4. Assignment

Clustering and Decision Trees

Deadline: 26.01.2017

Description of Assignment

This assignment covers the following topics and is based on the lectures 06, 07 and 08:

- Clustering (Task 1)
- Clustering-Bonus (Task 2)
- Decision Trees (Task 3)
- Evaluation (Task 4)

Guidelines

(1) Edit the provided file `readme.txt` to inform us about your environment (i.e.: Octave or MATLAB version, operating system).
(2) It is of utmost importance that your **code is executable**. Make sure that all results are printed and plotted by executing `main.m` and no errors occur.
(3) In the **second line of every file** fill in the following information as a comment: first name, last name and student id.
(4) All provided script and function files have to be used and completed if necessary. If you create additional files make sure that the code is still executable by `main.m`.
(5) All files have to be collected in a **zip-archive** and be uploaded in TUWEL (until the deadline).

Synthetic Clustering Data Set

In Task 1 and 2 synthetic data sets[1] are employed (see Figure 1). You can find eight 2D data sets in the folder `code/datasets/`:

Aggregation.txt: 788 samples, 7 clusters
Compound.txt: 399 samples, 6 clusters
D31.txt: 3100 samples, 31 clusters
Flame.txt: 240 samples, 2 clusters
Jain.txt: 373 samples, 2 clusters
Pathbased.txt: 300 samples, 3 clusters
R15.txt: 600 samples, 15 clusters

[1] http://cs.joensuu.fi/sipu/datasets/

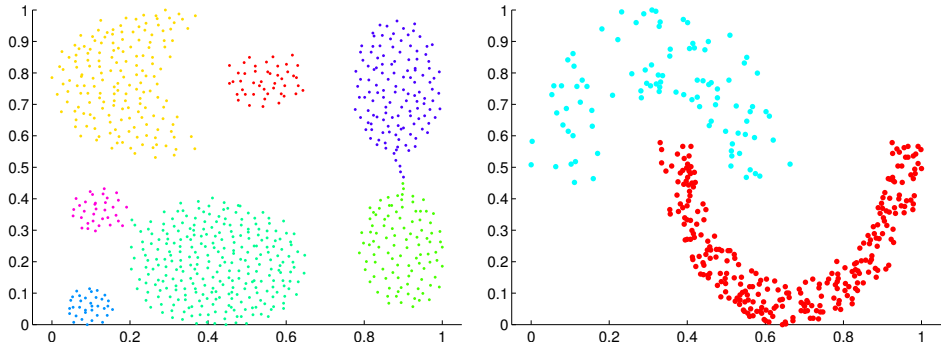

FIGURE 1. Two examples of the data sets for the clustering task.

Spiral.txt: 312 samples, 3 clusters

All data sets use the same format: column one and two represent the 2D samples and column three the ground truth (cluster label), i.e. each row is a sample. Figure 1 shows two examples of the data sets.

Task 1: Clustering

Total points: 15
Files: `main.m`, `clustering.m`, `kmeansClustering.m`

Aim: Implement and understand k-Means clustering. Compare k-Means with hierarchical, agglomerative clustering. The main file of this task is `clustering.m`, which is called in `main.m`.

Choose at least two of the eight synthetic data sets provided in the folder `datasets` for Task 1.

Task 1.1: Pre-Processing [1 Point]. Read-in the selected data sets into MATLAB/Octave and store them appropriately. Do the necessary pre-processing steps for each data set (see `Task 1.1` in `clustering.m`). You can create a separate function file or solve the task directly in `clustering.m`, as you like.

Hint: You will use the Euclidean distance metric in the following clustering tasks.

Task 1.2: Your Own k-Means [7 Points]. The main task here is to implement your own k-Means clustering algorithm. Implement your k-Means in the file `kmeansClustering.m` and call it in `clustering.m` for all selected data sets (see `Task 1.2` in `clustering.m`).

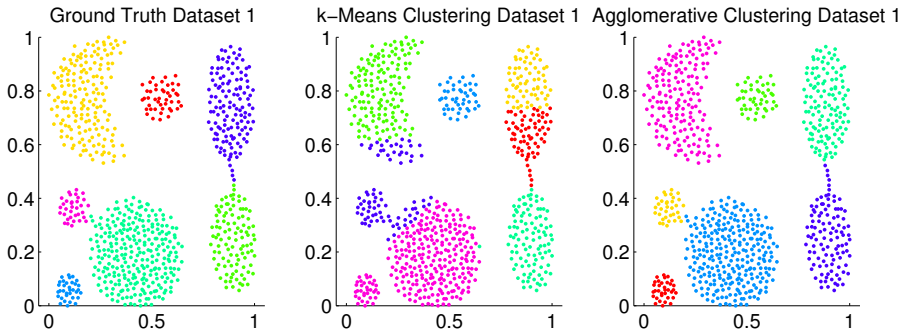

FIGURE 2. Left: Ground Truth; Middle: k-Means Clustering; Right: Agglomerative Clustering

You can assume that the input data is always 2D. Furthermore, the number of clusters is known a prior (see introduction of synthetic clustering data set).

Make use of the function `rng` to always work with the same randomized starting values of the cluster centers.

Task 1.3: Agglomerative Clustering [2 Points]. In this task you have to apply the MATLAB/Octave function `clusterdata` (see `Task 1.3` in `clustering.m`) to do agglomerative clustering on the selected data sets.

Use the same distance measure as in your own k-Means implementation: Euclidean. Furthermore, find out which other parameters are useful to achieve the best clustering results (see drawing next task).

Task 1.4: Show It All [1 Point]. Create one figure for each data set (see `Task 1.4` in `clustering.m`). The figure should contain three sub-plots allowing a visual comparison of the ground truth (correct cluster labels) with the results of k-Means and agglomerative clustering (see Figure 2). Add a meaningful title to each subplot.

Directly code your visualization in `clustering.m` or use a separate script file, as you like.

Hint: Use the MATLAB/Octave function `subplot`.

Task 1.5: k-Means VS. Agglomerative Clustering [4 Points]. Add the figures of Task 1.4 to the report. Answer the following questions in `report.odt`:

(1) Which linkage criterion in agglomerative clustering is best for each of the two chosen data sets and why?
(2) Which clustering algorithm (k-Means or agglomerative) is superior on each of the two chosen data sets and why?
(3) What are the problems you could identify for the two clustering algorithms?

Remark: You can simply compare the results with the help of your drawings (might be tedious for some data sets). Of course, it is also allowed that you quantitatively determine the best clustering by doing some additional coding.

Task 2: Clustering-Bonus

Total points: 5
Files: `main.m`, `clustering.m`, `kmedianClustering.m`

Task 2.1: Implement Your Own k-Median [2,5 Points]. Adapt your implementation of k-Means to do a k-Median clustering. You can make the same assumptions about the data as for k-Means (2D data, number of clusters is known a priori).

Task 2.2: Show It All [0,5 Points]. Add the result of k-Median to the existing plots comparing ground truth, k-Means and agglomerative clustering. Make sure to give the subplot a meaningful title (e.g.: k-Median) to distinguish it from the rest.

Task 2.3: Comparison [2 Points]. This task is to be solved in `report.odt`. Add the figures of Task 2.2 to the report. Based on the answers in Task 1.5, comment on the following:

(1) How does k-Median perform in comparison to k-Means and agglomerative Clustering on the selected data sets?
(2) Do you profit from the increased robustness towards outliers of k-Median in the selected data sets?

Task 3: Decision Trees

Total points: 10
Files: `main.m`, `trees.m`

Aims: Understanding and implementing of CART (=Classification and Regression Trees).

The main file of this task is `trees.m`.

Task 3.1: Your Own CART Method [7 Points]. In this task, the aim is to grow a tree based on the following rules (see lecture 07):

(1) branching factor = 2
(2) numerical criterion $x \leq t$ or $y \leq t$, where t is a threshold, which lies in the middle (geometrically) between data points

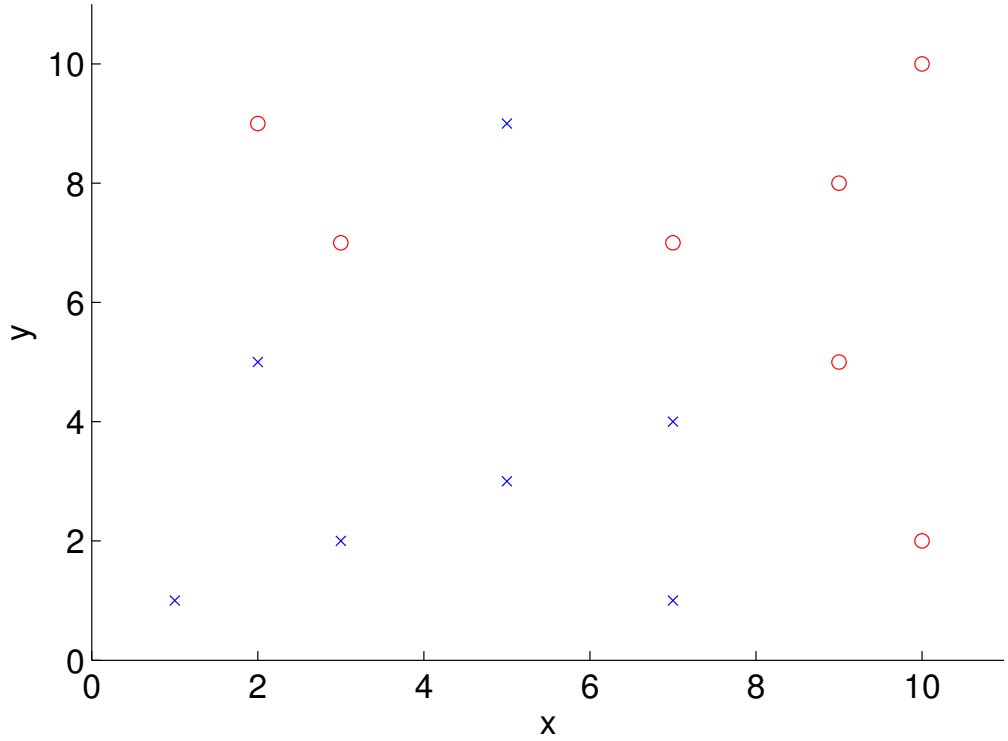

FIGURE 3. Training data for CART method. x: class 1; o: class 2.

(3) always chose the dimension (x or y) and the threshold t, which lead to the biggest change in impurity
(4) a node N becomes a leaf if its impurity $i(N) = 0$
(5) growing of tree stops if all data points are assigned to leafs

Important remark: You have to implement the CART method yourself. It is not allowed to use MATLAB/Octave functions solving this task.

The training data X for your CART can be found in `trees.m` and can be seen in Figure 3. Implement your algorithm in `growTree.m`.

The output of your method should be in text on the console. You have to print the following two lists:

(1) Nodes: list all nodes of the tree including criterion, e.g. Node 1: separation in x, threshold: 8,00
(2) Leafs: list all leafs with their corresponding data points (give the coordinates (x, y) of the data points), e.g.: leaf #1: (1,1) (3,2) etc.

Task 3.2: Drawing Decision Tree and Regions [4 Points]. This task should be solved in `report.odt`. Create the drawings by hand (and scan them) or with any arbitrary drawing tool.

Use the text output of your CART method to:

(1) draw the resulting decision regions into the original feature space (see Figure 3).
(2) draw the resulting decision tree, where nodes should be visualized by rectangular shapes and leafs by ellipses. Write the corresponding numerical criteria into every node's shape and the coordinates of the data points into every leaf's shape.

Task 4: Evaluation

Total points: 5
Files: none

Aim: Determine a confusion matrix for a binary classification problem and calculate index numbers from a given confusion matrix.

Task 4 is to be solved solely in `report.odt`.

Task 4.1: Confusion-Matrix [1 Point]. Determine the following index numbers from the data given in Figure 4:

- True Positive Rate (TPR)
- False Negative Rate (FNR)
- True Negative Rate (TNR)
- False Positive Rate (FPR)

Task 4.2: Index Numbers [4 Points]. Calculate for each class *Precision* and *Recall*. Furthermore, determine the *Overall Accuracy*. Use the data provided in the table in Figure 5.

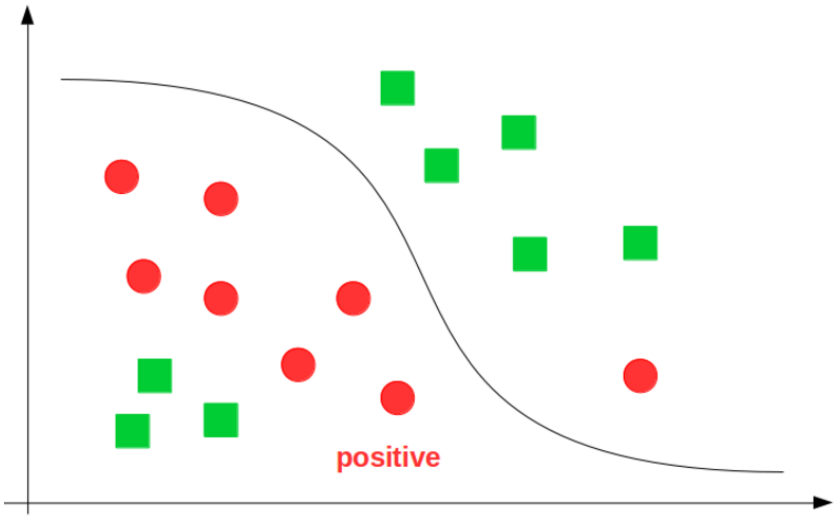

FIGURE 4. Binary classification problem. Decision boundary: black curve. Decision region of positive class marked with "positive" (in red). Positive class: red circles. Negative class: green squares.

Hund	25	5	0	5	0	0
Katze	8	20	0	2	0	0
Vogel	0	0	27	1	0	2
Hase	1	3	1	25	0	0
Igel	0	0	2	1	26	1
Fisch	0	0	2	0	3	25
	Hund	Katze	Vogel	Hase	Igel	Fisch

FIGURE 5. Data for Task 4.2.