582744 Advanced Course in Machine Learning

Exercise 3 **Due April 4, 23:55 AM**

Rules:

1. Return your solutions in Moodle by the deadline.
2. The submission consists of two parts: (a) A single PDF file containing your answers to all questions. (b) A single file containing your code (either a single plain text source code or a compressed file). Do not include datasets, plots etc in this file, only the code.
3. If you feel comfortable, add an estimate of how many hours you worked on the problems in the beginning of your report.
4. Please typeset your work using appropriate software such as LaTeX. However, there is no need to typeset the pen and paper answers – you can also include a scanned hand-written version.
5. Pay attention to how you present the results. Be concise.
6. Spotted a mistake? Something is unclear? Ask for clarifications in Moodle.

This set of exercises is due on Tuesday April 4, before 23:55 AM.

0 How many hours did you work on these?

1 Mixture model for binary data (3 points)

Consider the model

$$p(z_n) = \text{Categorical}(\boldsymbol{\pi}),$$
$$p(\mathbf{x}_n|z_n = k, \boldsymbol{\mu}) = \prod_{d=1}^{D} \text{Bernoulli}(\mu_{kd}),$$

which defines a mixture model for binary vectors $\mathbf{x}_n \in [0,1]^D$. In verbal terms, the generative process is such that we first pick a cluster index k (with probabilities π_k) and then generate D independent observations based on the parameters μ_{kd}. The notation for the Bernoulli distribution means that the dth element of $\mathbf{x}_n$ has probability μ_{kd} to be one if the nth sample belongs to the kth cluster. To simplify the problem we consider maximum likelihood estimation, not placing any priors on $\boldsymbol{\mu}$ or $\boldsymbol{\pi}$.

Write down *the observed data log-likelihood*, *the complete data log-likelihood*, and *the expected complete data log-likelihood* of the model. Then derive the expectation maximization algorithm for inferring the parameters $\theta = \{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_k, \boldsymbol{\pi}\}$. You can largely follow the derivation for mixture of Gaussians provided in Section 11.4.2 of the course book.

Do you think this would be a useful model in practice?

2 Spectral clustering (programming, 4 points)

Read in the data set ($N = 120$ samples represented in two dimensions) from Moodle and compute the pairwise distance matrix d containing all Euclidean distances between the samples (remember to take the square-root). Draw a scatter-plot of the data to see how it looks like. Run also a k-means algorithm with $K = 2$ clusters, using some publicly available code (all programming environments should have one), and color the data points in the scatter-plot according to the cluster indices.

Next implement the spectral clustering algorithm:

(a) Create two types of adjancency matrices W based on the data:

- Connect each sample to all other samples that are within distance $e = 0.5$ ($W_{i,j} = 1 \quad \text{if} \quad d_{i,j} \leq e$).
- Connect each sample to its A closest neighbors (not counting the sample itself), using $A = 8$. Do this in a symmetric fashion, so that two nodes are connected if either one of them is within the A closest neighbors of the other one.

Then perform the following steps for both alternatives.

(b) Find the eigenvalues and eigenvectors of the graph Laplacian $L = D - W$, where D is a diagonal matrix with the degrees of the nodes on its diagonal. Plot the eigenvectors corresponding to four smallest eigenvalues as a line plot (preferably in a single plot; the x-axis corresponds to the samples and each eigenvector is drawn as a line) – how do they look like?

Hint: The samples in the data matrix are ordered so that the first 60 samples correspond to one of the natural clusters. Furthermore, the samples within each cluster are ordered along the half-circle. This information should help interpreting the eigenvectors.

(c) Now represent the data using the $M = 4$ eigenvectors corresponding to the smallest eigenvalues, creating a new representation $Y \in \mathbb{R}^{120 \times 4}$. Draw scatter-plots of the data in this new representation. How do they look like? Can you see the clusters?

(d) Cluster the data with k-means into two clusters using the new representation Y.

Now compare the three clustering solutions you have: One based on the original data and two based on spectral clustering with different adjacency matrices. What is the difference? Do all three methods solve the clustering problem equally well?

Finally, play around with the numbers e, A and M above to see how things change, answering briefly the following questions. No need to produce separate plots for these, unless you feel it is necessary for understanding your answer.

1. What happens if e is too small? What if it is too big?
2. What happens if A is too small? What it it is too big?
3. Would $M = 2$ be enough? What happens if you use too big M? Why?

3 Deflationary orthogonalization (1 point)

Optimization of ICA (and PCA) requires finding a matrix $\mathbf{W}$ that is orthonormal, meaning that $\mathbf{w}_i^T \mathbf{w}_i = 1$ for all i and $\mathbf{w}_i^T \mathbf{w}_j = 0$ for all $i \neq j$. Here $\mathbf{w}_i$ are the columns of $\mathbf{W}$.

Assume we are given an iterative optimization algorithm for minimizing the loss function over a single vector, updating $\mathbf{w}_i^{t+1} = f(\mathbf{w}_i^t)$, where the superscript t refers to the iteration. Assuming we have already found the first k columns of the solution, we can use the following algorithm to find the next solution:

1. Initialize $\mathbf{w}_i$ randomly
2. Perform one iteration of the optimization algorithm to get updated $\mathbf{w}_i$
3. Normalize the solution vector using

$$\begin{aligned}\mathbf{v}_i &= \mathbf{w}_i - \sum_{j \leq k} (\mathbf{w}_i^T \mathbf{w}_j) \mathbf{w}_j \\ \mathbf{w}_i &= \frac{\mathbf{v}_i}{\sqrt{\mathbf{v}_i^T \mathbf{v}_i}}.\end{aligned}$$

4. Repeat the two previous steps until convergence of the optimization process

Prove that the resulting $\mathbf{W}$ is orthonormal.