Introduction

This report describes what has been done to solve exercises in Home Assignment 1 and what were the results. To run code, execute 'main.m' file. Code repository: https://gitlab.cs.ttu.ee/totahv/iti8565

Exercise 1. Metric function

Implemented distance functions: Canberra, Mahalanobis, Cosine and Minkowski. Minkowski distance function takes in parameter p, so it can compute also Manhattan, Euclidean and Chebyshev distances. Each distance function takes in two points of arbitrary dimension and can be used in the K-means and DBSCAN algorithms (except Mahalanobis).

Distance function results have been compared against built-in functions and results are in the Table 1.

Table 1 Comparison of distance function implemented in home assignment and compared to Matlab built-in functions

Distance	Cosine	Canberra*	Manhattan	Euclidean	Chebyshev	Mahalanobis
Implemented function	1.94	3.00	13.00	7.88	6.30	2.45
Built-in function	1.94	3.00	13.00	7.88	6.30	2.45

*Matlab does not have built-in function for Canberra distance, used Wolfram Mathematica.

Exercise 2. Representative based clustering

Implemented K-means++ algorithm. By default, it uses Euclidean distance. Implemented function, that uses silhouette method to find optimal K value. Mean values are not completely randomized like in standard K-means but uses K-means++ method to find optimal mean values. To test results, unit test was written to compare implemented algorithm and Matlab built-in algorithm, test ran for N = 1000 times and results were the same. Figure 1 shows scatterplot in 2D and 3D using K-means algorithm.

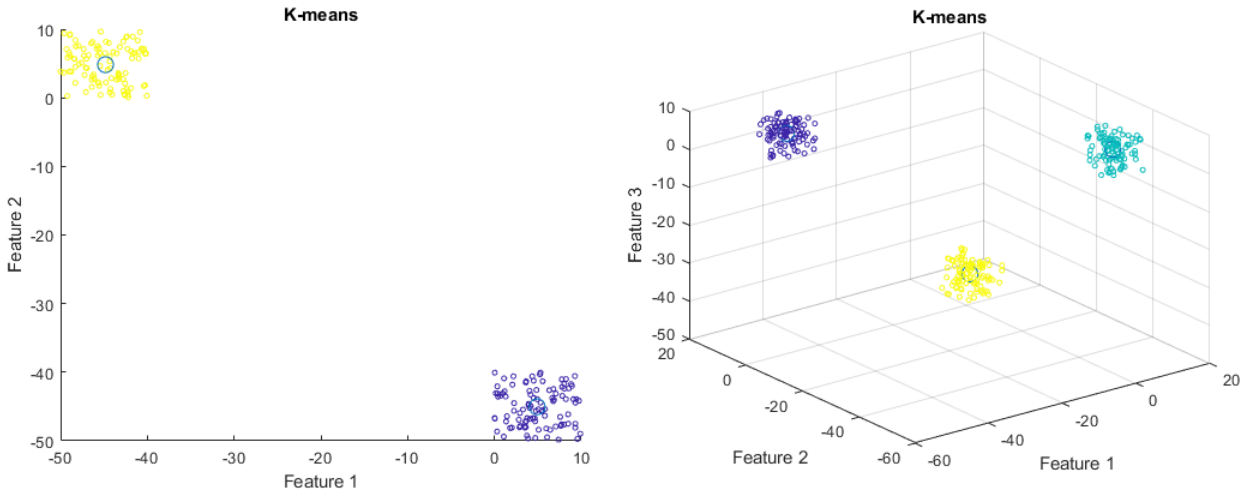

Figure 1 K-means clustering results in 2D and 3D

Exercise 3. Density based clustering

Implemented DBSCAN algorithm. By default, it uses Euclidean distance. Matlab does not have built-in DBSCAN algorithm. To test results, unit test was written to compare results between DBSCAN algorithm written by Yarpiz (found on the Internet) and home assignment implementation. Maximum distance = 20 and minimum points = 10 values were given up front. Results were the same, so clustering worked the same for two separately developed algorithms. Figure 2 shows clustering using DBSCAN algorithm.

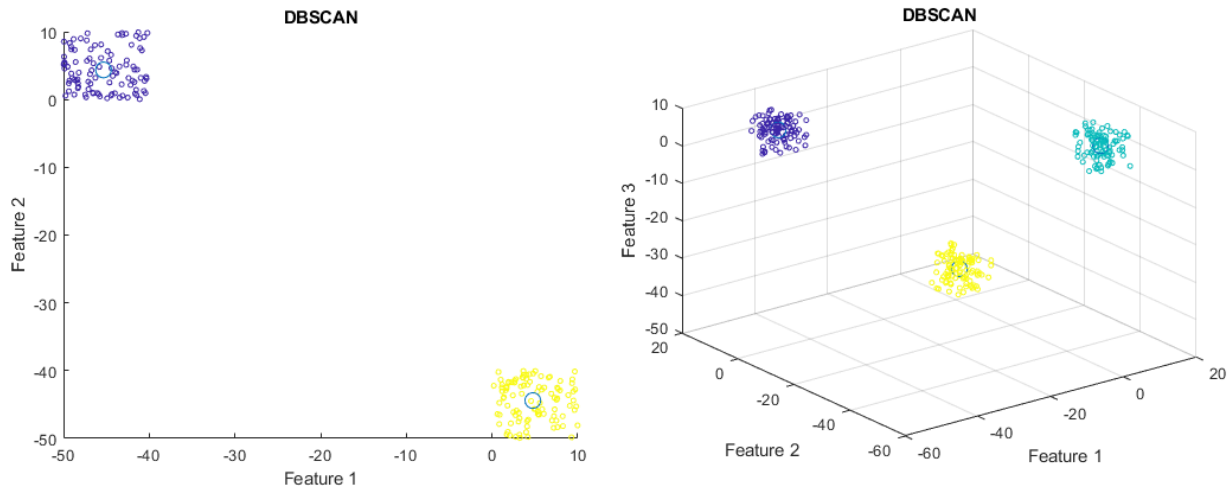

Figure 2 DBSCAN clustering results in 2D and 3D

Exercise 4. Dataset generation

Generated data that has 3 features. In addition to scatterplot, implemented entropy function to describe data. Entropy function is not unit tested because did not find algorithms to compare results with. Figure 3 illustrates features and how values are distributed in selected range, it shows that there are clustering opportunities.

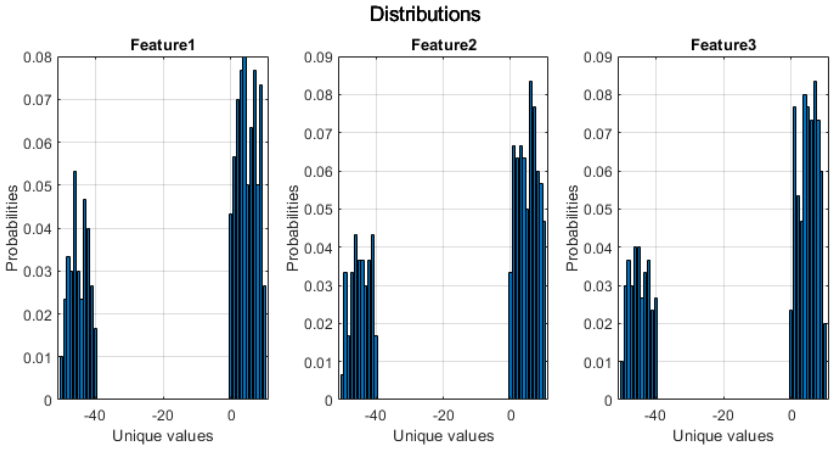

Figure 3 Relation between unique values and probabilities

Conclusion

Implemented 6 distance functions, K-means and DBSCAN algorithms, plotted 2D and 3D results on scatterplot, wrote data generator, calculated entropy and described distributions using bar plot.