

Knowledge Discovery in Databases II  
SS 2017

**Exercise 1: High dimensional data introduction**

**Exercise 1-1      Getting familiar with WEKA**

Download the Data Mining Tool WEKA (version **3.8.0**) and its documentation from <https://sourceforge.net/projects/weka/files/weka-3-8/3.8.0/>. Install the WEKA package and execute the tool. Play around with the WEKA Explorer and get familiar with its functionalities. Download the 5 ARFF datasets (2d, 4d, 8d, 16d and 32d) in the zip file provided. Install the package 'optics\_dbscan' via the tools tab in the WEKA GUI Chooser. The algorithms will then be available as new clusterer methods within the Cluster section of the Explorer. Use the OPTICS algorithm to generate OPTICS plots for the datasets. You can change the parameters for an algorithm by clicking on the field next to the button 'Choose'. For our case, use the Manhattan Distance with MinPts=6 and  $\epsilon=10$  to compute an OPTICS plot for each of the given datasets.

Consider the following questions:

- Can you detect hierarchical clusters?
- How has  $\epsilon$  to be chosen to detect these clusters with the DBSCAN algorithm?
- How do core and reachability distances change for the different data sets?

What are the reasons for the observed effects in the OPTICS plots?

**Exercise 1-2      High-dimensional Data generator (optional)**

Implement a program (in any language of your choice) to generate a high-dimensional dataset with subspace clusters. It shall generate a dataset of specified dimensionality  $d$ , where the values lie within  $[0,100]$  for all dimensions. The data shall have  $k$  subspace clusters. For each cluster a certain dimensionality (i.e. the number of relevant dimensions for the cluster) can be specified. The cluster dimensions are then chosen at random from all dimensions. The objects within a subspace cluster are uniformly distributed within a specified radius around a randomly chosen center point in the relevant cluster dimensions. The clusters shall be generated in a way so that they don't overlap and all cluster points lie within the  $d$ -dimensional hypersphere (no values larger than 100). Furthermore, it shall be possible to generate 'noise' objects, which are points uniformly distributed in the data space (range $[0,100]$ ) in all dimensions.

Write a program that creates a 2-dimensional array of data points, with the columns representing the different dimensions and an additional column indicating the cluster the points belong to ('-1' for noise points). Then the program shall save the data in an ARFF-file (you can also create a CSV-file and edit the header information manually, watch out for appropriate data types though).

The following parameters are to be passed to your program:

- number  $d$  of dimensions
- number  $k$  of clusters

- number of objects in the cluster (for each cluster)
- radius for the cluster in the relevant dimensions (for each cluster)
- dimensionality of the cluster (for each cluster)
- number of noise objects
- output-file

An example of the parameter specification (using Java and passing the parameters in the command line) and the according ARFF-File output can be found below:

Command line:

```

java generator 4 2 3 5 15 20 2 3 2 out.arff

```

Exemplary output file (out.arff):

```

@relation generatedData

@attribute dim1 real
@attribute dim2 real
@attribute dim3 real
@attribute dim4 real
@attribute class {1,2,-1}

% Relevant dimensions
% C1: 2,4
% C2: 1,2,3

@data
18.5,78.2,17.9,40.5,1
17.9,10.2,58.2,38.4,1
20.0,45.9,79.4,43.1,1
65.2,40.0,22.4,53.8,2
70.3,30.0,21.8,89.7,2
74.9,40.2,24.5,7.5,2
73.0,35.2,19.8,34.1,2
68.2,39.5,24.1,15.3,2
15.2,20.1,84.3,19.2,-1

```

Generate a dataset with the following parameters:

- $d = 4$
- $k = 3$
- number of objects in the cluster [40, 80, 70]
- radius for the cluster in the relevant dimensions [10, 7, 15]
- dimensionality of the clusters [3, 2, 4]
- number of noise objects 20
- output-file 'subspace1.arff'

and visualize it in WEKA (use the tab 'preprocess' to import the data and than the visualization tab in the Explorer).

### Exercise 1-3 High dimensional data analysis

- (a) With the data generator of Exercise 1-2, create a sequence of datasets with increasing dimensionality  $D$ . If you didn't implement your own generator you can use the java code (*ArffGen.java*) or the dataset (*ArffGen.zip*) provided on the website.

Use the following parameter settings “ $D$  2 100 100 20 20 2 2 50 *sequenceD.arff*” for generating the data and vary  $D = 2, 3, 4, 5, 10, 25, 50$ .

For each object calculate the ratio “farthest-neighbor-distance”/“nearest-neighbor-distance” by using the Euclidean distance and calculate the average ratio for all objects (of the same dataset). Plot the average ratio for the sequence of datasets with increasing dimensionality. What conclusions can be drawn from this result with respect to the empty space problem/curse of dimensionality? Do you get the same results when using the Manhattan-Distance or the Maximum-Metric instead of the Euclidean distance?

- (b) Use the same sequence of datasets as in the previous task. Let us assume the data space is partitioned into a regular grid with 4 partitions per dimension. For each dataset, generate a histogram (bar chart) that counts the number of cells containing 1 object, 2 objects, 3 objects,  $\dots$ , 250 objects. How do the histograms change with increasing dimensionality of the data? What are your observations? Plot exemplarily the histograms for different dimensions  $D$  above.

- (c) Let  $U_d$  be a  $d$ -dimensional hypersphere with the radius 1 and the volume  $V_d$ . Calculate the radius  $r_d$  of the  $d$ -dimensional hypersphere  $X_d$  that comprises double the volume (i.e.  $V_{new} = 2V_d$ ). Provide a closed-form expression for  $r_d$ , give the limit of the function for  $d \rightarrow \infty$ , and plot the values of  $r_d$  in the range  $d \in [1 \dots 50]$ .

What conclusions can be drawn from these results with respect to the empty space problem/curse of dimensionality?

### Exercise 1-4 Feature Selection

What are irrelevant and redundant features mean? Which feature/features in the following data set is irrelevant or redundant?

| ID  | attribute $X$ | attribute $Y$ | attribute $Z$ | class |
|-----|---------------|---------------|---------------|-------|
| $A$ | 2             | red           | yes           | 1     |
| $B$ | 3             | red           | yes           | 1     |
| $C$ | 3             | green         | yes           | 1     |
| $D$ | 4             | green         | yes           | 2     |
| $E$ | 1             | red           | yes           | 2     |
| $F$ | 1             | green         | yes           | 2     |