Machine Learning Exercise Sheet 11

Dimensionality Reduction & Matrix Factorization

Homework

t-SNE

Problem 1: The similarity in the low dimensional space is defined as:

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

The objective is to obtain a low-dimensional projection capturing the similarity structure of the high-dimensional data. This is achieved via optimizing the Kullback-Leibler divergence

$$C = \mathrm{KL}(P||Q) = \sum_i \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Please derive the gradient $\frac{\partial C}{\partial y_s}$ for t-SNE for the coordinate y_s in the low dimensional space. Please note that this gradient can be used to update y_s with first-order methods.

Arguably, the most difficult part is to keep track of y_s in various sums. To simplify this we denote the intermediate distance term $d_{ij} = 1 + \|y_i - y_j\|^2$.

Next, we use the chain rule on C, so that we can take the derivative with respect to the individual "interactions" d_{ij} instead of the coordinate y_s.

$$\begin{aligned}\frac{\partial C}{\partial y_s} &= \frac{\partial C(d(y))}{\partial y_s} \\ &= \sum_i \sum_j \frac{\partial C}{\partial d_{ij}} \frac{\partial d_{ij}}{\partial y_s}\end{aligned}$$

$\frac{\partial d_{ij}}{\partial y_s}$ is only non-zero if either $i = s$ or $j = s$. Furthermore, $d_{ij} = d_{ji}$.

$$= 2 \sum_j \frac{\partial C}{\partial d_{sj}} \frac{\partial d_{sj}}{\partial y_s}$$

At this point we can already compute $\frac{\partial d_{sj}}{\partial y_s}$.

$$= 4 \sum_j (y_s - y_j) \frac{\partial C}{\partial d_{sj}}$$

Upload a single PDF file with your homework solution to Moodle by 03.02.2021, 11:59pm CET. We recommend to typeset your solution (using LaTeX or Word), but handwritten solutions are also accepted. If your handwritten solution is illegible, it won't be graded and you waive your right to dispute that.

Now we are left with computing the gradient of C with respect to some d_{nm}.

$$\begin{aligned}\frac{\partial C}{\partial d_{nm}} &= \frac{\partial}{\partial d_{nm}}\left[\sum_i \sum_{j\neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}}\right] \\ &= \frac{\partial}{\partial d_{nm}}\left[\sum_i \sum_{j\neq i} p_{ij} \log p_{ij} - p_{ij} \log q_{ij}\right]\end{aligned}$$

Linearity of differentiation and p is constant with respect to d.

$$= -\sum_i \sum_{j\neq i} p_{ij} \frac{\partial \log q_{ij}}{\partial d_{nm}}$$

Expand the definition of q_{ij}.

$$\begin{aligned}&= -\sum_i \sum_{j\neq i} p_{ij} \frac{\partial}{\partial d_{nm}}\left[\log \frac{d_{ij}^{-1}}{\sum_k \sum_{k\neq l} d_{kl}^{-1}}\right] \\ &= -\sum_i \sum_{j\neq i} p_{ij} \frac{\partial}{\partial d_{nm}}\left[-\log d_{ij} - \log \sum_k \sum_{k\neq l} d_{kl}^{-1}\right] \\ &= \sum_i \sum_{j\neq i} p_{ij} \frac{\partial \log d_{ij}}{\partial d_{nm}} + \sum_i \sum_{j\neq i} p_{ij} \frac{\partial}{\partial d_{nm}} \log \sum_k \sum_{k\neq l} d_{kl}^{-1}\end{aligned}$$

$\frac{\partial \log d_{ij}}{\partial d_{nm}}$ is only non-zero for $i = n$ and $j = m$.

$$= p_{nm} d_{nm}^{-1} + \sum_i \sum_{j\neq i} p_{ij} \frac{1}{\sum_k \sum_{k\neq l} d_{kl}^{-1}} \frac{\partial}{\partial d_{nm}} \sum_k \sum_{k\neq l} d_{kl}^{-1}$$

The same is true for $\frac{\partial d_{kl}^{-1}}{\partial d_{nm}}$.

$$\begin{aligned}&= p_{nm} d_{nm}^{-1} + \sum_i \sum_{j\neq i} p_{ij} \frac{1}{\sum_k \sum_{k\neq l} d_{kl}^{-1}} \frac{\partial d_{nm}^{-1}}{\partial d_{nm}} \\ &= p_{nm} d_{nm}^{-1} - \frac{1}{\sum_k \sum_{k\neq l} d_{kl}^{-1}} d_{nm}^{-2} \sum_i \sum_{j\neq i} p_{ij}\end{aligned}$$

$\sum_i \sum_{j\neq i} p_{ij} = 1$ and we can also find the definition of q_{nm} in there.

$$\begin{aligned}&= p_{nm} d_{nm}^{-1} - \frac{d_{nm}^{-1}}{\sum_k \sum_{k\neq l} d_{kl}^{-1}} d_{nm}^{-1} \\ &= (p_{nm} - q_{nm})\, d_{nm}^{-1}\end{aligned}$$

Finally, we plug this result into $\frac{\partial C}{\partial y_s}$ and resolve the definition of d_{sj}.

$$\frac{\partial C}{\partial y_s} = 4 \sum_j (y_s - y_j) \frac{\partial C}{\partial d_{sj}}$$

$$
\begin{aligned}
&= 4 \sum_j (y_s - y_j)\,(p_{sj} - q_{sj})\, d_{sj}^{-1} \\
&= 4 \sum_j (y_s - y_j)\,(p_{sj} - q_{sj})\, \left(1 + \|y_s - y_j\|^2\right)^{-1}
\end{aligned}
$$

Autoencoders

Problem 2: We train a linear autoencoder to D-dimensional data. The autoencoder has a single K-dimensional hidden layer, there are no biases, and all activation functions are identity ($\sigma(x) = x$).

- Why is it usually impossible to get zero reconstruction error in this setting if $K < D$?
- Under which conditions is this possible?

We have $f(\boldsymbol{x}) = \boldsymbol{X}\boldsymbol{W}_1\boldsymbol{W}_2$ where $\boldsymbol{X}$ is the data matrix and the dimensions of the weight matrices are $D \times K$ for $\boldsymbol{W}_1$ and $K \times D$ for $\boldsymbol{W}_2$.

The final multiplication $\boldsymbol{W}_2$ brings points from K-dimensions up into D-dimensions but the points will still all be in a K-dimensional linear subspace. Unless the data happen to lie exactly in a K-dimensional linear subspace, they can't be exactly fitted.

Coding Exercise

Problem 3: Download the notebook `exercise_11_notebook.ipynb` and `exercise_11_matrix_factorization_ratings.npy` from Piazza. Fill in the missing code and run the notebook. Convert the evaluated notebook to PDF and append it to your other solutions before uploading.