Intelligent Data Analysis Computer Laboratory
Practice on classification and clustering software

26th of January – 3rd of February 2012

Arnaud Quirin <arnaud.quirin@softcomputing.es>
http://aquirin.ovh.org/mastersoft/

Part III : The KEEL Software (version 18-12-2011)

KEEL is a software tool to assess evolutionary algorithms for Data Mining problems including regression, classification, clustering, pattern mining and so on. It contains a big collection of classical knowledge extraction algorithms, preprocessing techniques, Computational Intelligence based learning algorithms, including evolutionary rule learning algorithms based on different approaches, and hybrid models such as genetic fuzzy systems, evolutionary neural networks, etc. It allows to perform a complete analysis of any learning model in comparison to existing ones, including a statistical test module for comparison.

The main website of the program is : http://www.keel.es/ or http://sci2s.ugr.es/keel/

Some datasets can be downloaded from this address:

http://sci2s.ugr.es/keel/category.php?cat=clas#sub2

The list of the implemented algorithms, along with additional references can be browsed from this address:

http://sci2s.ugr.es/keel/algorithms.php

Credits: Pr. María José Gacto (2009) for some parts of this tutorial.

Typical use of KEEL

The three main uses of KEEL are:

- Classification of labelled data
- Regression of unlabelled data
- Clustering (unsupervised learning) of unlabelled data

When KEEL opens, a set of choices is given to the user:

- *Data Management*: allows the user to pre-process the data, to convert the data to the KEEL format

- *Experiments*: allows the user to make real experiments with the classification/regression algorithms
- *Educational*: a small mode with a step-by-step tutorial to quickly launch a learning algorithms on some datasets. There is no statistical test in this mode, so we will leave it for the moment.
- *Modules*: some extensions of the original KEEL tool: a tool to manage imbalance datasets, a non-parametric statistical analysis module and a module for multiple instance learning.

Data Management

On the top, a set of different tabs allow to access the different options to pre-process the data.

- The tab/button "Import" allows the user to convert datasets from a large number of formats to the format of KEEL.
- The tab/button "Export" do the opposite.
- The tab/button "Visualize" allows the user to have a preliminary scatter plot drawing of 2 attributes of the data.
- The tab/button "Edit" allows the user to modify some data (useful for missing attributes mainly).
- The tab/button "Preparation" is equivalent to the "filter" mode of Weka: it allows the user to do instance selection, feature selection, discretization, etc. [note: it seems that this tab is absent in the last version of KEEL, but is still documented in the Help menu]
- The tab/button "Partition" allows the user to prepare subset of data: K-fold or 5x2-cross-validation, partitions and stratification methods are implemented.

Experiments

These steps show the typical use of the KEEL Software in this mode:

1. Choose on the left panel a kind of partitions, and a kind of experiments
2. On the next panel, choose a set of datasets (the difference with Weka is that several datasets can be selected at once, and the experiment will be conducted for each of them). To import your own dataset, press the "import" button, select "Import Dataset" and select a .dat file (which has already to be in the KEEL format).
3. Once the dataset(s) are selected, click on the right blank panel to place the "data" icon. After this time, to add/remove datasets, you can use the small buttons on the left panel.
4. The buttons one the left are now accessible. Press the "Preprocess Algorithm" button, select a pre-processing algorithm and place it on the panel on the right. Additional parameters are available by right-clicking on the new icon. The list of algorithms is available here: http://sci2s.ugr.es/keel/algorithms.php#sub1
5. Do the same with the "Algorithm" section. The list of algorithms is available here: http://sci2s.ugr.es/keel/algorithms.php#sub2
6. Do eventually the same with the "Test Analysis" section.
7. Then, select an object in the "Visualize Results" section to visualize your results.
8. Finally, go in the Dataflow section and link all the icons. Some conflicts could appear, in this case you will have to find the proper component which will help you to solve it. Usually a pre-processing algorithm is missing, all the nodes have to be connected, data icon cannot have input, etc.

9. When ready, press the play button (on the top panel). KEEL will generate a ZIP file which contain all the file needed for the experiment. This is another particularity of KEEL: this file can be uploaded to any computer having the Java Virtual Machine installed, and be executed there. This will allow to batch the experiments, to run them in parallel on a super-computer, etc. To run your experiment, you will have to uncompress this file, go in the ./scripts folder and run this command: java -jar RunKeel.jar

Note that KEEL is still in beta-test, and the behaviour of the program can still be unexpected. The program is also still not stable: from a version to another, the GUI can evolve a lot. Please, save your experiments in a regular basis to not loose them (menu File / Save Experiment As).

Objectives of the practice

The objective of this practice is to test and analyze the learning process of various methods to obtain fuzzy rules, with regression datasets. More concretely, we will analyze the learning of the Wang and Mendel method (WM), the methods MOGUL Mamdani and the methods MOGUL TSK.

Data

We will make our experiments using regression data and using the 10-fold cross-validation. The datasets that we will use are the following:

- Electrical-Length: 3 attributes and 495 samples. This dataset is available in KEEL with the name "ELE1".
- Daily-Energy: 7 attributes and 364 samples. This dataset is available in KEEL with the name "daily-electrical-energy".

For each algorithm we want to generate a table of results which will include at least the following information:

- Average number of rules obtained,
- Mean Square Error (MSE) obtained for the learning of the 10 folds,
- Standard deviation of the MSE obtained during the learning,
- SME obtained for the test of the 10 folds,
- Standard deviation of the MSE obtained during the test,
- Execution time in seconds

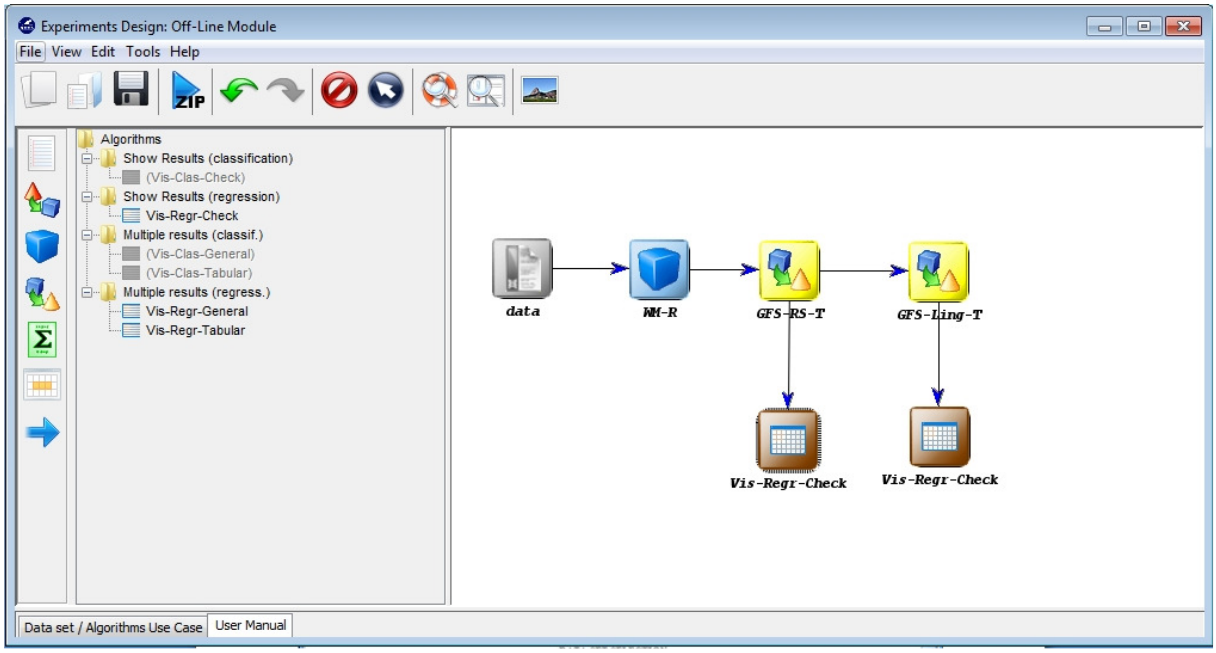
Method WM

It is a method to generate fuzzy rules using a example-based learning. In this section we will analyze the behaviour of the WM method [Wang and Mendel, 1992] using the given datasets. For each one, we will do the following experiments:

- Experiment using the method WM. In KEEL: **Fuzzy Rule Learning / WM-R**.

- Application of a post-processing method (Fuzzy Rule Post Processing) for the selection of the rule base. In KEEL: Genetic Selection of Linguistic Rule Bases (**GFS-RS-T**).
- Application of a post-processing method, to realize the tuning of the fuzzy partitions. In KEEL: Global Genetic Tuning of the Fuzzy Partition of Linguistic FRBSs (**GFS-Ling-T**).
- To analyze the behaviour of these algorithms, it can be useful to use the tool of KEEL: **Shows Results/Vis-Regr-Check**.

In the next figure, we can see an example of an experiment ready to be launched, with the WM method.



Apart from the table of results (see the previous section), you will need to generate a complete analysis of the results, using graphs (obtained on the train as on the test datasets), and comparison about what is the influence of the selection and the adjustment of the membership functions on the obtained results.

Finally, you will have to make an analysis of some properties of the rule base obtained with WM, before and after the steps of the rule selection and the tuning, and try to explain them:

- number of rules,
- complexity of the rules (easily interpretable, hardly interpretable, etc).

These are other properties that could be useful to analyze about a rule base, but unfortunately cannot be obtained (yet) with KEEL:

- consistency: do the rules are consistent, i.e. they are not conflicting for the same input?
- complete: do the rules use all the possible input?
- continuous: do the rules give all the possible output?

You will have to do two analysis (for each dataset, but for only one partition among the 10). To do this analysis you can use the following files:

- <experiment folder> / results / WM-R.<dataset> / result0e0.txt which contain the Rule Base (RB)
- <experiment folder> / results / WM-R.<dataset> / result0e1.txt which contain the Data Base (DB)

Thanks to these files, you will be able to perform you analysis of results.

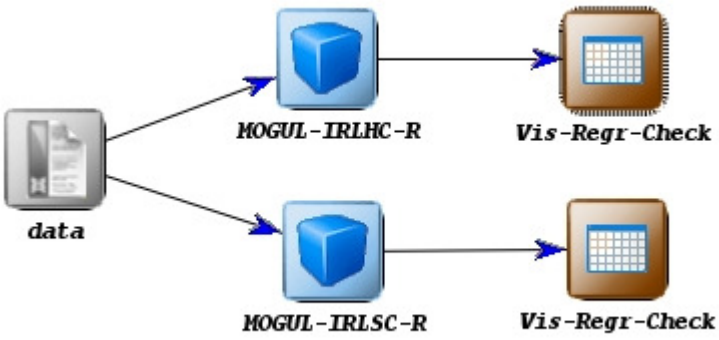
Method MOGUL Mamdani

The MOGUL method is a family of algorithms to obtain genetic fuzzy rule-based systems.

In this section, we will realize an experimentation and an analysis of results obtained by the learning of iterative rules of the Mamdani type, based on two different constrained approaches. For this experiment, we will use the following algorithms (inside the category "Evolutionary Rule Learning"):

- Using the Small Constrained Approach [Cordón and Herrera, 1997]. In KEEL: **MOGUL-IRLSC-R** (in three steps: generation of the rules, selection and adjustment).
- Using the High Constrained Approach [Cordón and Herrera, 2001]. In KEEL: **MOGUL-IRLHC-R** (in three steps: generation of the rules, selection and adjustment).

The next figure show an example of an experiment using the MOGUL Mamdani methods.



To allow the generation of diverse rules between the students, you will use the numbers of your ID card (or passport, driving licence, ..., removing the letters!) as the random seed (see the menu Tools / Seed).

Generate your analysis in the same way as before, using tables and graphs, as well for the train and the test set, and do also a comparison with the WM method.

Method TSK

In this section, we will realize an experimentation and an analysis of results obtained by the learning method of descriptive TSK rules. For this experiment, we will use the following algorithms:

- Learning of descriptive TSK rules [Cordón and Herrera, 1999]. In KEEL: **TSK-IRL-R** (in two steps: generation of the rules, and adjustment).
- Learning of descriptive TSK rules [Alcalá, Alcalá-Fdez, et al., 2007]. In KEEL: **MOGUL-TSK-R** (in three steps: generation of the rules, selection and adjustment).

The same random seed as before should be used. Generate your analysis in the same way as before, using tables and graphs, as well for the train and the test set, and do also a comparison with the WM and the MOGUL Mamdani methods.

Bibliography

- L.-X. Wang y J.M. Mendel. Generating fuzzy rules by learning from examples. IEEE Transactions on Systems, Man, and Cybernetics 22:6 (1992) 1414-1427.

- O. Cordón, F. Herrera, A three-stage evolutionary process for learning descriptive and approximate fuzzy logic controller knowledge bases from examples. International Journal of Approximate Reasoning 17:4 (1997) 369-407.
  URL: http://sci2s.ugr.es/publications/ficheros/ijar-17(4)-369-407.pdf

- O. Cordón, F. Herrera, A Two-Stage Evolutionary Process for Designing TSK Fuzzy Rule-Based Systems, IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics 29:6 (1999) 703-715.
  URL: http://sci2s.ugr.es/publications/ficheros/IEEE-TSMCB-29(6)-703-715.pdf

- R. Alcalá, J. Alcalá-Fdez, J. Casillas, O. Cordón, F. Herrera, Local Identification of Prototypes for Genetic Learning of Accurate TSK Fuzzy Rule-Based Systems. International Journal of Intelligent Systems 22:9 (2007) 909-941.
  URL: http://sci2s.ugr.es/publications/ficheros/IJIS22_9-Alcala_etal-PUBL.pdf