Elements of Machine Learning, WS 2022/2023

Jilles Vreeken and Aleksandar Bojchevski

Exercise Sheet #5: *Unsupervised Learning*

Deadline: Thursday, January 19, 2023, 16:00

Before solving the exercises, read the instructions on the course website.

- For each theoretical problem, submit a single `pdf` file that contains your answer to the respective problem. This file may be a scan of your (legible) handwriting.
- For each practical problem, submit a single `zip` file that contains
  - the completed jupyter notebook (`.ipynb`) file,
  - any necessary files required to reproduce your results, and
  - a `pdf` report generated from the jupyter notebook that shows all your results.
- For the bonus question, submit a single `zip file` that contains
  - a `pdf` file that includes your answers to the theoretical part,
  - the completed jupyter notebook (`.ipynb`) file for the practical component,
  - any necessary files required to reproduce your results, and
  - a `pdf` report generated from the jupyter notebook that shows your results.
- Every team member has to submit a signed Code of Conduct.

Problem 1 (T, 5 Points). **Principal Component Analysis**

The first principal component is the direction of maximum variance in the data. Show that this first principal component also minimizes the residual sum of squares, which is here the Euclidean squared distance between the projected data points and the original data points.

Problem 2 (T, 4+1 Points). **Different flavors of embeddings**

In the course, you learned about different techniques for dimensionality reductions that embed high dimensional data into a low dimensional space.

1. For the three methods *PCA*, *MDS*, and *tSNE* explain in at most 50 words for each method, what the respective method does. In your description, put your focus on describing what high dimensional properties are captured and aimed to be preserved in the low dimensional space.
2. Name one disadvantage for each of the embeddings.

Problem 3 (T, 4 + 2 + 4 Points).
Suppose that for a particular data set, we perform hierarchical clustering using single linkage and using complete linkage. We obtain two dendrograms.

1. (Exercise 12.6.4 in ISLR)

    - At a certain point on the single linkage dendrogram, the clusters $\{1, 2, 3\}$ and $\{4, 5\}$ fuse. On the complete linkage dendrogram, the clusters $\{1, 2, 3\}$ and $\{4, 5\}$ also fuse at a certain point.Which fusion will occur higher on the tree, or will they fuse at the same height, or is there not enough information to tell?
    - At a certain point on the single linkage dendrogram, the clusters $\{5\}$ and $\{6\}$ fuse. On the complete linkage dendrogram, the clusters $\{5\}$ and $\{6\}$ also fuse at a certain point. Which fusion will occur higher on the tree, or will they fuse at the same height, or is there not enough information to tell?

    Explain your reasoning for both the cases.

2. Name and explain one other choice of dissimilarity measure for hierarchical clustering apart from the Euclidean distance metric. Give an example where your stated dissimilarity measure would be a better than the Euclidean distance metric.

3. What are some practical considerations that one needs to take into account when applying clustering on the data? Describe a total of four practical considerations with at least one consideration for hierarchical and at least one for K-Means clustering.

Problem 4 (T,8+2 Points).

1. (K-Means clustering, Exercise 12.3.1 in ISLR)
   Please read section 12.4.1 from the ISLR book before attempting this question.

   - Show that equation 12.18 (given below) on ISLR page 519 holds :

     $$\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 = 2\sum_{i\in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2,$$

     where $|C_k|$ denotes the number of observations in cluster C_k, and $\bar{x}_{kj}$ the mean for feature j in cluster C_k. Argue on the basis of this identity, that the K-means clustering algorithm decreases the objective

     $$\underset{C_1,\ldots,C_K}{\text{minimize}}\left\{\sum_{k=1}^{K}\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2\right\}$$

     at each iteration.

   - Explain in your own words (a) what equality you are proving and (b) what you can conclude from it.

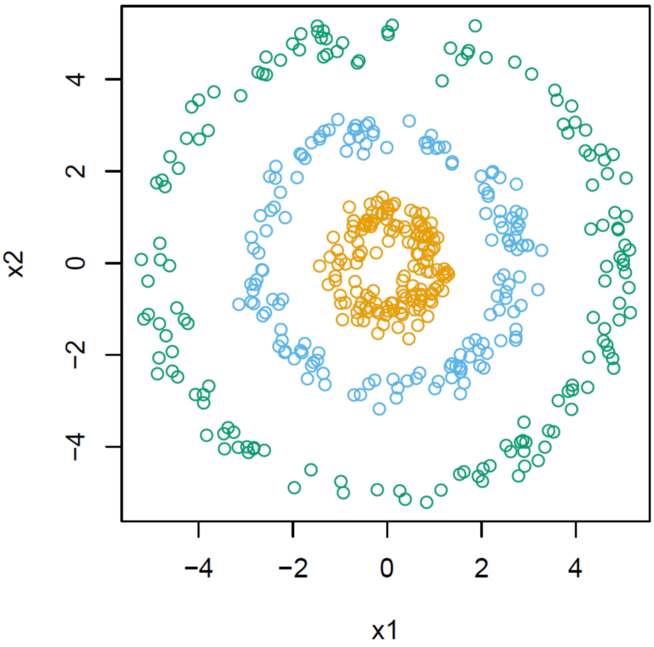

Figure 1: 2 dimensional data for task 3 (from ESL Fig. 14.29).

2. Consider the data plot shown in Figure 1 where each colour denotes one cluster.

   - Can we use k-means clustering to correctly cluster the data points? Why or why not?
   - If you should use hierarchical clustering for this data, which linkage (complete, average, single or centroid) would do best and why?

Problem 5 (P, 5+5 Points). **Programming Problem**
In this problem, you will perform and analyze different types of clustering on the data.

1. PCA and K-Means (Partially taken from Ex. 12.6.10 ISLR Book).
   (a) Download and load the file `data1.csv` into your environment.
   (b) Perform PCA on the 60 observations and plot the first two principal component score vectors. Use a different color to indicate the observations in each of the three classes.
   (c) Perform K-means clustering of the observations with K = 3. How well do the clusters that you obtained in K-means clustering compare to the true class labels?
   (d) Perform K-means clustering with K = 2 and K = 4. Describe your results for each.
   (e) Now perform K-means clustering with K = 3 on the first two principal component score vectors, rather than on the raw data. That is, perform K-means clustering on the 60×2 matrix of which the first column is the first principal component score vector, and the second column is the second principal component score vector. Comment on the results.
2. Hierarchical clustering
   (a) Load the `data2.csv` file provided to you. Visualize the data and familiarize yourself with the data points. How many clusters can you see in the data?
   (b) Calculate the inter-observational distance between each of the data points using the Euclidean distance metric.
   (c) Perform Hierarchical clustering using average, complete and single linkage on the given data. Plot the resulting dendograms for each of the clustering.
   (d) Determine the cluster labels for each observation associated with a given cut of the dendogram. Use your proposed number of clusters from the first sub-part, as the number of clusters.

Note: Useful Python packages for this assignment are `sklearn.decomposition.PCA`, `sklearn.cluster.KMeans`, `scipy.cluster.hierarchy.dendrogram` and `scipy.cluster.hierarchy.linkage`.

Problem 6 (Bonus). **Clustering with Gaussian Mixture Models**
In this problem you will use a Gaussian Mixture Model (GMM) to perform clustering instead of K means. GMM is a parametric method that assumes data is drawn from multiple Gaussian distributions. Each Gaussian corresponds to a cluster with the mean of the Gaussian representing the cluster center and the Covariance captures the spread of the data in the cluster. The K means algortihm is a special case of the GMM. GMMs enable soft cluster boundaries i.e. each datapoint belongs to a cluster k with a certain probability.
Each data point i belongs to cluster k with probability γ_{ik} where $\sum_{k=1}^{K} \gamma_{ik} = 1$. γ_{ik} are also called the responsibilities of each data point. The GMM is then given by $p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \boldsymbol{\Sigma}_k)$. $\mathcal{N}(\mathbf{x}|\mu_k, \boldsymbol{\Sigma}_k)$ represents a Gaussian distribution with mean $\mu_{\mathbf{k}}$ and covariance $\boldsymbol{\Sigma}_{\mathbf{k}}$. The parameter π_k represents the mixing coefficient for each cluster. Estimating the parameters $\gamma_{ik}, \pi_k, \mu_{\mathbf{k}}$ and $\boldsymbol{\Sigma}_{\mathbf{k}}$ of the Gaussian Mixture Model can be done using the Expectation Maximization (EM) algorithm which maximizes the log likelihood of the model over the data. In the EM algorithm, the parameters are estimated iteratively. Each iteration involves two step, the E step followed by the M step.

1. E step: In this step you will update the responsibilities as

$$\gamma_{nk} = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\mu_j, \boldsymbol{\Sigma}_j)} \tag{6.1}$$

2. M step: In this step you will estimate the remaining parameters by maximizing the likelihood of the Gaussian parameters over the data.

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \mathbf{x}_n \tag{6.2}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} (\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^T \tag{6.3}$$

$$\pi_k = \frac{N_k}{N} \text{ where } N_k = \sum_{n=1}^{N} \gamma_{nk} \tag{6.4}$$

   N is the number of data points.

The E step and M step are repeated till the log likelihood $\sum_{n=1}^{N} ln\{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\mu_j, \boldsymbol{\Sigma}_j)\}$ converges and does not change anymore i.e. the log likelihood is maximized.
In the `Bonus_assignment5.ipynb` notebook provided you will estimate a GMM for the `data2.csv` used in the previous problem. The following steps need to be implemented:

1. First you will initialize the parameters appropriately. Choose a K of your choice based on the results from the previous problem. Assume a uniform mixing coefficient $\pi_k = 1/K$. Initialize the means μ_k for each cluster randomly (Can you think of a smart way to initialize the means?). Lastly, initialize the covariances for each cluster to be a diagonal matrix with diagonal elements equal to one i.e. uniform unit variance.

2. Now repeat the E step and M step iteratively till convergence. You can easily verify convergence by plotting the Gaussians with the means and covariances you have estimated and verify how well they fit the data.

3. Make a contour plot of the learnt Gaussian distribution. Evaluate the log likelihood for the estimated model.

Here the dimensions of the parameters are $\gamma \in \mathcal{R}^{N,K}, \pi \in \mathcal{R}^{K}, \mu \in \mathcal{R}^{K,d}$ and $\boldsymbol{\Sigma} \in \mathcal{R}^{K,d,d}$ where d is the number of features in the data.
Note: The K means clustering algorithm assigns hard probablities to each point i.e. either one or zero, for belonging to a cluster. In fact the K means algorithm can be shown as a special case of GMMs. A

Gaussian Mixture Model with covariances $\Sigma_k = \epsilon \mathbf{I}$ where $\epsilon \to 0$ results in the K means algorithm with hard assignments for each point to their respective clusters. Read 9.2.2 from the book Pattern Recognition and Machine Learning for a detailed explanation.