Machine Learning Exercise Sheet 12

Dimensionality Reduction & Clustering

Homework

Matrix Factorization

Problem 1: Download the notebook `exercise_12_matrix_factorization.ipynb` and `exercise_12_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.

Autoencoders

Problem 2: We train a linear autoencoder on 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.

Gaussian Mixture Model

Problem 3: Consider a mixture of K Gaussians

$$\mathrm{p}(\boldsymbol{x}) = \sum_k \boldsymbol{\pi}_k \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

Derive the expected value $\mathbb{E}[\boldsymbol{x}]$ and the covariance $\mathrm{Cov}[\boldsymbol{x}]$.

Hint: it is helpful to remember the identity $\mathrm{Cov}[\boldsymbol{x}] = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] - \mathbb{E}[\boldsymbol{x}]\,\mathbb{E}[\boldsymbol{x}]^T$.

Upload a single PDF file with your homework solution to Moodle by 02.02.2020, 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.

For $\mathbb{E}[\boldsymbol{x}]$ we use the law of iterated expectations.

$$\mathbb{E}[\boldsymbol{x}] = \mathop{\mathbb{E}}_{\boldsymbol{z}}\left[\mathop{\mathbb{E}}_{\boldsymbol{x}}[\boldsymbol{x} \mid \boldsymbol{z}]\right] = \sum_{k=1}^{K} \boldsymbol{\pi}_k \, \mathbb{E}[\boldsymbol{x} \mid \boldsymbol{z} = k] = \sum_{k=1}^{K} \boldsymbol{\pi}_k \boldsymbol{\mu}_k$$

For covariance, we first compute $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T]$ again using the law of iterated expectations

$$\begin{aligned}
\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] &= \mathop{\mathbb{E}}_{\boldsymbol{z}}\left[\mathop{\mathbb{E}}_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{x}^T \mid \boldsymbol{z}]\right] \\
&= \sum_{k=1}^{K} \boldsymbol{\pi}_k \, \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T \mid \boldsymbol{z} = k] \\
&= \sum_{k=1}^{K} \boldsymbol{\pi}_k \left(\mathrm{Cov}[\boldsymbol{x} \mid \boldsymbol{z} = k] + \mathbb{E}[\boldsymbol{x} \mid \boldsymbol{z} = k] \, \mathbb{E}[\boldsymbol{x} \mid \boldsymbol{z} = k]^T\right) \\
&= \sum_{k=1}^{K} \boldsymbol{\pi}_k \left(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T\right)
\end{aligned}$$

and thus

$$\mathrm{Cov}[\boldsymbol{x}] = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] - \mathbb{E}[\boldsymbol{x}] \, \mathbb{E}[\boldsymbol{x}]^T = \sum_{k=1}^{K} \boldsymbol{\pi}_k (\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T) - \sum_{k=1}^{K} \sum_{j=1}^{K} \boldsymbol{\pi}_k \boldsymbol{\pi}_j \boldsymbol{\mu}_k \boldsymbol{\mu}_j^T$$

Problem 4: Consider two random variables $\boldsymbol{x} \in \mathbb{R}^D$ and $\boldsymbol{y} \in \mathbb{R}^D$ distributed according to two different Gaussian mixture models with $\boldsymbol{\theta}^x = \{\boldsymbol{\pi}^x, \boldsymbol{\mu}^x, \boldsymbol{\Sigma}^x\}$ and $\boldsymbol{\theta}^y = \{\boldsymbol{\pi}^y, \boldsymbol{\mu}^y, \boldsymbol{\Sigma}^y\}$, i.e.

$$\begin{aligned}
\mathrm{p}(\boldsymbol{x} \mid \boldsymbol{\theta}^x) &= \sum_{k=1}^{K_x} \boldsymbol{\pi}_k^x \, \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k^x, \boldsymbol{\Sigma}_k^x), \\
\mathrm{p}(\boldsymbol{y} \mid \boldsymbol{\theta}^y) &= \sum_{l=1}^{K_y} \boldsymbol{\pi}_l^y \, \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{\mu}_l^y, \boldsymbol{\Sigma}_l^y),
\end{aligned}$$

and the random variable $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y}$.

a) Describe a generative process (process of drawing samples) for $\boldsymbol{z}$.

b) Explain in a few sentences why $\mathrm{p}(\boldsymbol{z} \mid \boldsymbol{\theta}^x, \boldsymbol{\theta}^y)$ is again a mixture of Gaussians.

c) State the probability density function $\mathrm{p}(\boldsymbol{z} \mid \boldsymbol{\theta}^x, \boldsymbol{\theta}^y)$ of $\boldsymbol{z}$.

a) Draw a sample $\boldsymbol{x}$ from $\mathrm{p}(\boldsymbol{x} \mid \boldsymbol{\theta}^x)$ with the usual GMM sampling method and the same for $\boldsymbol{y}$ from $\mathrm{p}(\boldsymbol{y} \mid \boldsymbol{\theta}^y)$. Now add them together to get $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y}$.

b) Let $\boldsymbol{x}$ be drawn from the component k of $\mathrm{p}(\boldsymbol{x} \mid \boldsymbol{\theta}^x)$ and $\boldsymbol{y}$ be drawn from the component l of $\mathrm{p}(\boldsymbol{y} \mid \boldsymbol{\theta}^y)$. Then $\boldsymbol{z}$ is the sum of two normally distributed random variables $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_k^x, \boldsymbol{\Sigma}_k^x)$ and $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}_l^y, \boldsymbol{\Sigma}_l^y)$. Therefore, it also follows a normal distribution $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}_k^x + \boldsymbol{\mu}_l^y, \boldsymbol{\Sigma}_k^x + \boldsymbol{\Sigma}_l^y)$. There are $K_x \cdot K_y$ such possible (k, l) combinations, each having probability $\boldsymbol{\pi}_k^x \boldsymbol{\pi}_l^y$ respectively.

That is, $\mathrm{p}(\boldsymbol{z} \mid \boldsymbol{\theta}^x, \boldsymbol{\theta}^y)$ is a mixture of $K_x K_y$ Gaussians.

c) It follows from the argument in b) that the probability density function of $\boldsymbol{z}$ is

$$\mathrm{p}(\boldsymbol{z} \mid \boldsymbol{\theta}^x, \boldsymbol{\theta}^y) = \sum_{k=1}^{K_x} \sum_{l=1}^{K_y} \boldsymbol{\pi}_k^x \boldsymbol{\pi}_l^y \, \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{\mu}_k^x + \boldsymbol{\mu}_l^y, \boldsymbol{\Sigma}_k^x + \boldsymbol{\Sigma}_l^y).$$

Problem 5: Download the notebook `exercise_12_clustering.ipynb` 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.

In-class Exercises

K-Medians

Problem 6: Consider a modified version of the K-means objective, where we use L_1 distance instead.

$$\mathcal{J}(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\mu}) = \sum_{i=1}^{N} \sum_{k=1}^{K} \boldsymbol{z}_{ik} ||\boldsymbol{x}_i - \boldsymbol{\mu}_k||_1$$

This variation of the algorithm is called K*-medians*. Derive the Lloyd's algorithm for this model.

1. Updating the cluster assignments $\boldsymbol{z}_{ik}$ is the same as for the K-means algorithm:

$$\boldsymbol{z}_{ik}^{new} = \begin{cases} 1 & \text{if } \ k = \arg\min_j ||\boldsymbol{x}_i - \boldsymbol{\mu}_j||_1 \\ 0 & \text{else.} \end{cases}$$

2. The updates for $\boldsymbol{\mu}_k$'s should solve

$$\boldsymbol{\mu}_k^{new} = \arg\min_{\boldsymbol{\mu}_k} \sum_{i=1}^{N} \boldsymbol{z}_{ik} ||\boldsymbol{x}_i - \boldsymbol{\mu}_k||_1$$

The objective for each single centroid $\boldsymbol{\mu}_k$ can be rewritten as

$$\begin{aligned} \mathcal{J}(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\mu}_k) &= \sum_{i=1}^{N} \boldsymbol{z}_{ik} ||\boldsymbol{x}_i - \boldsymbol{\mu}_k||_1 \\ &= \sum_{i=1}^{N} \boldsymbol{z}_{ik} \sum_{d=1}^{D} |\boldsymbol{x}_{id} - \boldsymbol{\mu}_{kd}| \end{aligned}$$

Clearly, this is a convex function of $\boldsymbol{\mu}_k$, as it is a sum of piecewise linear functions. We can actually solve for each $\boldsymbol{\mu}_{kd}$ separately, as they do not interact in the objective, by finding the roots of the derivatives.

Observe, that

$$\frac{\partial}{\partial \boldsymbol{\mu}_{kd}} |\boldsymbol{x}_{id} - \boldsymbol{\mu}_{kd}| = \begin{cases} 1 & \text{if } \boldsymbol{\mu}_{kd} > \boldsymbol{x}_{id} \\ -1 & \text{if } \boldsymbol{\mu}_{kd} < \boldsymbol{x}_{id} \\ 0 & \text{if } \boldsymbol{\mu}_{kd} = \boldsymbol{x}_{id}. \end{cases}$$

(Note: actually the absolute value function is not differentiable at 0, so the derivative is undefined. A rigorous treatment of this problem would require us to use subgradients (see `https://web.stanford.edu/class/ee364b/lectures/subgradients_notes.pdf`), but just "pretending" that the gradient is 0 suffices for our purpose.)

Hence, the derivative of the entire objective is

$$\frac{\partial}{\partial \boldsymbol{\mu}_{kd}} \mathcal{J}(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\mu}) = \sum_{i=1}^{N} \boldsymbol{z}_{ik} |\boldsymbol{x}_{id} - \boldsymbol{\mu}_{kd}|$$
$$= \sum_{i=1}^{N} \boldsymbol{z}_{ik} \mathbb{I}[\boldsymbol{x}_{id} < \boldsymbol{\mu}_{kd}] - \sum_{i=1}^{N} \boldsymbol{z}_{ik} \mathbb{I}[\boldsymbol{x}_{id} > \boldsymbol{\mu}_{kd}] \stackrel{!}{=} 0$$

The first sum represents "number of points $\boldsymbol{x}_i$ assigned to class k, such that $\boldsymbol{x}_{id} < \boldsymbol{\mu}_{kd}$". Each of these sums represents the number of points in class k, that are located to the left (right) of the given value of $\boldsymbol{\mu}_{kd}$. Because we want to set the gradient to zero, we are looking for such a $\boldsymbol{\mu}_{kd}$, that along the axis d exactly $N_k/2$ points are to left of it, and another $N_k/2$ points are to the right (where $N_k = \sum_{i=1}^{N} \boldsymbol{z}_{ik}$). This is exactly the definition of a *median*.

Therefore, the optimal update is given as

$$\boldsymbol{\mu}_{kd} = \text{median}\left\{\boldsymbol{x}_{id} \text{ such that } \boldsymbol{z}_{ik} = 1\right\}$$

Gaussian Mixture Model

Problem 7: Derive the E-step update for the Gaussian mixture model.

In the E-step we have to evaluate the posterior distribution over the latent variables given the current parameters, i.e. $\gamma_t(\boldsymbol{Z})$. Because GMMs assume that the latent variables are independent, $\gamma_t(\boldsymbol{Z}) = \prod_{i=1}^{N} \gamma_t(\boldsymbol{z}_i)$ and it is enough to derive the E-step for a single data point. The update rule follows directly from Bayes' theorem.

$$\gamma_t(\boldsymbol{z}_i = k) = \mathrm{p}(\boldsymbol{z}_i = k \mid \boldsymbol{x}_i, \boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)})$$
$$= \frac{\mathrm{p}(\boldsymbol{x}_i \mid \boldsymbol{z}_i = k, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)})\ \mathrm{p}(\boldsymbol{z}_i = k \mid \boldsymbol{\pi}^{(t)})}{\mathrm{p}(\boldsymbol{x}_i \mid \boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)})}$$
$$= \frac{\mathrm{p}(\boldsymbol{x}_i \mid \boldsymbol{z}_i = k, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)})\ \mathrm{p}(\boldsymbol{z}_i = k \mid \boldsymbol{\pi}^{(t)})}{\sum_{j=1}^{K} \mathrm{p}(\boldsymbol{x}_i \mid \boldsymbol{z}_i = j, \boldsymbol{\pi}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)})\ \mathrm{p}(\boldsymbol{z}_i = j \mid \boldsymbol{\pi}^{(t)})}$$
$$= \frac{\boldsymbol{\pi}_k^{(t)} \mathcal{N}\left(\boldsymbol{x}_i \mid \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)}\right)}{\sum_{j=1}^{K} \boldsymbol{\pi}_j^{(t)} \mathcal{N}\left(\boldsymbol{x}_i \mid \boldsymbol{\mu}_j^{(t)}, \boldsymbol{\Sigma}_j^{(t)}\right)}$$

Problem 8: Derive the M-step update for the Gaussian mixture model.

In the M-step we maximize $\mathcal{L} = \mathbb{E}_{\boldsymbol{Z} \sim \gamma_t(\boldsymbol{Z})} \left[\log \mathrm{p}(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})\right]$ with respect to $\boldsymbol{\pi}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. When we plug in the definition of the expected value and expand, we get

$$
\begin{aligned}
\mathcal{L} &= \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_t(\boldsymbol{z}_i = k)\, \log \mathrm{p}(\boldsymbol{x}_i, \boldsymbol{z}_i = k \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_t(\boldsymbol{z}_i = k)\, \log \mathrm{p}(\boldsymbol{x}_i \mid \boldsymbol{z}_i = k, \boldsymbol{\mu}, \boldsymbol{\Sigma})\ \mathrm{p}(\boldsymbol{z}_i = k \mid \boldsymbol{\pi}) \\
&= \underbrace{\sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_t(\boldsymbol{z}_i = k)\, \log \mathrm{p}(\boldsymbol{x}_i \mid \boldsymbol{z}_i = k, \boldsymbol{\mu}, \boldsymbol{\Sigma})}_{\mathcal{L}_{\boldsymbol{x}}} + \underbrace{\sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_t(\boldsymbol{z}_i = k)\, \log \mathrm{p}(\boldsymbol{z}_i = k \mid \boldsymbol{\pi})}_{\mathcal{L}_{\boldsymbol{z}}}
\end{aligned}
$$

where $\mathcal{L}_{\boldsymbol{z}}$ only depends on $\boldsymbol{\pi}$ and $\mathcal{L}_{\boldsymbol{x}}$ only depends on $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. To find the optimal $\boldsymbol{\pi}$, we need to maximize $\mathcal{L}_{\boldsymbol{z}}$ with respect to $\boldsymbol{\pi}$. Since $\boldsymbol{\pi}$ has several constraints placed on it, we will have to solve the following convex optimization problem.

$$
\begin{aligned}
\text{maximize} \quad & \mathcal{L}_{\boldsymbol{z}} \\
\text{subject to} \quad & \sum_{k=1}^{K} \boldsymbol{\pi}_k - 1 = 0
\end{aligned}
$$

Before we formulate the Lagrangian, we simplify $\mathcal{L}_{\boldsymbol{z}}$ as

$$
\mathcal{L}_{\boldsymbol{z}} = \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_t(\boldsymbol{z}_i = k)\, \log \mathrm{p}(\boldsymbol{z}_i = k \mid \boldsymbol{\pi}) = \sum_{k=1}^{K} N_k \log \boldsymbol{\pi}_k
$$

where $N_k = \sum_{i=1}^{N} \gamma_t(\boldsymbol{z}_i = k)$ is the size of the k-th cluster. The Lagrangian is given by

$$
f(\boldsymbol{\pi}, \lambda) = \sum_{k=1}^{K} N_k \log \boldsymbol{\pi}_k + \lambda \left(1 - \sum_{k=1}^{K} \boldsymbol{\pi}_k\right)
$$

and it has its maximum in $\boldsymbol{\pi}$ at

$$
\frac{\partial f}{\partial \boldsymbol{\pi}_k} = \frac{N_k}{\boldsymbol{\pi}_k} - \lambda \stackrel{!}{=} 0 \Leftrightarrow \boldsymbol{\pi}_k = \frac{N_k}{\lambda}
$$

because f is concave as a function of $\boldsymbol{\pi}$. This gives us the dual function as

$$
g(\lambda) = \max_{\boldsymbol{\pi}} f(\boldsymbol{\pi}, \lambda) = f\left(\left(\frac{N_1}{\lambda}, \ldots, \frac{N_K}{\lambda}\right), \lambda\right) = \sum_{k=1}^{K} N_k \log \frac{N_k}{\lambda} + \lambda - N.
$$

When f is concave, the dual is convex and we find the minimum of g at

$$
\frac{\partial g}{\partial \lambda} = \sum_{k=1}^{K} N_k \frac{\lambda}{N_k} \left(-\frac{N_k}{\lambda^2}\right) + 1 = 1 - \frac{N}{\lambda} \stackrel{!}{=} 0 \Leftrightarrow \lambda = N.
$$

This means that the M-step for $\boldsymbol{\pi}$ is $\boldsymbol{\pi}_k^{(t+1)} = \frac{N_k}{N}$.

To find the M-step rules for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, we need to examine $\mathcal{L}_{\boldsymbol{x}}$.

$$
\begin{aligned}
\mathcal{L}_{\boldsymbol{x}} &= \sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_t(\boldsymbol{z}_i = k)\, \log\left(\mathcal{N}\left(\boldsymbol{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)\right) \\
&= -\frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K} \gamma_t(\boldsymbol{z}_i = k)\left((\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k) + D\log(2\pi) + \log\det\boldsymbol{\Sigma}_k\right).
\end{aligned}
$$

where D is the feature dimension. We can take the derivative with respect to $\boldsymbol{\mu}_k$

$$
\frac{\partial \mathcal{L}_{\boldsymbol{x}}}{\partial \boldsymbol{\mu}_k} = -\frac{1}{2}\sum_{i=1}^{N} \gamma_t(\boldsymbol{z}_i = k)\left((-1)\cdot\left(\boldsymbol{\Sigma}_k^{-1} + \boldsymbol{\Sigma}_k^{-T}\right)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right) = \sum_{i=1}^{N} \gamma_t(\boldsymbol{z}_i = k)\left(\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right)
$$

and then find its root

$$
\frac{\partial \mathcal{L}_{\boldsymbol{x}}}{\partial \boldsymbol{\mu}_k} = 0 \Leftrightarrow \sum_{i=1}^{N} \gamma_t(\boldsymbol{z}_i = k)\boldsymbol{\Sigma}_k^{-1}\boldsymbol{x}_i = N_k \boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k \Leftrightarrow \boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{i=1}^{N} \gamma_t(\boldsymbol{z}_i = k)\boldsymbol{x}_i
$$

which gives us the update rule

$$
\boldsymbol{\mu}_k^{(t+1)} = \frac{1}{N_k}\sum_{i=1}^{N} \gamma_t(\boldsymbol{z}_i = k)\boldsymbol{x}_i.
$$

It remains to find the M-step for $\boldsymbol{\Sigma}$. Again we proceed by taking the derivative with respect to $\boldsymbol{\Sigma}_k$

$$
\begin{aligned}
\frac{\partial \mathcal{L}_{\boldsymbol{x}}}{\partial \boldsymbol{\Sigma}_k} &= -\frac{1}{2}\sum_{i=1}^{N} \gamma_t(\boldsymbol{z}_i = k)\left[-\boldsymbol{\Sigma}_k^{-T}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-T} + \boldsymbol{\Sigma}_k^{-T}\right] \\
&= -\frac{1}{2}\left(N_k I_D - \sum_{i=1}^{N} \gamma_t(\boldsymbol{z}_i = k)\left[\boldsymbol{\Sigma}_k^{-T}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T\right]\right)\boldsymbol{\Sigma}_k^{-T}
\end{aligned}
$$

where I_D is the D-dimensional identity matrix. We finish by finding its root

$$
\frac{\partial \mathcal{L}_{\boldsymbol{x}}}{\partial \boldsymbol{\Sigma}_k} = 0 \Leftrightarrow N_k I_D = \boldsymbol{\Sigma}_k^{-T}\sum_{i=1}^{N} \gamma_t(\boldsymbol{z}_i = k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T
$$

which produces the final update rule

$$
\boldsymbol{\Sigma}_k^{(t+1)} = \frac{1}{N_k}\sum_{i=1}^{N} \gamma_t(\boldsymbol{z}_i = k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T.
$$

In this exercise we have used the following matrix calculus rules which you can look up in the matrix cookbook.

$$
\frac{\partial \boldsymbol{a}^T\boldsymbol{X}\boldsymbol{a}}{\partial \boldsymbol{a}} = \left(\boldsymbol{X} + \boldsymbol{X}^T\right)\boldsymbol{a}^T \qquad \frac{\partial \boldsymbol{a}^T\boldsymbol{X}^{-1}\boldsymbol{b}}{\partial \boldsymbol{X}} = -\boldsymbol{X}^{-T}\boldsymbol{b}\boldsymbol{a}^T\boldsymbol{X}^{-T} \qquad \frac{\partial \log|\det\boldsymbol{X}|}{\partial \boldsymbol{X}} = \boldsymbol{X}^{-T}
$$

Expectation Maximization Algorithm

Problem 9: Consider a mixture model where the components are given by independent Bernoulli variables. This is useful when modelling, e.g., binary images, where each of the D dimensions of the image $\boldsymbol{x}$ corresponds to a different pixel that is either black or white. More formally, we have

$$\mathrm{p}(\boldsymbol{x} \mid \boldsymbol{z} = k) = \prod_{d=1}^{D} \boldsymbol{\theta}_{kd}^{x_d} (1 - \boldsymbol{\theta}_{kd})^{1-x_d}.$$

That is, for a given mixture index $\boldsymbol{z} = k$, we have a product of independent Bernoullis, where $\boldsymbol{\theta}_{kd}$ denotes the Bernoulli parameter for component k at pixel d.

Derive the EM algorithm for the parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{kd} \mid k = 1, \ldots, K, d = 1, \ldots, D\}$ of a mixture of Bernoullis.

Assume here for simplicity, that the distribution of components $\mathrm{p}(\boldsymbol{z})$ is uniform: $\mathrm{p}(\boldsymbol{z}) = \prod_{k=1}^{K} \boldsymbol{\pi}_k^{z_k} = \prod_{k=1}^{K} \left(\frac{1}{K}\right)^{z_k}$.

Due to the uniform prior on $\boldsymbol{z}$, the $\mathrm{p}(\boldsymbol{z})$ cancel and the responsibilities compute as

$$\gamma_t(\boldsymbol{z}_i = k) = \frac{\mathrm{p}(\boldsymbol{x}_i \mid \boldsymbol{z} = k, \boldsymbol{\theta}) \cdot \mathrm{p}(\boldsymbol{z} = k)}{\sum_{l=1}^{K} \mathrm{p}(\boldsymbol{x}_i \mid \boldsymbol{z} = l, \boldsymbol{\theta}) \cdot \mathrm{p}(\boldsymbol{z} = l)} = \frac{\mathrm{p}(\boldsymbol{x}_i \mid \boldsymbol{z} = k, \boldsymbol{\theta})}{\sum_{l=1}^{K} \mathrm{p}(\boldsymbol{x}_i \mid \boldsymbol{z} = l, \boldsymbol{\theta})}$$

which constitues the E-step.

It remains to derive the M-step. Similiar to mixture of Gaussians:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{z} \sim \gamma_t(\boldsymbol{z})}[\log \mathrm{p}(\boldsymbol{X}, \boldsymbol{z} \mid \boldsymbol{\theta}^{(t)})] &= \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_t(\boldsymbol{z}_i = k) \log \left(\frac{1}{K} \prod_{d=1}^{D} \boldsymbol{\theta}_{kd}^{\boldsymbol{x}_{id}} (1 - \boldsymbol{\theta}_{kd})^{1-\boldsymbol{x}_{id}} \right) \\
&= C + \sum_{i=1}^{N} \underbrace{\sum_{k=1}^{K} \gamma_t(\boldsymbol{z}_i = k) \sum_{d=1}^{D} (\boldsymbol{x}_{id} \log \boldsymbol{\theta}_{kd} + (1 - \boldsymbol{x}_{id}) \log(1 - \boldsymbol{\theta}_{kd}))}_{=:\mathcal{L}_i}
\end{aligned}$$

The constant C collects all terms independent of $\boldsymbol{\theta}$ and hence irrelevant for further optimization.

We now need to take derivatives with respect to $\boldsymbol{\theta}$.

$$\begin{aligned}
\frac{\partial \mathcal{L}_i}{\partial \boldsymbol{\theta}_{k',d'}} &= \sum_{k=1}^{K} \gamma_t(\boldsymbol{z}_i = k) \sum_{d=1}^{D} \left(\boldsymbol{x}_{id} \frac{\partial \log \boldsymbol{\theta}_{kd}}{\partial \boldsymbol{\theta}_{k',d'}} + (1 - \boldsymbol{x}_{id}) \frac{\partial \log(1 - \boldsymbol{\theta}_{kd})}{\partial \boldsymbol{\theta}_{k',d'}} \right) \\
&= \gamma_t(\boldsymbol{z}_i = k) \left(\frac{\boldsymbol{x}_{id}}{\boldsymbol{\theta}_{k',d'}} - \frac{1 - \boldsymbol{x}_{id}}{1 - \boldsymbol{\theta}_{k',d'}} \right)
\end{aligned}$$

We observe that the $\boldsymbol{\theta}_{kd}$ do not interact, so their optimal values are independent from each other and we can handle them individually.

$$\frac{\partial \, \mathbb{E}_{\boldsymbol{z} \sim \gamma_t(\boldsymbol{z})}[\log \mathrm{p}(\boldsymbol{X}, \boldsymbol{z} \mid \boldsymbol{\theta})]}{\partial \boldsymbol{\theta}_{kd}} = \sum_{i=1}^{N} \frac{\partial \mathcal{L}_i}{\partial \boldsymbol{\theta}_{kd}} = \sum_{i=1}^{N} \gamma_t(\boldsymbol{z}_i = k) \left(\frac{\boldsymbol{x}_{id}}{\boldsymbol{\theta}_{kd}} - \frac{1 - \boldsymbol{x}_{id}}{1 - \boldsymbol{\theta}_{kd}} \right)$$

By finding the roots with respect to $\boldsymbol{\theta}_{kd}$, we obtain the optimal update in a similar fashion as in the standard Bernoulli MLE:

$$\boldsymbol{\theta}_{kd} = \frac{\sum_{i=1}^{N} \gamma_t(\boldsymbol{z}_i = k)\, \boldsymbol{x}_{id}}{\sum_{i=1}^{N} \gamma_t(\boldsymbol{z}_i = k)}$$