Machine Learning 2 - Homework 2

Linh Nguyen

January 2023

1 Problems

1. Biến đổi lại công thức toán SNE, t-SNE, có tính đạo hàm loss với các parameter
2. Dùng thư viện sklearn, chạy lại với các dataset dưới, nhận xét khi thay đổi perplexity, dataset
3. Dùng word embedding, chọn ra 10 từ bất kì, với mỗi từ tìm 10 từ có embedding gần nhất
   (a) Nhận xét về ngữ nghĩa các từ có embedding gần nhau
   (b) Dùng t-SNE giảm chiều các vector embedding về 2 chiều, nhận xét các cụm
4. So sánh t-SNE và PCA
5. Đọc paper
6. (optional) Tự implement lại t-SNE, variance có thể fix hoặc từ perplexity tìm ra

2 Answers

2.1 SNE

Stochastic Neighbor Embedding (SNE) starts by converting the high-dimensional Euclidean distances between datapoints into conditional probabilities that represent similarities.The similarity of datapoint x_j to datapoint x_i is the conditional probability, $p_{j|i}$, that x_i would pick x_j as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at x_i. Mathematically, the conditional probability $p_{j|i}$ is given by:

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

For the low-dimensional counterparts y_i and y_j of the high-dimensional datapoints x_i and x_j, it is possible to compute a similar conditional probability, which we denote by $q_{j|i}$. Hence, we model the similarity of map point y_j to map point y_i:

$$q_{j|i} = \frac{exp(-||y_i - y_j||^2)}{\Sigma_{k \neq i} exp(-||y_i - y_k||^2)}$$

with $p_{i|i} = 0$ and $q_{i|i} = 0$.
We try to make the map points correctly model the similarity between the datapoints. In other words, we want to achieve $p_{i|i} = q_{i|i}$.
Therefore, SNE aims to find a low-dimensional data representation that minimizes the mismatch between $pj|i$ and $q_{j|i}$. SNE minimizes the sum of Kullback-Leibler divergences over all datapoints using a gradient descent method. The cost function C is given by:

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

The remaining parameter to be selected is the variance σ_i of the Gaussian that is centered over each high-dimensional datapoint, x_i. For each distribution Pi (depends on σ_i) we define the perplexity:

$$perp(P_i) = 2^{H(P_i)}$$

and

$$H(P_i) = -\sum p_{j|i} \log_2 p_{j|i} \text{ is the entropy}$$

2.2 Symmetric SNE

As an alternative to minimizing the sum of the Kullback-Leibler divergences between the conditional probabilities pj|i and qj|i, it is also possible to minimize a single Kullback-Leibler divergence between a joint probability distribution, P, in the high-dimensional space and a joint probability distribution, Q, in the low-dimensional space. Final distribution over pairs is symmetrized:

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

2.3 Gradient of SNE

To perform gradient descent, let derive the cost function with respect to y_i. First, we define

$$q_{j|i} = \frac{exp(-||x_i - x_j||^2)}{\Sigma_{k\neq i} exp(-||x_i - x_k||^2)} = \frac{E_{ij}}{\Sigma_{k\neq i} E_{ik}} = \frac{E_{ij}}{Z_i}$$

Note that $E_{ij} = E_{ji}$. The cost function is defined as

$$\begin{aligned} C &= \sum_{k,l\neq k} p_{l|k} \log \frac{p_{l|k}}{q_{l|k}} = \sum_{k,l\neq k} (p_{l|k} \log p_{l|k} - p_{l|k} \log q_{l|k}) \\ &= \sum_{k,l\neq k} (p_{l|k} \log p_{l|k} - p_{l|k} \log E_{kl} + p_{l|k} \log Z_k) \end{aligned}$$

We derive with respect to y_i. To make the derivation less cluttered, omitting the the ∂y_i term at the denominator.

$$\frac{\partial C}{\partial y_i} = \sum_{k,l\neq k} (-p_{l|k} \partial \log E_{kl} + p_{l|k} \partial \log Z_k)$$

We start with the first term, noting that the derivative is non-zero when $\forall j \neq i$, $k = i$ or $l = i$

$$\sum_{k,l\neq k} -p_{l|k} \partial \log E_{kl} = \sum_{j\neq i} (-p_{j|i} \partial \log E_{ij} - p_{i|j} \partial \log E_{ji})$$

Since $\partial E_{ij} = E_{ij}(-2(y_i - y_j))$, we have

$$\begin{aligned} \sum_{j\neq i} (-p_{j|i} \partial \log E_{ij} - p_{i|j} \partial \log E_{ji}) &= \sum_{j\neq i} \Big[- p_{j|i} \frac{E_{ij}}{E_{ij}} (-2(y_i - y_j)) - p_{i|j} \frac{E_{ji}}{E_{ji}} (2(y_j - y_i)) \Big] \\ &= 2 \sum_{j\neq i} (p_{j|i} + p_{i|j})(y_i - y_j) \end{aligned}$$

We conclude with the second term . Since $\sum_{l\neq j} p_{l|j} = 1$ and z_j does not depend on k, we write (changing variable from l to j to make it more similar to the already computed terms)

$$\sum_{j,k\neq j} = p_{k|j} \partial \log Z_j = \sum_j \partial \log Z_j$$

The derivative is non-zero when $k = i$ or $j = i$ (also, in the latter case we can move Zi inside the summation because constant)

$$
\begin{aligned}
\sum_j \partial \log Z_j &= \sum_j \frac{1}{Z_j} \sum_{k \neq j} \\
&= \sum_{j \neq i} \frac{E_{ji}}{Z_j}(2(y_j - y_i)) + \sum_{j \neq i} \frac{E_{ij}}{Z_i}(-2(y_i - y_j)) \\
&= 2 \sum_{j \neq i} (-q_{j|i} - q_{i|j})(y_i - y_j)
\end{aligned}
$$

Combining two equations, we arrive at the final results

$$\frac{\partial C}{\partial y_i} = 2 \sum_{j \neq i} (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j)$$

2.4 The Crowding Problem

In high dimension we have more room, points can have a lot of different neighbors and In 2D a point can have a few neighbors at distance one all far from each other - what happens when we embed in 1D? This is the ”crowding problem” - we don’t have enough room to accommodate all neighbors. This is one of the biggest problems with SNE.

2.5 t-SNE

Since symmetric SNE is actually matching the joint probabilities of pairs of datapoints in the highdimensional and the low-dimensional spaces rather than their distances, we have a natural way of alleviating the crowding problem that works as follows. In the high-dimensional space, we convert distances into probabilities using a Gaussian distribution. In the low-dimensional map, we can use a probability distribution that has much heavier tails than a Gaussian to convert distances into probabilities. In t-SNE, we employ a Student t-distribution with one degree of freedom (which is the same as a Cauchy distribution) as the heavy-tailed distribution in the low-dimensional map. Using this distribution, the joint probabilities qi j are defined as

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\Sigma_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}$$

2.6 Gradient of t-SNE

Define

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

Notice that $E_{ij} = E_{ji}$. The cost function is defined as

$$\begin{aligned} C = \sum_{k,l \neq k} p_{lk} \log \frac{p_{lk}}{q_{lk}} &= \sum_{k,l \neq k} (p_{lk} \log p_{lk} - p_{lk} \log q_{ik}) \\ &= \sum_{k,l \neq k} (p_{lk} \log p_{lk} - p_{lk} \log E_{lk}^{-1} + p_{lk} \log Z) \end{aligned}$$

We derive with respect to y_i. To make the derivation less cluttered, omitting the ∂y_i term at the denominator.

$$\frac{\partial C}{\partial y_i} = \sum_{k,l \neq k} ({}_{lk} \partial \log E_{lk}^{-1} + p_{lk} \partial \log Z)$$

We start with the first term, noting that the derivation is non-zero when $\forall j$, $k = i$ or $l = i$, that $p_{ji} = p_{ij}$ and $E_{ji} = E_{ij}$

$$\sum_{k,l \neq k} (-p_{lk} \partial \log E_{kl}^{-1} = -2 \sum_{j \neq i} p_{ji} \partial \log E_{ij}^{-1})$$

Since $\partial E_{ij}^{-1} = E_{ij}^{-2}(-2(y_i - y_j))$ we have

$$-2 \sum_{j \neq i} p_{ji} \frac{E_{ij}^{-2}}{E_{ij}^{-1}} (-2(y_i - y_j)) = 4 \sum p_{ji} E_{ij}^{-1} (y_i - y_j)$$

We conclude with the second term. Using the fact that $\sum_{k,l \neq k} p_{kl} = 1$ and that Z does not depend on k or l

$$\begin{aligned} \sum_{k,l \neq k} p_{lk} \partial \log Z &= \frac{1}{Z} \sum_{k',l' \neq k'} \partial E_{kl}^{-1} \\ &= 2 \sum_{j \neq i} \frac{E_{ij}^{-2}}{Z} (-2(y_j - y_i)) \\ &= -4 \sum q_{ij} E_{ji}^{-1} (y_i - y_j) \end{aligned}$$

Combining the two equations, we arrive at the final result

$$\frac{\partial C}{\partial y_i} = 4\sum_{j\neq i}(p_{ji} - q_{ji})E_{ji}^{-1}(y_i - y_j)$$

$$\frac{\partial C}{\partial y_i} = 4\sum_{j\neq i}(p_{ji} - q_{ji})(1 + ||y_i - y_j||^2)^{-1}(y_i - y_j$$

2.7 t-SNE vs PCA

PCA	**t-SNE**
It is a linear Dimensionality reduction technique.	It is a non-linear Dimensionality reduction technique.
It tries to preserve the global structure of the data.	It tries to preserve the local structure(cluster) of data.
It does not involve Hyperparameters.	It involves Hyperparameters such as perplexity, learning rate and number of steps.
PCA is a deterministic algorithm.	It is a non-deterministic or randomised algorithm.
It works by rotating the vectors for preserving variance.	It works by minimising the distance between the point in a gaussian.