Wine quality analyses using self organizing maps

Michal Horák

***Abstract*— with data mining techniques we can predict wine taste preferences based on physicochemical properties from wine analyses. This work analyzes wine data using self organizing maps. Setting msize value to 45 to 35 is possible to find model with the final quantization error 0.140 and final topographic error 0.02.**

I. Assignment

Pick the dataset, which can be used for clustering. Prepare data and import them to chosen application. Choose clustering algorithm (k-means, hierarchic clustering, SOM). Set up the algorithm parameters for giving the best results.

II. Introduction

The right wine quality evaluation is important for the market. Evaluation prevents the illegal adulteration of wines and assures quality for the wine market.

Wine can be classified by human experts or by physicochemical laboratory tests – pH or alcohol values or density determination.

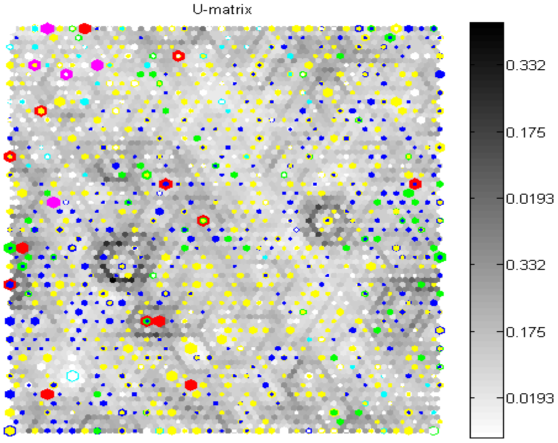

Fig. 1. Shows the U-matrix, where every each color represents value of wine quality.

But there is another method how to classify wine. We can use Data mining techniques on collected wine sets and evaluate wine quality by given raw data.

The dataset is in (.cvs) format and represent ***white*** wine set with over than 4000 rows and each row has 11 attributes.

Each sample was also evaluated by mark in a range from 0 (very bad) to 10 (excellent) which represent a wine quality.

III. Methodology

A. Setup of experiments

For this work is used data mining technique which uses Self Organizing Maps (SOM). Experiments I done is about to find best ***msize*** value – map grid size. Other important parameters: **init**, **neigh.** All these ones are used as parameters for SOM algorithm.

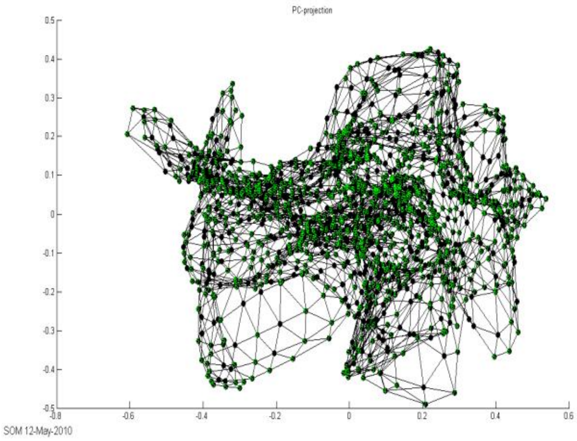

Fig. 2. There is a principal component analysis projection.

B. Configuration of algorithms

From my observation I set following parameters to these values:

- **msize** parameter I set to 45 to 35. So the map grid size is set to 45to 35.
- Value of parameter **init** I set to *linear initialization*.
- And **neigh** parameter I set to a "*gaussian*".

C. Tool used

For this semestral work was used mathematical tool called Matlab - version 7.9.0.529 (R2009b). For my experiments I used prepared scripts which I modified for my needs for experiments. For data analyses I used scripts from SOM tool box 2.0 developed in Helsinki University of Technology.

IV. Experiments

Dataset has all attributes in numerical interpretation. But there were a few rows, with zero attributes, so I had to delete that rows. For experiments I chose 2000 rows for quicker processing.

Wine set wasn't normalized, so there was a range of values from 0, 0100 to 366,500. Because that range, the first step was to normalize the data to range from 0 to 1 value.

The second step was to load the data for processing. For better processing I sort the wine set according to wine evaluation.

The next phase was built and learnt the SOM network. I used the algorithm from SOM toolbox which initialize and train Self-Organizing Map. In this step I set map-size ratio to 45 to 35. For initialization step I set the parameter to linear initialization. And last parameter - **neigh** I set to gaussian, which means that gaussian function is used as a neighborhood function.

After training SOM network, the errors are computed. With that setting I explained above. **Final quantization error is 0.142** and **final topographic error is 0.020.**

Next phase was visualition the results. The **Fig. 4.** represents 12 small graphs of wine attributes where you can see white and black spots. The more white cluster at the graph the more common properities of wine dataset at that attribute.

On the **Fig. 3.** at the left is shown [2] U-matrix or unified distance matrix that visualizes the distance between adjacent units in the SOM. On the right side is shown map grid with winning wine evaluation instance as a label.

Quite similar figure as fig. 3. is **Fig. 1.** where you can find colored U-Matrix, where every each color represents wine evaluation.

Last figure (**Fig. 2.**) represents principal component projection. This means we can display n-size data in 2D with origin distances.

V. Discussion

In this semestral work I decided to use smaller size of map grid for neural network because of time processing and for clearer visualition of results. For example If I used bigger map [90 65] the final quantization error was 0.091 and final topographic error was 0.1 But it took over 1 hour of processing and the graphs wasn’t good.

VI. Conclusion

Due to advances in the data mining, we can extract knowledge from raw data. In this work I solved the regression problem. The algorithm I chose using SOM technique. With this algorithm and with right setting of parameters like an msize, the init parameter and neigh parameter, we can achieve very good results.

The final quantization errors is about 0.140 and final topographic error is about 0.02.

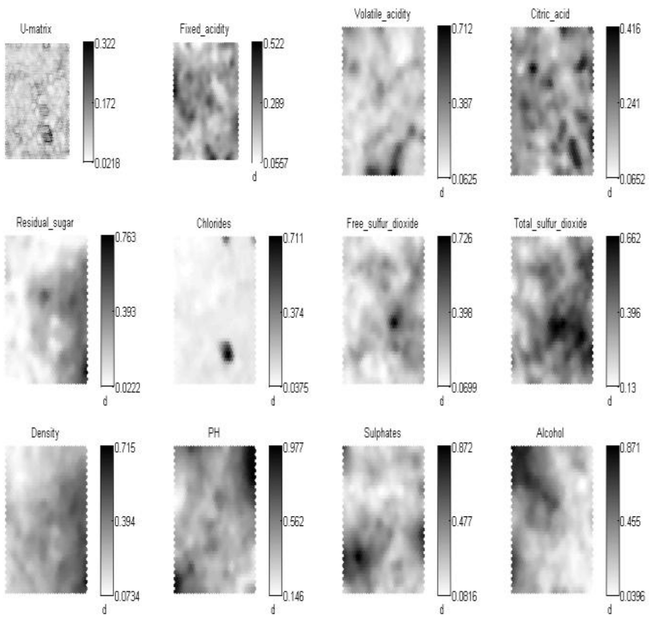

Fig. 4. There are shown symptom figures for every attribute in wine dataset.

References

[1] SOM Toolbox 2.0,], [12. 5. 2010], Available from WWW: http://www.cis.hut.fi/projects/somtoolbox/

[2] [U-Matrix,], [citation: 12. 5. 2010], Available from WWW: http://www.peltarion.com/doc/index.php?title=Self-organizing_map

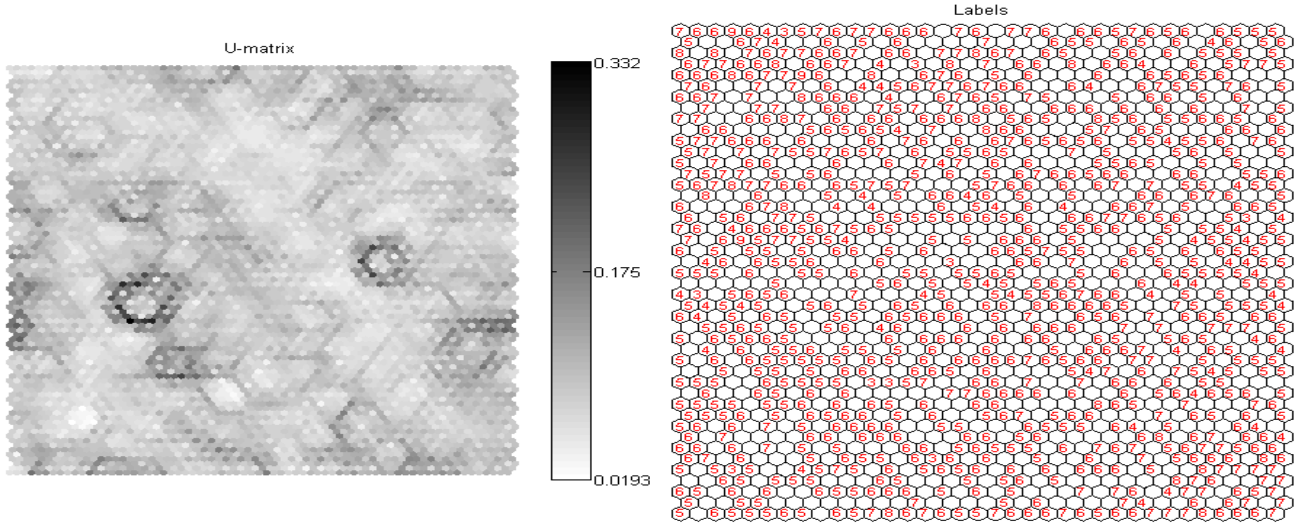

Fig. 3. There are two figures. On the Left you can see the grid with neurons, where the white places represents clusters. The right figure represents the same map, but label with most instances is kept.