# T-SNE (week 2 - NL2)

17.1.23.

① Biến đổi lại công thức toán SNE, t-SNE, có tính đạo hàm loss với các parameter

② **SNE basic idea:**

- "Encode" high dimensional neighborhood information as a distribution
- Find low dimensional points such that their neighborhood distribution is similar
- Intuition: Random walk between data points.
  + High probability to jump to a close point
- How do you measure distance between distribution?.
  + Most common measure: KL divergence.

**SNE Implementation:**

- Consider the neighborhood around an input data point $x_i \in \mathbb{R}^d$
- Imagine that we have a Gaussian distribution centered around $x_i$
- Then the probability that $x_i$ chooses some other datapoint "$x_j$" as its neighbor is proportional with the density under this Gaussian.
- A point closer to $x_i$ will be more likely than one further away

we have the probability that point $x_i$ chooses $x_j$ as it neighbor:

$$p_{j|i} = \frac{e^{\{-\|x^{(i)} - x^{(j)}\|^2 / 2\sigma_i^2\}}}{\sum_{k \neq i} e^{\{-\|x^{(i)} - x^{(k)}\|^2 / 2\sigma_i^2\}}}$$

As $j \to j$, $p_{i|i} = 0$, $p_{i|j} \neq p_{j|i}$

Final distribution over pairs is symmetrized:

$$p_{ij} = \frac{1}{2N}(p_{i|j} + p_{j|i})$$ → The distribution of original input data. [1]

Perplexity:

- The parameter $\sigma_i$ sets the [redacted] size of the neighborhood
  - ⇒ very low $\sigma_i$ - all the probs to all the nearest neighbor
  - ⇒ very high $\sigma_i$ - uniform weights.
- Here we set $\sigma_i$ apparently per each data point
- Results depend heavily on $\sigma_i$ - it depends the neighborhood we are trying to preserve.

- For each distribution $P_{j|i}$ (depend on $\sigma_i$) we define the perplexity

$$perp(P_{j|i}) = 2^{H(P_{j|i})}$$

with $H(P) = -\sum P_i \log(P_i)$ is the entropy.

- If P is uniform over k elements - perplexity = k.
  low perx = small $\sigma$
  high perx = large $\sigma$
- Values between 5-50 usually work well
- Important parameter - different perx can capture different scales in the data.

SNE objective

- Given $x^{(1)}, x^{(2)}, \dots, x^{(N)} \in \mathbb{R}^D$ we defined the distribution $P_{ij}$
- Goal: Find good embedding $y^{(1)}, \dots, y^{(N)} \in \mathbb{R}^d$ for some $d < D$ (normally 2 or 3) → distribution of reduced dimension

$x_1 \xrightarrow{\text{reduced}} y_1$
$x_2 \longrightarrow y_2$
$x_n \longrightarrow y_n$

→ parameter as defined → random initialization → distribution Q

- For point $y^{(1)}, y^{(2)}, \dots y^{(N)} \in \mathbb{R}^d$ ($d < D$) we can defined distribution Q similarly the base (notice no $\sigma_i$ and not symmetric)

[2]

$$Q_{ij} = \frac{e^{-\|y^{(i)} - y^{(j)}\|^2}}{\sum_k \sum_{l \neq k} e^{-\|y^{(l)} - y^{(k)}\|^2}}$$

⇒ Optimize Q close P: Minimize KL-divergence → to find the embedding (parameter) $y^{(1)}, \dots, y^{(N)} \in R^d$

Mesure the distance between two distribution P and Q (loss func)

$$L(Q) = KL(Q \| P) = \sum_{ij} Q_{ij} \log\left(\frac{Q_{ij}}{P_{ij}}\right)$$

KL properties: $KL(Q \| P) \geq 0$ and zero only when $Q = P$.
$KL(Q \| P)$ is a convex function

Now,
we have P, are looking for $y^{(1)}, \dots, y^{(N)} \in R^d$ such that dist Q will minimize $L(Q) = KL(P \| Q)$

$$L(Q) = KL(P \| Q) = \sum P_{ij} \log\left(\frac{P_{ij}}{Q_{ij}}\right) = -\sum_{ij} P_{ij} \log(Q_{ij}) + \text{const}$$

Taking gradient w.r.t $y(i)$

$$\frac{\partial L}{y^{(i)}} = \sum_j (P_{ij} - Q_{ij})(y^{(i)} - y^{(j)})$$

Recall: we defined the similarity between point (i) and (j) as a conditional probability:

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

$P_{j|i}$ represents the probability "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 $x_i$" We let $P_{i|i} = 0$

Low dimentional mapping

We wish to obtain $(y_i)_i$ belong to a lower dimentional space, where each $y_i$ represent a point in the plaine correspondig to $x_i$

We depind the similarity between point i and j in this space

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

let $q_{i|i} = 0$

A good map from $(x_i)_i$ to $(y_i)_i$ is a map for which distribution $P_i : j \mapsto p_{j|i}$ and $Q_i : j \mapsto q_{j|i}$ are equal (for all i)

~~We measure the distance between~~

Let define : $q_{j|i} = \frac{e^{-\|y_i - y_j\|^2}}{\sum_{k \neq i} e^{-\|y_i - y_k\|^2}} = \frac{E_{ij}}{\sum_{k \neq i} E_{ik}} = \frac{E_{ij}}{Z_i}$

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

$$C = \sum_{k, l \neq k} p_{l|k} \log \frac{p_{l|k}}{q_{l|k}} = \sum_{k, l \neq k} p_{l|k} \log p_{l|k} - p_{l|k} \log q_{l|k}$$

$$= \sum_{k, l \neq k} p_{l|k} \log p_{l|k} - p_{l|k} \log E_{kl} + p_{l|k} \log Z_k .$$

We derive wrt to $y_i$. To make the derivation less cluttered, I will omit the $\partial y_i$ term at the denominator

$$\frac{\partial C}{\partial y_i} = \left( \sum p_{l|k} \log p_{l|k} - p_{l|k} \log E_{kl} + p_{l|k} \log Z_k \right) \frac{\partial}{\partial y_i}$$

$$= \sum_{k, l \neq k} - p_{l|k} \partial \log E_{kl} + \sum_{k, l \neq k} p_{l|k} \partial \log Z_k$$

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:

$$\sum_{j\neq i} -p_{j|i}\,\frac{E_{ij}}{E_{ij}}\,(-2(y_i - y_j)) - p_{i|j}\,\frac{E_{ji}}{E_{ji}}\,(2(y_j - y_i))$$

$$= 2\sum_{j\neq i} (p_{j|i} + p_{i|j})(y_i - y_j)) \quad (4)$$

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

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

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

$$= \sum_j \frac{1}{Z_j} \sum_{k\neq j} \partial E_{jk}$$

$$= \sum_{j\neq i} \frac{E_{ji}}{Z_j}\,(2(y_j - y_i)) + \sum_{j\neq i} \frac{E_{ij}}{Z_i}\,(-2(y_i - y_j))$$

$$= 2\sum_{j\neq i} (-q_{j|i} - q_{i|j})(y_i - y_j) \quad (5)$$

Combine (4) + (5) we arrive at the final result, □

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

## t-distributed Stochastic Neighbor Embedding (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 log function is defined as

$$C = \sum_{k,l \neq k} p_{lk} \log \frac{p_{lk}}{q_{lk}} = \sum_{k,l \neq k} p_{lk} \log p_{lk} - p_{lk} \log q_{lk}$$

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

We derive wrt $y_i$. To make the derivative less cluttered, I will omit the $\partial y_i$ term at the denominator.

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

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

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

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

$$-2 \sum_{j \neq i} p_{ji} \frac{E_{ij}^{-2}}{E_{ij}^{-1}} (-2 (y_i - y_j)) = 4 \sum p_{ji} E_{ij}^{-1} (y_i - y_j) \quad (10)$$

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

$$\sum_{k,l \neq k} p_{lk} \partial \log Z = \frac{1}{Z} \sum_{k',l' \neq k'} \partial E_{ke}^{-1}$$

$$= 2 \sum_{j \neq i} \frac{E_{ij}^{-2}}{Z} (-2 (y_j - y_i)) = -4 \sum_{j \neq i} q_{ji} E_{ji}^{-1} (y_i - y_j) \quad (11)$$

Combine (10) + (11), final result:

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

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

Ex 4 So solve t-SNE vs PCA

| PCA | t-SNE |
|---|---|
| ① It is a linear Dimensionality reduction technique | It is a non linear Dimensionality reduction technique. |
| ② It tries to preserve the global structure of the data | It tries to preserve the local structure (cluster) of data |
| ③ It doesn't work well as compared to t-SNE | It is one of the best dimensionality reduction technique. |
| ④ It does not involve hyperparameters | It involve hyperparameters such as perplexity, learning rate and number of steps |
| ⑤ It gets highly affected by outliers | It can handle outliers |
| ⑥ PCA is a deterministic algorithm | It is a non deterministic or randomised algorithm. |
| ⑦ It works by rotating the vectors for preserving variance | It works by minimising the distance between the point in a gaussian |
| ⑧ We can find decide on how much variance to preserve using eigen values | we can not preserve variance instead we can preserve distance using hyperparameters. |