National Economics University, Vietnam
Faculty of Mathematics Economics
Data Science in Economics and Business
Machine Learning 2

Homework Week 2: Stochastic Neighbor Embedding

Student: Nguyễn Anh Tú - ID: 11207333

Problem 1

Explain the math behind t-SNE algorithm.

Solution.

t-distributed Stochastic Neighbor Embedding (SNE) algorithm is a dimensionality reduction algorithm that aims to preserve local structure of the original data in lower subspace. Neighborhood information of high dimensional dataset is embedded into a distribution. Then in lower space, we will try to find points such that their neighborhood distribution is similar to that of the original set. t-SNE is the development of SNE, where the neighborhood distribution is modified to alleviate complicated optimization and "the crowding" problem.

Given a high dimensional dataset, we consider a neighborhood around an input data point $x_i \in \mathbb{R}^D$. Imagine that we have a Gaussian distribution centering around x_i, the probability that x_i chooses another point x_j as its neighbor is in proportion with the density under this Gaussian.

Based on the idea that the distance between two points will customize the probability of neighborhood, the formula that a point x_i chooses x_j as its neighbors is given by:

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

where the Euclid distance of two points is encoded in the numerator and the denominator is sum of pair distances between x_i and other points.

The existence of conditional probability makes it more difficult to optimize the cost function (that will be introduced later). Therefore, final distribution over one pair is symmetrized:

$$p_{ij} = \frac{1}{2N}(p_{i|j} + p_{j|i})$$

The parameter σ_i sets the size of the neighborhood of point x_i, therefore we set different σ for each point. If a point has very low σ, all the probability is in the nearest neighbor. By contrast, if σ is high, every neighbor has uniform weight. Results of SNE depend heavily on the value of σ_i because it defines the neighborhood we try to preserve.

SNE performs a binary search for the value of σ_i that produces a P_i with a fixed perplexity that is specified by the user. The perplexity is defined as:

$$perp(P_i) = 2^{H(P_i)} \text{ where } H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}$$

The perplexity can be interpreted as a smooth measure of the effective number of neighbors. For instance, if P uniform over k elements, perplexity is k. The performance of SNE is fairly robust to changes in the perplexity, and typical values are **between 5 and 50**.

By the end of this step, we have defined the distributions P_{ij} for each pair of points in the high dimensional data set $X = x_1, x_2, \dots, x_n \in \mathbb{R}^D$. In the next step, we wish to find good embedding $Y = y_1, y_2, \dots, y_n \in \mathbb{R}^d$ for some $d < D$ (normally $d = 2$ or $d = 3$). The map points y_i and y_j should correctly model the similarity between the high-dimensional data points x_i and x_j.

In SNE, for the map points $y_1, y_2, \dots, y_n$, we define distribution Q which quite is similar to P as:

$$Q_{ij} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_k \sum_{k \neq l} \exp(-\|y_k - y_l\|^2)}$$

However, this is when the "crowding" problem arises: In high dimension we have more room, points can have a lot of different neighbors. But in lower dimension, we don't have enough room to accommodate all neighbors. Gaussian distribution has quite short tail, which makes further points less likely to be picked as neighbors as the neighborhood probability falls more quickly to zero. The solution to this problem is proposed in t-SNE: we will use Student t-distribution with a single degree of freedom instead of Gaussian form to define distribution Q because t-distribution's long tail will help further points avoid getting squashed into single point. The formula of Q is modified as:

$$Q_{ij} = \frac{\exp(1 + \|y_i - y_j\|^2)^{-1}}{\sum_k \sum_{k \neq l} \exp(1 + \|y_k - y_l\|^2)^{-1})}$$

Now we want Q to be close to P by minimizing their Kullback-Leibler divergence (a measurement of distance between two distribution). We also consider this as t-SNE's cost function:

$$C = KL(P\|Q) = \sum_k \sum_l P_{kl} \log\left(\frac{P_{kl}}{Q_{kl}}\right) = -\sum_k \sum_l P_{kl} \log(Q_{kl}) + \text{ const}$$

t-SNE's objective function is optimized using Gradient Descent method, therefore we need to find its gradients knowing that model's parameters are new data presentation $y_1, y_2, \ldots, y_n$. We denote:

$$Q_{ij} = \frac{\exp(1 + \|y_i - y_j\|^2)^{-1}}{\sum_k \sum_{k \neq l} \exp(1 + \|y_k - y_l\|^2)^{-1})} = \frac{E_{ij}^{-1}}{\sum_k \sum_{k \neq l} E_{kl}^{-1}} = \frac{E_{ij}^{-1}}{Z}$$

Take partial derivative of cost function w.r.t y_i:

$$\begin{aligned}
\frac{\partial C}{\partial y_i} &= \frac{\partial(-\sum_k \sum_l P_{kl} \log Q_{kl})}{\partial y_i} \\
&= \frac{\partial(-\sum_k \sum_l P_{kl} \log E_{kl}^{-1} + P_{kl} \log Z)}{\partial y_i} \\
&= -\sum_k \sum_l P_{kl} \frac{\partial \log E_{kl}^{-1}}{\partial y_i} + \sum_k \sum_l P_{kl} \frac{\partial \log Z}{\partial y_i}
\end{aligned}$$

Here we notice that only distances related to point i have non-zero derivative and the symmetric property gives us $P_{ij} = P_{ji}$ and $E_{ij} = E_{ji}$. Therefore, the number of distances related to point i is doubled, as given below:

$$\frac{\partial C}{\partial y_i} = -2 \sum_j P_{ij} \frac{\partial \log E_{ij}^{-1}}{\partial y_i} + \sum_k \sum_l P_{kl} \frac{1}{Z} \frac{\partial E_{kl}^{-1}}{\partial y_i}$$

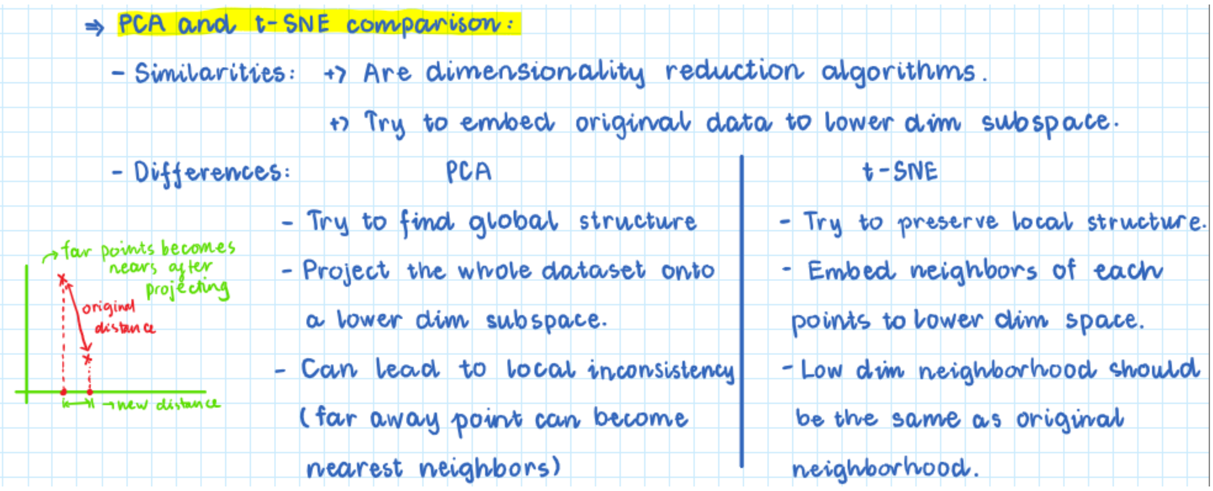
Since $\partial E_{ij}^{-1} = E_{ij}^{-2}(-2(y_i - y_j))$ and using the fact that $\Sigma_{k, l \neq k} p_{kl} = 1$ and Z does not depend on k or l, we have:

$$\begin{aligned}
\frac{\partial C}{\partial y_i} &= \frac{1}{\partial y_i} \cdot \left(-2 \sum_{j \neq i} p_{ji} \frac{E_{ij}^{-2}}{E_{ij}^{-1}} (-2(y_i - y_j)) + \frac{1}{Z} \sum \partial E_{kl}^{-1} \right) \\
&= \frac{1}{\partial y_i} \cdot \left(4 \sum_{j \neq i} p_{ji} E_{ij}^{-1} (y_i - y_j) + 2 \sum_{j \neq i} \frac{E_{ij}^{-2}}{Z} (-2(y_i - y_j)) \right) \\
&= 4 \sum_{j \neq i} p_{ji} E_{ij}^{-1} (y_i - y_j) - 4 \sum_{j \neq i} q_{ji} E_{ij}^{-1} (y_i - y_j) \\
&= 4 \sum_{j \neq i} (p_{ji} - q_{ji}) E_{ij}^{-1} (y_i - y_j) \\
&= 4 \sum_{j \neq i} (p_{ji} - q_{ji}) (1 + ||y_i - y_j||^2)^{-1} (y_i - y_j) \quad \square
\end{aligned}$$

Problem 4

Compare PCA and t-SNE

Solution.



⇒ PCA and t-SNE comparison:

- Similarities: +) Are dimensionality reduction algorithms.
  +) Try to embed original data to lower dim subspace.

- Differences:

PCA	t-SNE
- Try to find global structure	- Try to preserve local structure.
- Project the whole dataset onto a lower dim subspace.	- Embed neighbors of each points to lower dim space.
- Can lead to local inconsistency (far away point can become nearest neighbors)	- Low dim neighborhood should be the same as original neighborhood.