MACHINE LEARNING 2

Homework Week 2

1. t-distributed Stochastic Neighbor Embedding

Answer

SNE converts euclidean distances to similarities, that can be interpreted as probabilities.

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

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

with

$$p_{i|i} = 0, q_{i|i} = 0$$

We are reducing the dataset dimension, so that the pair-wise similarity (or distribution) should stay the same.
In other word, our target is to find y so that:

$$p_{i|j} = q_{i|j}$$

The breaking point where "SNE" turned into "t-SNE"

We have that:

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

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

But, the thing is that, let's say x_j is a point in the dataset, with $j \neq i$, the further x_j is, the lower the $p_{j|i}$ and, at the extreme point, this value will approach and

will be 0.
Moving on, notice that, we have the σ^2, the higher the σ the more spread out the Gaussian distribution is, so that, with lower standard deviation, the further point from x_i will have a higher probability.
Let's remind our self back to the purpose of the algorithm. The goal is to find similar probability distribution in lower-dimensional space. The most obvious choice for new distribution would be Gaussian distribution, but that's not the case here. One of the properties of Gaussian is that it has a "short tail" and because of that, it creates a problem called: "the crowding problem". If we use Gaussian again, the data will be crowed (aka stick too close with each other), so we need to use a distribution that has a heavier tail, or more spread out. So the ideal solution is to use t-Student distribution with a single degree of freedom.
Using t-Student distribution has exactly what we need. The distribution falls quickly and has a "long tail" so points won't get squashed into a single point. Welp, by using t-Student, we don't have to care about the σ anymore.

Hence, the name "t"-SNE.

Perplexity

$$Perp(P) = 2^{H(P)}$$

The larger the perplexity, the more non-local information will be retained in the dimensionality reduction result

Perplexity parameter in t-SNE sets the effective number of neighbours that each point is attracted to. In t-SNE optimization, all pairs of points are repulsed from each other, but only a small number of pairs feel attractive forces.

So if perplexity is very small, then there will be fewer pairs that feel any attraction and the result will tend to look like a round bubble shape.

On the other hand, if perplexity is large, clusters will tend to shrink into denser structures.

However, at some point, larger perplexity does not make much difference but the run time is significantly longer.

Kullback-Leiber Divergence

If the map points y_i and y_j correctly model the similarity between the high-dimensional datapoints x_i and x_j, the conditional probabilities $p_{j|i}$ and $q_{j|i}$ will be equal. Motivated by this observation, SNE aims to find a low-dimensional data representation that minimizes the mismatch between $p_{j|i}$ and $q_{j|i}$. A natural measure of the faithfulness with which $q_{j|i}$ models $p_{j|i}$ is the KullbackLeibler divergence (which is in this case equal to the cross-entropy up to an additive constant). The cost function C is given by

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

in which P_i represents the conditional probability distribution over all other datapoints given datapoint x_i, and Q_i represents the conditional probability distribution over all other map points given map point y_i. Because the Kullback-Leibler divergence is not symmetric, different types of error in the pairwise distances in the low-dimensional map are not weighted equally. In particular, there is a large cost for using widely separated map points to represent nearby datapoints (i.e., for using a small $q_{j|i}$ to model a large $p_{j|i}i$, but there is only a small cost for using nearby map points to represent widely separated datapoints. This small cost comes from wasting some of the probability mass in the relevant Q distributions. In other words, the SNE cost function focuses on retaining the local structure of the data in the map (for reasonable values of the variance of the Gaussian in the high-dimensional space, σ_i).

Loss calculation

Define

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

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

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

We derive with respect to y_i.

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

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

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

Since $\frac{\partial E_{ij}^{-1}}{\partial y_i} = E_{ij}^{-2}\left(-2\left(y_i - y_j\right)\right)$, then

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

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

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

Combining we arrive at the final result

$$\begin{aligned}\tfrac{\partial C}{\partial y_i} &= 4\textstyle\sum_{j\neq i}\left(p_{ji} - q_{ji}\right)E_{ji}^{-1}\left(y_i - y_j\right)\\ \tfrac{\partial C}{\partial y_i} &= 4\textstyle\sum_{j\neq i}\left(p_{ji} - q_{ji}\right)\left(1 + \|y_i - y_j\|^2\right)^{-1}\left(y_i - y_j\right)\end{aligned}$$