Data Mining
Homework Assignment #8

Dmytro Fishman, Anna Leontjeva and Jaak Vilo

April 3, 2014

~~You are free to use any programming language you are comfortable with~~. You will use Weka in this homework. Install weka from `http://www.cs.waikato.ac.nz/~ml/weka`. Click around. Check the tutorial and other documentation here: (http://www.cs.waikato.ac.nz/ml/weka/documentation.html.

Task 1

We will start working with Weka using "Explorer" functionality. Now load diabetes dataset, which comes with the weka. To do so click on "Open file" and find "Weka" folder with "datasets" inside. Use file diabetes.arff for this task. Open it and explore options of "Preprocess" tab. The description of diabetes dataset is here: `http://classes.soe.ucsc.edu/cmps142/Winter10/handouts/diabetes.arff`. Next, choose "Classify" tab. Select J48 classifier ("choose" button and folder "trees"), and run it, leaving default parameters. Interpret the learned tree, plot it(right click on the model in the "Result list"). Characterize the TP, FP, TN, FN rates, accuracy, precision, recall on this data. Make sure that you understand these metrics (intuitively and how they are calculated). What can be learned from this output?

Task 2

Read more about ROC from Konstantin Tretyakov's blog post (`http://fouryears.eu/2011/10/12/roc-area-under-the-curve-explained/`) and/or the following Tom Fawcett's article: `http://tsam-fich.wdfiles.com/local--files/apuntes/ROCintro.pdf`. Plot the Receiving Operating Characteristic curve (ROC) for the model in Task 1 (right click on the model in "Result list" and "Visualize threshold curve"). Interpret it. Take a look at three examples below and answer the questions:

Task 3

Take a look at the following tool: `http://biit.cs.ut.ee/misc/imgvalidate/`. There is a file already uploaded for you. It contains 388 rows (from rev0 to rev387) of a picture. Values in each row correspond to "pixel color values" - rgb components of a pixel. In this file rows are shuffled. For example, if you insert the row ids as they are provided in the shuffled document (like here:), you will see a mess. Your task is to recover an original picture using 1-D Self-organizing maps (SOM), where a vector of grid coordinates $\mathbf{v} = \{0, 1, 2, 3.., 387\}$ and a vector of weights $\mathbf{w} = \{\boldsymbol{w_0}, \boldsymbol{w_1}, .., \boldsymbol{w_{387}}\}$, where $\boldsymbol{w_i}$ is each row of a picture(see HW 6.5 for the explanation of SOM notations). If you manage to cluster and/or serialize rows properly, then you can restore the original picture. What was on the original picture?

Task 4

Compare results from task 3 with another clustering or seriation method (e.g. K-means, DBSCAN, two-way seriation etc.).

Task 5

Look at example of binary matrices from here (tarball here). Write a script that calculates some goodness measure for estimating how good the two-way seriation is. For example, give a reward for 1-s maximally surrounded by 1-s. Explain your measure and describe on toy examples how to maximize this measure by reordering. In the following example, the proceeding matrices would be more expensive than previous ones:

```
10101       10101     11100
01010 ->    10101 ->  11100
10101       01010     00011
```

Task 6 (2pt)

Write a program that performs re-ordering of rows and columns and tries to optimize for measure devised in the task 5. You can attempt some brute force, or evolutionary strategies (genetic programming, differential evolution, simulated annealing, etc). How big examples can you handle and how certain can you be that your code gives a good (or best) answer?