Exercises
Computational Intelligence Lab
SS 2019

Machine Learning Institute
Dept. of Computer Science, ETH Zürich
Prof. Dr. Thomas Hofmann
Web http://cil.inf.ethz.ch/

Series 7, April 4-5, 2019 (K-means and Mixture Models)

1 Probability Refresher

Problem 1 (Conditional Probability):

A couple has two children, each of them being independently a boy or a girl with 50% probability. Compute the probabilities of the following events.

1. At least one of the children is a girl.
2. Both children are girls.
3. Both children are girls given that the first born is a girl.
4. Both children are girls given that one of them is a girl.
5. Both children are girls given that one of them is a girl named Cassiopeia.
   Note: Cassiopeia is an extremely rare name with a frequency of less than 1 in 1,000,000.

Problem 2 (Bayes' Rule):

There is an uncommon disease that has infected 1% of the human population. Assume that we have a test for this disease that is positive on an infected person with probability 99% and negative on a healthy person also with probability 99%.

If my test comes out positive, what is the probability that I am infected?

2 The K-means algorithm

Problem 1 (Theory):

In this exercise, you will elaborate on some of the formal results connecting K-means theory and matrix factorization.

1. Show that the K-means algorithm always converges. In particular, consider the following cost function

$$J := \sum_{n=1}^{N} \sum_{k=1}^{K} z_{k,n} \|\mathbf{x}_n - \mathbf{u}_k\|_2^2,$$

   and show that steps 2 and 3 of the K-means algorithm from the lecture minimize this cost function for $\mathbf{z}_n$ and $\mathbf{u}_k$, respectively.

2. Show that the K-means algorithm solves a matrix factorization problem, in the sense that

$$\arg\min_{\boldsymbol{Z}} \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{Z}\|_F^2 = \arg\min_{\boldsymbol{Z}} \sum_{n=1}^{N} \sum_{k=1}^{K} z_{k,n} \|\boldsymbol{x}_n - \boldsymbol{u}_k\|_2^2,$$

when $\boldsymbol{Z} \in \mathbb{R}^{K\times N}$ is additionally restricted to be an assignment matrix (having exactly a single non-zero entry of 1 in each column). The other matrices are given as follows:

- data matrix $\boldsymbol{X} := [\boldsymbol{x_1} \cdots \boldsymbol{x_N}] \in \mathbb{R}^{D\times N}$,
- centroid matrix $\boldsymbol{U} := [\boldsymbol{u_1} \cdots \boldsymbol{u_K}] \in \mathbb{R}^{D\times K}$,
- assignment matrix $\boldsymbol{Z} := [\boldsymbol{z_1} \cdots \boldsymbol{z_N}] \in \mathbb{R}^{K\times N}$.

3. Show intuitively that K-means always terminates, i.e. converges in a finite number of steps.

Problem 2 (Practical exercise):

1. You are given a dataset of points $\{-2, 9, 1, -3, 6, 5, 4, 8\}$ in $\mathbb{R}$. Cluster this dataset using the K-means algorithm with $K = 2$. Assume that the two clusters are initialized as follows: C_1 contains $\{9, -2, 5, 8\}$ and C_2 contains $\{6, 1, -3, 4\}$. Describe all steps carefully and solve until convergence.

Problem 3 (Implementation):

In this exercise, you will implement a vector quantization scheme for image color compression (one of the most basic forms of image compression where each pixel is compressed independently).

1. Load the RGB image `eth.jpg` (Figure 1), which consists of 8 bits per channel. What is its uncompressed size (considering only pixels and not metadata)?

2. Implement K-means and run it on the image, treating each pixel as a 3D vector (one dimension per color channel R, G, and B). Initialize the clusters as randomly chosen points that are part the image, and try $k = \{4,\ 16,\ 64\}$. What size reduction can you achieve for each k?
Hint: while representing $\mathbf{Z}$ *in matrix form comes handy for theoretical analysis, in an actual implementation you do not need to store it explicitly. Storing an index for each data point is enough.*

3. As you increase k, you will probably notice that some clusters become empty. Why does this happen? How do you tackle this issue?

4. In data compression, after applying vector quantization, it is common to compress the assignment matrix using a coding scheme (e.g. Huffman trees). Assuming that all pixels are compressed independently of each other, what is the lower bound of bits per pixel that can be achieved?
Hint: compute a probability distribution over cluster assignments and estimate its entropy.

Figure 1: `eth.jpg`

3 Mixture Models

Problem 1 (EM Algorithm):

In this exercise, we derive the two steps of the Expectation Maximization algorithm. Assume we are given a data set $\boldsymbol{X}$ consisting of N i.i.d observations $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ and our goal is to cluster these observations using a mixture of K Gaussian distributions.

1. Write down the expression for the log-likelihood of the mixture model given data $\boldsymbol{X}$ (i.e., $\ln p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$).
2. Show that a lower bound of the log-likelihood is given by:

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma_{nk}\left[\log \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k) + \log \pi_k - \log \gamma_{nk}\right]$$

.

E-step: the goal is to maximize the lower bound with respect to the posterior probabilities of the latent variables.

3. Show that maximizing the bound w.r.t γ_{nk} held the following result:

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(x_n \mid \boldsymbol{\mu_k}, \Sigma_k)}{\pi_q \sum_{q=1}^{K} \mathcal{N}(x_n \mid \boldsymbol{\mu_q}, \Sigma_q)}$$

.

4. Let's introduce now the cluster assignment for each data point:

$$z_{nk} = \begin{cases} 1 & \text{if point n comes from k-th Gaussian component} \\ 0 & \text{ow} \end{cases}$$

How is γ_{nk} related to z_{nk} ?

M-step: the goal is to maximize the lower bound w.r.t. the parameters $\pi_k, \boldsymbol{\mu}, \Sigma$, assuming a current guess of γ_{nk}.
Note: this is equal to maximize the expected complete log-likelihood:

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\gamma_{nk}\Big(\log \pi_k + \log\ \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \Sigma_k)\Big)$$

since γ_{nk} is treated as constant.

5. Show that the optimal mixing coefficients, π_k, are given by:

$$\pi_k = \frac{1}{N}\sum_{n=1}^{N}\gamma_{nk}$$

Hint: remember to include the constraint $\sum_{k=1}^{K}\pi_k = 1$

6. Show that the optimal choice with respect to the *mean vectors* $\boldsymbol{\mu}_k$ for all $k = 1, \dots, K$, is given as

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N}\gamma_{nk}\mathbf{x}_n}{\sum_{n=1}^{N}\gamma_{nk}}$$

Hint: for a symmetric matrix $A \in \mathrm{R}^{n\times n}$ *and a vector* $\mathbf{x} \in \mathrm{R}^n$ *it holds that* $\frac{\partial}{\partial \mathbf{x}}\mathbf{x}^T A\mathbf{x} = 2A\mathbf{x}$.

Problem 2 (Singularities in Gaussian Mixture Models):

In this exercise we consider the problem of singularities when maximizing the likelihood of a Gaussian mixture model. Assume we are given a data set $\boldsymbol{X}$ consisting of N i.i.d observations $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ and our goal is to cluster these observations using a mixture of K Gaussian distributions. Now, consider a Gaussian mixture model whose components have covariance matrices given by $\boldsymbol{\Sigma}_k = \sigma_k^2\boldsymbol{I}$, where $\boldsymbol{I}$ is the unit matrix and suppose that one of the components, say the j-th, has a mean parameter $\boldsymbol{\mu}_j$ that is equal to one of the data points, i.e. $\boldsymbol{\mu}_j = \boldsymbol{x}_n$ for some n.

1. Write down the expression for the log-likelihood of the mixture model given $\boldsymbol{x}_n$ (i.e., $\ln p(\boldsymbol{x}_n|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$).
2. Compute the likelihood of the j-th mixture component given $\boldsymbol{x}_n$ (i.e. $\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$).

   Hint: The multivariate Gaussian probability density function is defined as

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) := \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right).$$

3. What happens to the likelihood of the previous question as $\sigma_j \to 0$? How does this affect the log-likelihood of the mixture model given in question 1?
4. Can the above situation occur when the mixture model consists of a single Gaussian distribution, i.e. $K = 1$?
5. Can you propose a heuristic to avoid such situations?

Problem 3 (Identifiability):

In this exercise we consider the problem of *identifiability* of maximum likelihood solutions of mixture models.

1. Suppose that we have solved a mixture of K Gaussians problem and have obtained the values of the parameters. How many equivalent solutions are there?
2. This problem is known as *identifiability*. Explain why this is not a problem in the context of data clustering.