Warsaw University of Technology
The Faculty of Electronics and Information Technology

Data Mining (EDAMI)
Project Documentation

Clustering based on density

Authors:

Aleksandra Kurdo
Adam Stelmaszczyk

June 14, 2013

Project task

Implementation and experimental evaluation of DBSCAN [1] and DENCLUE [2] algorithms.

Solution specification

Data set

Set with information about geometrical properties of kernels belonging to three different varieties of wheat was chosen.

Data set attribute information

To construct the data, seven geometric parameters of wheat kernels were measured:

- area A,
- perimeter P,
- compactness C,
- length of kernel,
- width of kernel,
- asymmetry coefficient
- length of kernel groove.

All of these parameters were real-valued continuous.

Algorithms description

DBSCAN algorithm

DBSCAN algorithm is relatively simple algorithm controlled with 2 parameters, namely EPS and MIN_PTS. [1] Basically, we are iterating over a set of unvisited points. If we found a core point (a point which has at least MIN_PTS in his Eps-neighbourhood), we are starting a new cluster. Looking at the neighbours of found core point we are trying to expand this new cluster as much as possible (basing on the Eps-neighbourhood).

DENCLUE algorithm

DENCLUE algorithm is based on the idea that the influence of each data point can be modeled using a mathematical function (influence function). The overall density of the data space can be calculated as the sum of the influence function of all data points. Clusters can be determined mathematically by identifying density-attractors, which are the local maxima of the overall density function.

The DENCLUE algorithm works in two steps.

- Step one:
  It is preclustering step, in which a map of the relevant portion of the data space is constructed. The map is used to speed up the calculation of the density function which requires to efficiently access neighboring portions of the data space.

- Step two:
  It is the actual clustering step, in which the algorithm identifies the density-attractors and the corresponding density-attracted points.

Implementation details

As implementation language for the project Java was chosen. Data set is read from a text file, that located in program root folder, and a set of objects that represent points is generated. On this set DBSCAN and DENCLUE algorithms are runs. Other arguments that algorithms need are declared directly in the program files.

Program consists from five packages:

- algorithms package. In this package abstract class ClusteringAlgorithm and inherited from it classes DBSCAN and DENCLUE are located. Diagram class is shown on the picture 1.

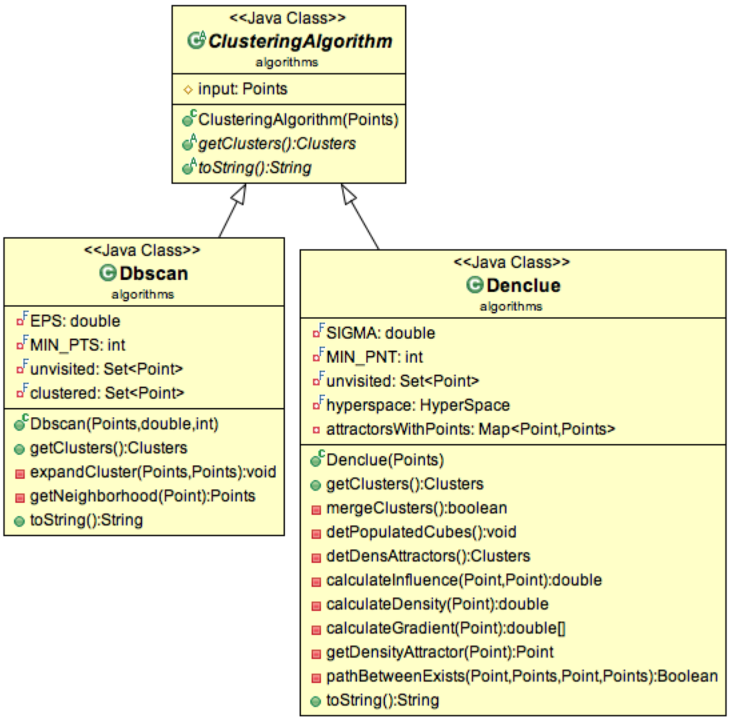


Figure 1: Class diagram for algorithms package

- structures package. Here are located classes Cluster, Point and Points, that used to describe set of points and generated clusters. Diagram class is shown on the picture 2.

- scorer package. This package consists of a class Scorer, that used to generate some statistics about created by algorithm clusters. Diagram class is shown on the picture 3.

- visualizer package. Used for the clusters visualization. Diagram class is shown on the picture 4.

- main package. Used for reading the data set from the file and executing algorithms. Diagram class is shown on the picture 5.

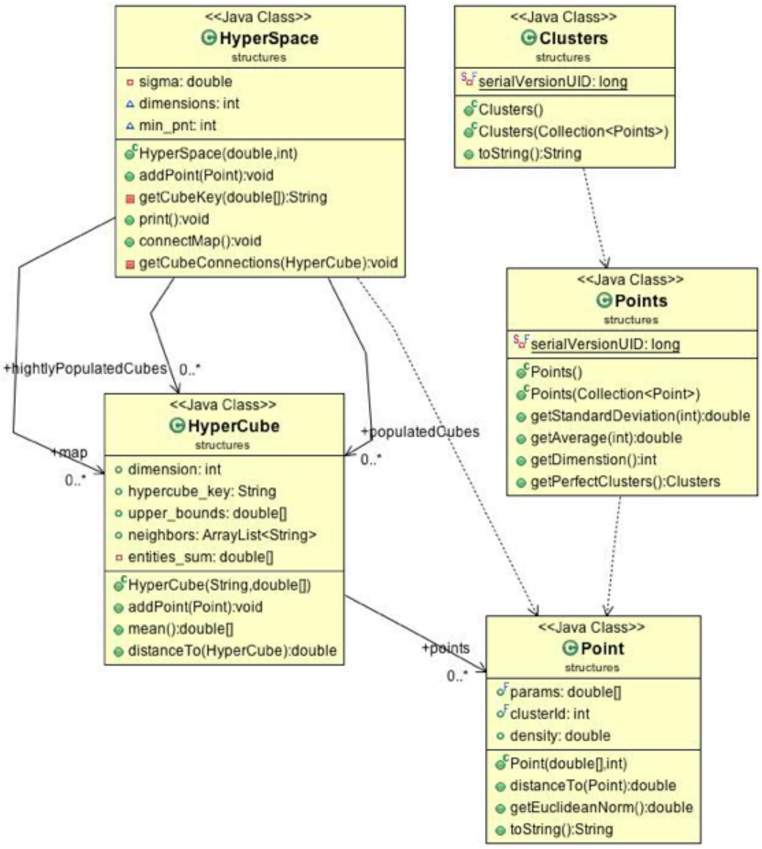


Figure 2: Class diagram for structures package

User guide

Execute files from edami_clustering.zip.

Create an empty java project in Eclipse IDE and import extracted files to it.

Build the project using Eclipse.

To run the program please add execution rights to the run.sh file by typing in console: chmod + ./run.sh

Run program: ./run.sh

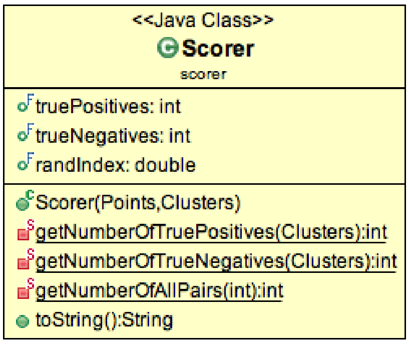


Figure 3: Class diagram for scorer package

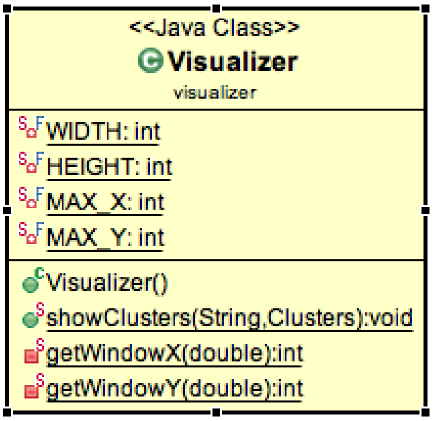


Figure 4: Class diagram for visualizer package

Experimentation results and analysis

As a result of execution DBSCAN and DENCLUE algorythms two sets of clusters are generated. This clusters are represented as points in 2D view as a set of points that belong to them (pictures 6, 7, 8). Each point represented

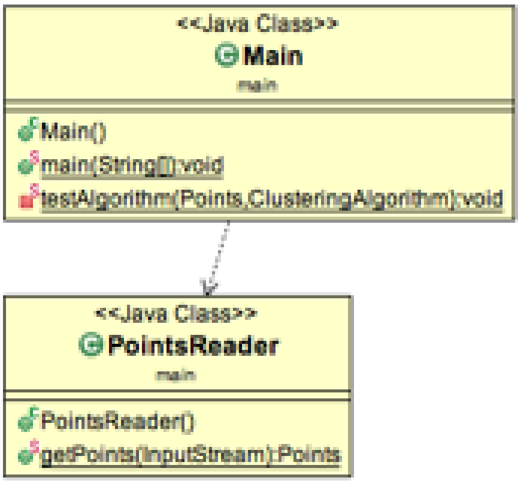


Figure 5: Class diagram for main package

as a number of it's cluster. Data set attributes area and asymmetry coefficient are chosen as a dimensions of the view because of their standard deviation.

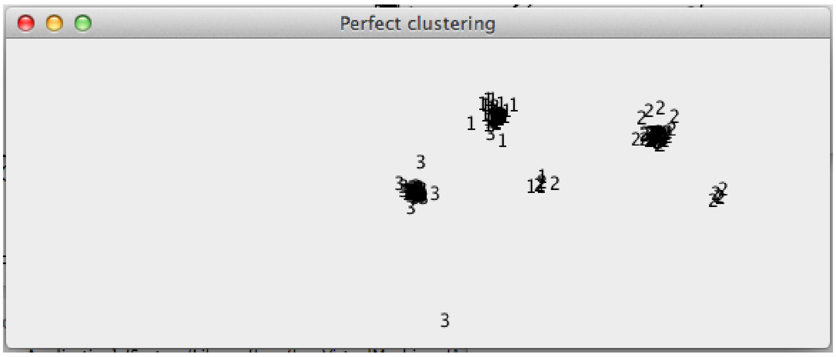


Figure 6: Class diagram for main package

Also some statistics for the running algorithms are shown, such as *true positives*, *true negatives* and *rand index* (pictire 9).

Parameters for the algorithms were chosen experimentally. The best results were achieved using for DENCLUE and DBSCAN algorithms values

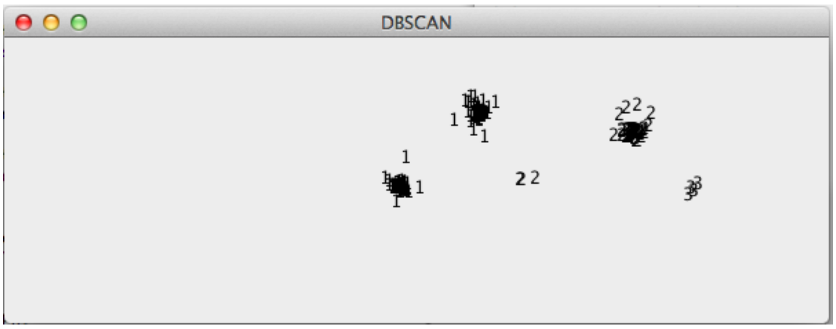


Figure 7: Class diagram for main package

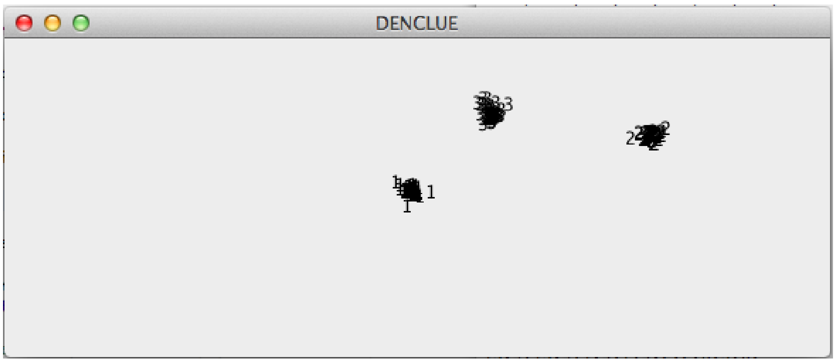
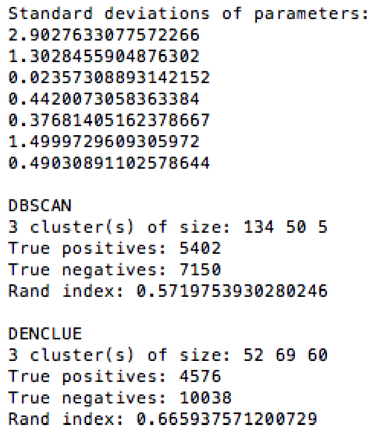


Figure 8: Class diagram for main package



```
Standard deviations of parameters:
2.9027633077572266
1.3028455904876302
0.02357308893142152
0.4420073058363384
0.37681405162378667
1.4999729609305972
0.49030891102578644

DBSCAN
3 cluster(s) of size: 134 50 5
True positives: 5402
True negatives: 7150
Rand index: 0.5719753930280246

DENCLUE
3 cluster(s) of size: 52 69 60
True positives: 4576
True negatives: 10038
Rand index: 0.665937571200729
```

Figure 9: The output of the program

near:

$$sigma = 0.7, eps = 3$$

Conclusion

What can be seen from achieved by DBSCAN and DENCLUE algorithms results that they are are quit different. Rand index for DENCLUE algorithm is higher, what means that its results for this dataset are better. But on the other hand DENCLUE algorithm was needed more time than DBSCAN to do the same work. Should be noticed, that chosen dataset wasn't very big and this difference in time could be changed when algorithms would try to deal with really huge datasets. It can be assumed at the first sight, that for bigger dataset DENCLUE algorithm will be much more slower then DENCLUE, but it's not completely true. Time for DBSCAN algorithm in such situation will rise very quickly and on the other hand such elements in DENCLUE algorithm as dividing data points to cubes, take for big calculations only highly populated cubes ant other elements that were provided to faster the algorithm, will optimize it. And on the tested dataset DENCLUE algorithm dividing cubes for populated and highly populated wasn't a cause of some big decrease of considered cubes.

References

[1] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xianowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, (KDD-96), 1996.

[2] Alexander Hinneburg and Daniel A. Keimm. An efficient approach to clustering in large multimedia databases with noise. 1998.