Exercises for the course

Machine Learning 2

Summer semester 2024

Fachgebiet Maschinelles Lernen
Institut für Softwaretechnik und theoretische Informatik
Fakultät IV, Technische Universität Berlin
Prof. Dr. Klaus-Robert Müller
Email: klaus-robert.mueller@tu-berlin.de

Exercise Sheet 2

Exercise 1: SNE and Kullback-Leibler Divergence (50 P)

SNE is an embedding algorithm that operates by minimizing the Kullback-Leibler divergence between two discrete probability distributions p and q representing the input space and the embedding space respectively. In 'symmetric SNE', these discrete distributions assign to each pair of data points (i, j) in the dataset the probability scores p_{ij} and q_{ij} respectively, corresponding to how close the two data points are in the input and embedding spaces. Once the exact probability functions are defined, the embedding algorithm proceeds by optimizing the function:

$$\begin{aligned} C &= D_{\mathrm{KL}}(p \,\|\, q) \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right) \end{aligned}$$

where p and q are subject to the constraints $\sum_{i=1}^{N} \sum_{j=1}^{N} p_{ij} = 1$ and $\sum_{i=1}^{N} \sum_{j=1}^{N} q_{ij} = 1$. Specifically, the algorithm minimizes q which itself is a function of the coordinates in the embedded space. Optimization is typically performed using gradient descent.

In this exercise, we derive the gradient of the Kullback-Leibler divergence, first with respect to the probability scores q_{ij}, and then with respect to the embedding coordinates of which q_{ij} is a function.

(a) *Show* that

$$\frac{\partial C}{\partial q_{ij}} = -\frac{p_{ij}}{q_{ij}}. \tag{1}$$

(b) The probability matrix q is now reparameterized using a 'softargmax' function:

$$q_{ij} = \frac{\exp(z_{ij})}{\sum_{k=1}^{N} \sum_{l=1}^{N} \exp(z_{kl})}$$

The new variables z_{ij} can be interpreted as unnormalized log-probabilities. *Show* that

$$\frac{\partial C}{\partial z_{ij}} = -p_{ij} + q_{ij}. \tag{2}$$

(c) *Explain* which of the two gradients, (1) or (2), is the most appropriate for practical use in a gradient descent algorithm. Motivate your choice, first in terms of the stability or boundedness of the gradient, and second in terms of the ability to maintain a valid probability distribution during training.

(d) The scores z_{ij} are now reparameterized as

$$z_{ij} = -\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2$$

where the coordinates $\boldsymbol{y}_i, \boldsymbol{y}_j \in \mathbb{R}^h$ of data points in embedded space now appear explicitly. *Show* using the chain rule for derivatives that

$$\frac{\partial C}{\partial \boldsymbol{y}_i} = \sum_{j=1}^{N} 4\,(p_{ij} - q_{ij}) \cdot (\boldsymbol{y}_i - \boldsymbol{y}_j).$$

Exercise 2: Programming (50 P)

Download the programming files on ISIS and follow the instructions.