Data Analysis Methods 2015
Assignment 4

Aasa Feragen, Christian Igel

In Assignment 4 you will work with classification using decision trees, and with clustering. Assignment 4 will be made available 18.3 and your report should be uploaded to Absalon no later than 5.4.

Guidelines for the assigment:

- **The assignments in DAM must be completed and written individually.** This means that your code and report must be unique and written completely by yourself.
- You are, however, encouraged to discuss in small groups while working on the assignments, If you do this, please list your discussion partners in the beginning of your report.
- For this assignment, you are allowed to use numpy functions such as mean, std, sum. You are, however, not allowed to use built-in functions for correlation or hypothesis testing. If you are in doubt, ask!
- In DAM, we will be using an automatic code checking system for some of the exercises. These exercises are written in blue text below, and will be evaluated solely based on your uploaded code. Your solutions to these assignments should therefore be uploaded to the automatic code checking system. To do this, you should:
- Use the supplied code templates, and do not rename them.
- Put all your code templates in a zipped folder called handin1.zip
- Upload your zipped code folder to the website `http://a00508.science.ku.dk/`, using your KU identifier and a password which is sent to you by email.
- If you have not received a password by email, contact aasa@diku.dk.

Decision trees

Exercise 1 (Decision trees)**.** Let us consider classification with input space $\mathbb{R}^d$. As discussed back in Assignment 2, normalizing each component to zero mean and variance one (measured on the training set) is a common preprocessing step, which can remove undesired biases due to different scaling (e.g., when using nearest neighbour classification or logistic regression).

How does normalization to zero mean and variance one (using an affine linear mapping) influence the training process and the classification accuracy of a CART or Random Forest?

Clustering

In this task you are supposed to cluster the data using k-means clustering. You should use the commonly used euclidean distance as distance measure.

Figure 1: Iris setosa, Iris versicolor and Iris virginica. Photos by Radomil Binek, Danielle Langlois and Frank Mayfield distributed under a CC license.

On Absalon, you will find the file `Irisdata.txt`, which consists of measurements made of 50 flowers from each of 3 different *Iris* species (see Figure 1: Iris setosa, Iris Versicolor and Iris virginica). You are given four features: Sepal length in cm, sepal width in cm, petal length in cm and petal width in cm.

Exercise 2. a) Implement k-means clustering and perform clustering on the dataset $X = \mathbf{x}_1, \ldots, \mathbf{x}_n$, where each $\mathbf{x}_i$ represents the measurements of a single flower in the form of a 4-dimensional vector. Use a $k > 1$ of your choice and k randomly chosen data points (from the data set) as initial cluster centers. Please upload your code to the automatic code checking system.

b) Write a script that performs PCA on the dataset and projects every data point onto the two first principal components (PCs). Recall that projection onto a line is given by dot product with the unit vector spanning the line. Please upload your code to the automatic code checking system.

c) Visualize the projected data points from b) in a 2-D scatter plot. How many clusters do you see?

d) Repeat the plot, giving different colors to the clusters found in a). Include the corresponding final cluster centers, visualized with a different marker but the same color as the cluster. What do you see?

Deliverables. a) Uploaded code, b) uploaded code, c) a plot and a one-liner, d) a plot and a one-liner.

Exercise 3. For the first four steps of your k-means algorithm, plot your dataset with color-coded cluster memberships and cluster centers just like in Exercise 2 c). What do you see?

Deliverables. The four plots and a one-liner.

Exercise 4 (Selecting k)**.** The choice of k is often based on prior knowledge about data. Without this knowledge k-means clustering is performed for different k in practice. Now, however, one needs to evaluate the performance for each k.

a) Implement a function `eval_clustering.py` based on the supplied template found in Absalon, which does the following:

1. Runs your k-means clustering for $k \in \{1, \ldots, 10\}$ on a given dataset with N data points $\mathbf{x}_i$ of dimension d, and saves the value of the k-means objective function

$$E(k) = \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in C_j} ||\mathbf{x}_i - \mu_j||^2 \ ,$$

   after termination for every k.
2. Creates a random benchmark dataset of same size by sampling uniformly N values for each of the d dimensions from the interval of the minimum to the maximum of the corresponding dimension
3. Runs your k-means clustering for $k \in \{1, \ldots, 10\}$ on the benchmark dataset and save the final objective error $E^{rand}(k)$ for every k.
4. Computes the gap statistic (as a function of k)

$$G(k) = \log E^{rand}(k) - \log E(k)$$

5. Repeats step (b), (c) and (d) 10 times and compute the average gap statistic for each k.
6. Returns the average gap statistic for $k = 1, \ldots, 10$.

Please upload your function to the automatic code checking system.

b) Plot the average gap statistic for the Iris dataset as a function of k. Argue that the maximum of the gap statistic is a reasonable choice for the number of clusters. What value of k has the best performance?