Data Mining
Homework Assignment #6

Dmytro Fishman, Anna Leontjeva and Jaak Vilo

March 22, 2014

You are free to use any programming language you are comfortable with.

Task 1

Use the data as in the task 4 from the last week, and simulate K-means algorithm (on paper). Use initial centers of (2,6), (2,8), (5,8). Explain the algorithm step-by-step. Next, use the same data and simulate K-medoids (on paper), starting from cluster center points D, E, and H. Data is the following:

	X	Y
A	2	4
B	7	3
C	3	5
D	5	3
E	7	4
F	6	8
G	6	5
H	8	4
I	2	5
J	3	7

Task 2

Can you find 3 initial "centers" for K-means that are different from the centers in the previous task and would produce a different final result? Use 2D plot of the data to assist you.

Task 3

Install and run mldemos (`http://mldemos.epfl.ch/`). Try out the clustering with K-means. Identify situations when K-means clearly does not cluster

as compared to the true clustering (i.e. the clusters expected by you). Make screenshots and discuss, why it happens.

Task 4

Once you have identified the unexpected situations, propose some remedy for it. In other words, propose some heuristics how to overcome these issues.

Task 5

In the lecture we have discussed the principle of self-organizing maps (SOM), which is very powerful and widely used clustering method. Now we shall practice using it.
You are given following 1-D input vector $\mathbf{p} = \{1, 1, 10, 10, 10\}$ that can be regarded as 5 separate 1-D data points. We are trying to cluster these data points using SOM that has four nodes (some time regarded as neurons). The tricky part is that every node in this grid (SOM) is at the same time associated with grid coordinate and weight. In our case four nodes are represented by the grid coordinate vector $\mathbf{v} = \{1, 2, 3, 4\}$ and vector of weights $\mathbf{w} = \{2, 4, 6, 8\}$, such that for example node with grid coordinate 2 has weight 4, and node with grid coordinate 4, has weight 8.
Your task will be to simulate five iterations ($\forall\ p \in \mathbf{p}$) of SOM algorithm, which can be summarized as follows:

- Choose one data point $p \in \mathbf{p}$ at random
- Find the best matching node (in literature best matching unit or BMU) u for the chosen point p, it is the node, whose "weight" has the smallest Euclidean distance to the data point.
- Update each node's weight $w[i], i \in \{1, 2, 3, 4\}$ using the following formula:

$$w[i] = w[i] + \theta(u, v[i]) \cdot \alpha \cdot (p - w[i]), \tag{1}$$

  where $\theta(u, v[i])$ is the neighborhood function, which is defined:

$$\theta(u, v[i]) = \begin{cases} 1 & \text{if } u = v[i] \\ 0.5 & \text{if } |u - v[i]| = 1 \\ 0 & \text{otherwise} \end{cases}$$

  and

$$\alpha = 0.1$$

  where, u and $v[i]$ are the grid node coordinates **not weights** and α is a **learning rate** of the algorithm.

Here we will show how to perform first iteration.

- So, first let us choose a random data point from $\mathbf{p}$, let it be $p = 10$.
- Then we find the best matching node for p, by calculating an Euclidean distance between p and every $w \in \mathbf{w}$ (Note, in 1-D space Euclidean distance between two points is just a difference between those points): p - w[1] = 8, p - w[2] = 6, p - w[3] = 4, p - w[4] = 2. Thus, the best matching node u is the node number 4 with the smallest distance 2.
- Use the formula (1) to update vector of weights, starting with the first element:

  $$w[1] = w[1] + \theta(u, v[1]) \cdot \alpha \cdot (p - w[1]),$$

  let us recall that $u = 4$, $v[1] = 1$, $p = 10$, $\alpha = 0.1$ and $w[1] = 2$. Also, $\theta(4, 1) = 0$, because $|4 - 1| > 1$. Thus,

  $$w[1] = 2 + 0 \cdot 0.1 \cdot 8 = 2,$$

  the same is applied to other three elements.

  $$w[2] = 4 + 0 \cdot 0.1 \cdot 6 = 4,$$

  $$w[3] = 6 + 0.5 \cdot 0.1 \cdot 4 = 6.2$$

  $$w[4] = 8 + 1 \cdot 0.1 \cdot 2 = 8.2$$

Finally, your task is to perform the rest four iterations and report resulting vector of grid node weights $\mathbf{w}$. For this purpose repeat this process four times. Suppose that the data point "10" is chosen during next two iterations, and the data point "1" - during the last two.
If you have not understood a word from above task, try to use several tutorials about SOMs:

- `https://courses.cs.ut.ee/MTAT.03.183/2014_spring/uploads/Main/DM_L05_cluster.pdf`
- `http://www.cbs.dtu.dk/courses/27618.chemo/27618_lecture_clustering_part2.pdf`
- `http://www.csc.kth.se/utbildning/kth/kurser/DD2432/ann14/som.pdf`

Task 6 (bonus, up to 2pt)

Implement SOM from task 5 and compare the results. 1 point will be granted for adapting existing solutions (functions, packages, source codes etc.) and 2 points for your own code effort in any programming language you like. In both cases explain your approach properly, include source code as a separate file (do not list it in the PDF).

Task 7 (bonus, up to 2pt)

Perform clustering analysis on the students' progress data that we have extracted from the grading spread sheet of our course. Data set is attached on the homework web-page. Due to the privacy reasons we could not show personal information. As this exercise is a bonus, don't limit yourself with K-means or K-medoids, try different clustering approaches that you have learned so far.
Pose interesting questions e.g. can you distinguish between different study groups, do you see the difference in grading style of TAs, or perhaps you can identify people that work together on homeworks etc. Try to visualize your hypothesis, use statistics that we have studied previously. Be creative!
Points will be granted depending on how many different data mining methods you will manage to cover - clustering, statistical methods, smart visualization etc.

This data set has missing values that you have to substitute with 0. If you use R, try these two lines (do not forget that copy paste from PDF can be misleading):

```
progress2 = sapply(progress, function(x) as.numeric(gsub(" ",
    "0", x)))
progress2[is.na(progress2)] <- 0
```