.

Also, we calculate the probability that the neighbor of $y^{(i)}$ is $y^{(j)}$ by fitting a Student t-distribution centered at $y^{(i)}$.

$$q_{ij} = \frac{\frac{1}{1 + \left||y^{(i)} - y^{(j)}\right||^2}}{\sum_{k \neq l} \frac{1}{1 + \left||y^{(k)} - y^{(l)}\right||^2}}$$

Note: $p_{ii} = q_{ii} = 0$, because we are only interested in modeling pairwise similarities.

Now, we want to find Y such that the mismatch between $p_{j|i}$ and $q_{j|i}$ is minimized.

A well-known measure to find the distance between two distributions is he Kullback-Leibler divergence.
So, our cost function is the sum of Kullback-Leibler divergences over all datapoints.

$$C = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right)$$

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

t-SNE minimizes C using a gradient descent method.
Here is the algorithm –

Sample initial solution $Y^{(0)} = \{y_1, y_2, \dots, y_n\}$.
Maximum number of iterations = T
for $t \in [1, T]$:
    $Y^{(t)} = Y^{(t-1)} + \eta \frac{\delta C}{\delta Y} + \alpha(t)\left(Y^{(t)} - Y^{(t-1)}\right)$
end

where η is the learning rate and $\alpha(t)$ is momentum at iteration t, which is added to speed up the optimization.

<u>Questions –</u>

1. What is the major difference between SNE and t-SNE algorithms?
   Ans. SNE uses a Gaussian distribution to model the low dimensional data, while t-SNE uses a Student t-distribution for the same.

2. Why is Student t-distribution used in t-SNE algorithm?
   Ans. It is used to avoid the Crowding Problem faced in SNE algorithm.

3. Which of the following is/are correct?
   (a) $p_{ij} = p_{ji}$
   (b) $p_{ij} \neq p_{ji}$
   (c) $q_{ij} = q_{ji}$
   (d) $q_{ij} \neq q_{ji}$
   Ans. (a), (c)

4. What is the advantage of using t-SNE over PCA?
   Ans. t-SNE is a non-linear method while PCA doesn't work well for non-linearly correlated data.

5. What is the disadvantage of using t-SNE?
   Ans.
   a. t-SNE is a resource-intensive algorithm because it inspects every single data point and measures the distances between every pair of points. Therefore, it takes a bit long to run this algorithm.
   b. t-SNE is not guaranteed to converge to a global optimum of its cost function.