Exercises for the course
Machine Learning 2
Summer semester 2019

Abteilung 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 1

In the first two exercises, we refer to the paper *An Introduction to Locally Linear Embedding* by Lawrence K. Saul and Sam T. Roweis, which is linked from the course website.

Exercise 1: Symmetries (30 P)

In the third paragraph of page 3, it is claimed that the optimal weights W_{ij} which minimize the cost function $\mathcal{E}$ are independent with respect to scaling, translation and rotation of the original data $\vec{X}_i$. Prove that. That is, *prove* that the minimum (or minima) of $\mathcal{E}$ is (or are) invariant under the following symmetries:

(a) Replace all $\vec{X}_i$ with $\alpha\vec{X}_i$, for an $\alpha \in \mathbb{R}^+ \setminus \{0\}$,

(b) Replace all $\vec{X}_i$ with $\vec{X}_i + \vec{v}$, for a vector $\vec{v} \in \mathbb{R}^D$,

(c) Replace all $\vec{X}_i$ with $U \cdot \vec{X}_i$, where U is an orthogonal $D \times D$ matrix (this additionally includes mirror symmetries).

Exercise 2: Lagrange Multipliers (30 P)

In the second paragraph on page 3, it is stated that finding the optimal W_{ij} is a least-squares problem, which is shown to have an explicit analytic solution in Appendix A. In the following, assume the notation of Appendix A. For abbreviation (and clarity of notation), additionally write $w = (w_1, \ldots, w_K)^\top$ for the weight vector which is optimized, $\eta = (\vec{\eta}_1, \ldots, \vec{\eta}_K)^\top$ for the $(K \times D)$-matrix of nearest neighbors of $\vec{x}$, $\mathbb{1} = (1, \ldots, 1)^\top$ for the K-dimensional vector of ones, and

$$C = (\mathbb{1}\vec{x}^\top - \eta)(\mathbb{1}\vec{x}^\top - \eta)^\top$$

for the local covariance matrix at $\vec{x}$. We would like to work out the following claims from Appendix A:

(a) *Prove* that the optimal weights for $\vec{x}$ are found by solving the following optimization problem:

$$\min_w \quad w^\top C w \qquad \text{subject to} \quad w^\top \mathbb{1} = 1.$$

In particular, prove equation (3) on page 9.

(b) *Show* by using the Lagrangian method for constrained optimization that the minimum of the optimization problem is explicitly given by

$$w = \frac{C^{-1}\mathbb{1}}{\mathbb{1}^\top C^{-1}\mathbb{1}}.$$

(c) *Show* that the minimum w can be equivalently found by solving the equation

$$Cw = \mathbb{1},$$

and then rescaling w such that $w^\top \mathbb{1} = 1$.

Exercise 3: Kullback-Leibler Divergence (40 P)

The objective of SNE (and t-SNE) is based on minimization of the Kullback-Leibler divergence between two probability distributions p and q over pairs of data points.

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

where $\sum_{i=1}^{N}\sum_{j=1}^{N} p_{ij} = 1$ and $\sum_{i=1}^{N}\sum_{j=1}^{N} q_{ij} = 1$. In this exercise, we derive the gradient of the Kullback-Leibler divergence, with respect to the probability scores, and a reparameterization of it, and the coordinates that have produced these probabilities.

(a) *Show* that

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

(b) The probability matrix q is now reparameterized as

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

where z_{ij} represent unnormalized log-probabilities. *Show* that

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

(c) *Explain* which of the two gradients, (a) or (b), is the most appropriate for practical use in a gradient descent algorithm. Motivate your choice (1) in terms of stability or boundedness of the gradient, and (2) in terms of 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 of data points $\boldsymbol{y}_i, \boldsymbol{y}_j \in \mathbb{R}^h$ now appear explicitly. *Show* using the chain rule for derivatives that

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