Institut für Informatik
Prof. Dr. Heiko Röglin
Magdalena Aretz

Lab Efficient Algorithms for Selected Problems: Design, Analysis and Implementation
Winter 2013/2014

Tasklist 2
Due Date: 18.11.2013

Basics

1. **Debugging Java in Eclipse**:
   Go through lessons 1 – 7 of Mark Dexter's *Using the Debugger* tutorial:
   `http://eclipsetutorial.sourceforge.net/debugger.html`.

2. **GNU R**:
   You do not have to understand every single detail of the following material, just program along with the examples to get a rough feeling about how `R` is working.

   - Get familiar with the `R` language. Browse through the examples provided in the sections *Basics*, *Reading Files* and *Graphs* of *R by example* by Ajay Shah:
     `http://www.mayin.org/ajayshah/KB/R/index.html`.
   - For a deeper understanding of the `R` language, have a look at the official manual:
     `http://cran.r-project.org/doc/manuals/R-intro.html`.
     Make sure to cover at least the following chapters: 1 – 3, 7, 8 and 12.1.
   - Look at *A Sample Session* of `R`:
     `http://cran.r-project.org/doc/manuals/R-intro.html#A-sample-session`.

Implement

1. In order to be able to test your code you should download the following datasets from the *UCI Repository*: `iris`, `glass`, `wine`, `cloud`, `seeds`, `abalone`.
   Note: The `cloud` data consists of two parts. Store them both separately.
   Make sure to store the data into your svn folder in order to make it referenceable by relative paths.

2. Implement and test a method that reads .csv files, parses the content and stores it into a twodimensional array.
   Note: Most of the datasets we use contain some dimensions that are not meaningful for clustering (IDs, categorical values etc.) make sure to either delete those by hand or ignore them when reading in the data.

3. Implement at least two different methods for initializing the centers for the `k-Means` algorithm. For example:

   - Take a random subset of the data points,
   - select random points from the input space,

- generate a random initial assignment of the points,
- implement the `k-means++-Algorithm` as described in `http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf`.

4. Make sure that your implementation of the `k-Means method` is working by testing it on the different data sets. If you encounter any problems, solve them by using the debugger (as described in Mark Dexter's tutorial).
5. Implement a `toString()`-method that outputs the state of all crucial variables after each iteration (e.g. number of steps, current value of the error function, size of the clusters, ...).
6. Make sure your code contains an adequate amount of comments.
7. Implement a method that checks after each iteration of the `k-Means method` whether there are empty clusters and solves this problem (e.g. by resetting the respective centers to randomly chosen points from the data).
8. If you want to, you can already start implementing a method that scales the data before running `k-Means` on it. The most common way of doing this is would be to standardize the content of each dimension.
9. `R`: Download `StatET`, a plugin with which you can run `R` in Eclipse. If you encounter any questions or problems combining Eclipse and `R`, have a look at: `http://www.splusbook.com/RIntro/R_Eclipse_StatET.pdf`.
   You will probably have to download the *jr* packages. For this, follow the instructions on: `http://www.walware.de/it/statet/installation.mframe?jump=install-rj-rpkg`.
10. `R`: Experiment with functions in `R` that read data and depict the general structure and distribution. For this, get used to working with the following functions: `read.csv`, `read.table`, `summary`, `plot`, `scatterplot3d`, `boxplot`.
    Note: You do not have to check in any `R` code yet. But be sure you get used to its main functionalities. We will use them later on for the visualization of our experimental outputs.