Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, Volume 21, No. 1, ISSN 1726-9679
ISBN 978-3-901509-73-5, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2010
Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird
DAAAM Symposium

IMPROVING THE CLASSIFY USER INTERFACE IN WEKA EXPLORER

ROBU, R[aul]; HORA, C[atalin] & STOICU - TIVADAR, V[asile]

Abstract: *The paper proposes a solution to improve some features of the WEKA data mining and machine learning software. The aim of the changes performed in the Classify section of the Explorer, consists in a simplification of the information displayed when the classifier model is built and tested. Also, a dynamic adaptation of the user interface was done (adding supplementary controls, according to the data set selected for analysis), so that the user could easily perform predictions on a built and tested classifier model.*
Key words: *data mining, WEKA, classification, prediction*

1. INTRODUCTION

WEKA is a machine learning / data mining, open-source, application developed in Java by the Waikato University of New Zeeland. The first internal version of *WEKA* was launched in 1994, and the version that was first made public, version 2.1 was released in 1996. At present, the last stable version reached number 3.6.3. *WEKA* is a very useful software for education, research and applications (Bouckaert et al.,2008). *WEKA*ôs 3.6.3 version offers 71 preprocessing instruments (for discretization, noise reduction, selection of attributes, etc), 117 classification and regression algorithms (among them *J48*, *NaiveBayes* (Wu et al.,2007), *Random Forest* can be found), 11 clustering algorithms (such as *SimpleKMeans*, *XMeans*), 6 algorithms for finding association rules among which the *Apriori* algorithm is encountered, 3 graphical interfaces: the *Explorer*, the *Experimenter* and the *KnoledgeFlow* (Hornik et al.,2009). *WEKA* was downloaded over 1,4 million times since it has been placed on Source ïForge in April 2000 (Hall et al.,2009). This short presentation of *WEKA*, is meant to point out that *WEKA* is a powerful instrument, widely used for the exploratory analysis of data.

The classification algorithms from *WEKA* allow the construction of a classifier model based on the training data and the performance evaluation of the classifier built on the test data. A small part of the measures displayed by default by *WEKA*, after building and testing a classifier model includes: *The kappa statistic* which measures the agreement of prediction with the true class ï1.0 signifies complete agreement (Bouckaert et al.,2008), *the mean absolute error* which is a quantity used to measure how close predictions are to the eventual outcomes, *Relative absolute error* and *Root relative squared error*.

The authors consider that these measures are useful, but a histogram which illustrates the number of the instances that were correctly or incorrectly predicted makes the interface more attractive. Hence, the interface of *WEKA* was modified in such a manner that in an initial phase it displays the number of correct or wrongly classified instances as 3d graphic and the access to the above mentioned information is obtained by pressing the newly added *Advanced Information* button.

One problem regards the use of a classifier model after it has been built and tested, in order to make predictions, because classification is one of the predictive techniques. This can be realized in *WEKA* indirectly, using the *Supplied test set* command (which is usually used in order to test the created model on a specified test data) and analyzing the values predicted by *WEKA* for this dataset. In order to use the *Supplied test set* command, the users must create a new *ARFF* file in which to place the instance or instances whose class they want to predict.

In order to facilitate the work of the users who wish to make predictions and to apply a more intuitive character to the prediction area, the authors conceived and implemented modifications on the *Classify* user interface. In the bottom part of the window, a panel with dynamic content was added. Its controls depend on the analyzed data set and allow the users to easily realize predictions.

2. THE ARCHITECTURE OF THE EXTENDED WEKA APPLICATION

WEKA realizes diverse processing of data sets. In *WEKA* the data set is implemented by the *weka.core.Instances* class. Each instance consists of a number of attributes that can be nominal, numeric or strings. The external representation of an Instances class is a *ARFF* file. The classification algorithms in *WEKA* derive from the abstract *weka.classifiers.Classifier* class which contains its own methods to generate a distribution of probabilities (Witten & Frank, 2005). Preprocessing the data is an important step for the algorithms of automated learning. A useful support for the preprocessing phase is available in the *weka.filters* package which consists of classes with the help of which modifications of the data set can be made. The architecture of *WEKA* application is presented in fig 1. *WEKA* is composed from a multitude of packages. The packages are organized hierarchical starting with the main package named *WEKA*. Any package may contain other packages, files with Java source code or both at the same time. The files in which the modifications were realized are: *GuiChoser.java*, *ClasifierPanel.java*, *Explorer.java*, *PreprocessPanel.java*. The *SimpleBarChart.java* and *SpringUtilities.java* files were added.

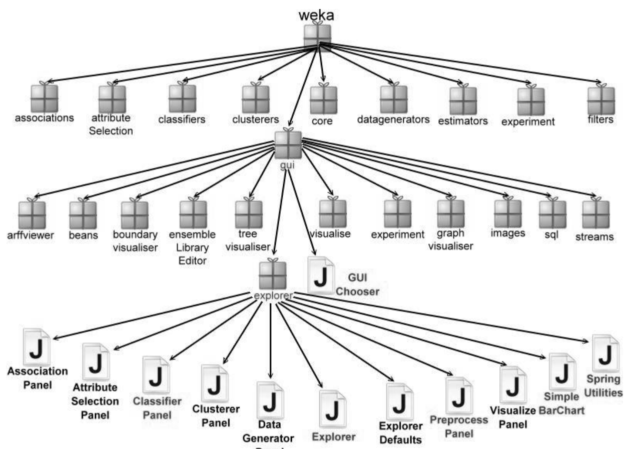
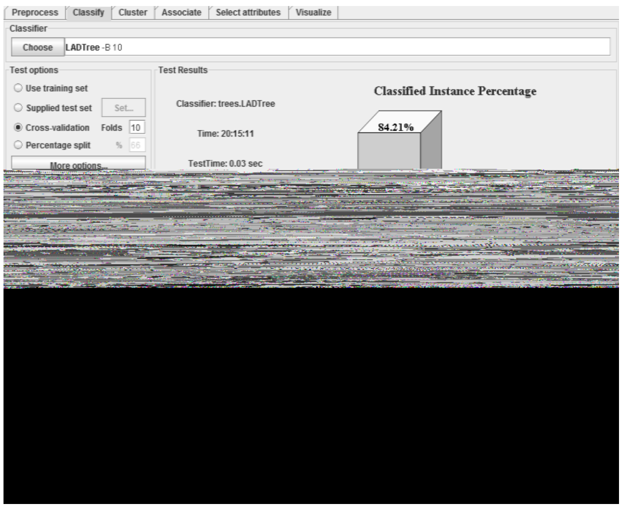


Fig. 1. WEKA architecture



LADTree -B 10

Use training set
Supplied test set
[✓ Cross-validation] Folds 10
Percentage split % 66

Classifier: trees.LADTree

Time: 20:15:11

TestTime: 0.03 sec

84.21%

Fig. 2. The new aspect of the Classify section

3. TEST RESULTS

The *Classifier Output* panel from the *GUI Classify* implicitly displays *the run information* (the number of instances, number of attributes, the testing manner of the model), *the classifier model, the test result (Correctly Classified Instances, Incorrectly Classified Instances, Kappa statistic, Mean absolute error, Root mean squared error, Relative absolute error, Root relative squared error, Total Number of Instances), Detailed Accuracy By Class (True Positive Rate, False Positive Rate, Precision, Recall, F-Measure, Class) and Confusion Matrix* (Tan et Al.,2005).

The outputs selected by default for visualization are the Output model, Output per-class stats, Output confusion matrix, Store predictions for visualization. The user can select the outputs he wants to view with the help of the *More options* command. Along the above mentioned outputs, he may choose between the following: Output entropy evaluation measures, Output predictions, Output additional attributes, Cost-sensitive evaluation, Random seed for xval / % Split, Preserve order for % Split, Output source code.

If no output is selected, run information and test results will be displayed in the *Classifier Output section (Correctly Classified Instances, Incorrectly Classified Instances, Kappa statistic, Mean absolute error, Root mean squared error, Relative absolute error, Root relative squared error, Total Number of Instances).* The authors considered that the majority of the information displayed by default, as well as the information displayed if no output is selected, require further documentation so the access to this information can be realized by pressing the *Advanced Information* button, newly introduced in the interface. Instead of displaying this information by default, the number of correctly or incorrectly classified instances will be displayed as a 3d graphic (see Fig. 2).

4. PREDICTION OF AN INSTANCE

Once a classifier model was built based on a training dataset and this classifier model was tested on the test dataset, with satisfying results, it can be used to classify new instances in order to predict the value of the class attribute with a certain degree of reliability (Frank et al., 2009). In *WEKA* this is rather difficult to achieve by taking the following steps: a new *ARFF* file in which the instance or instances whose class is to be predicted must be created. A random nominal value or a question mark must be filled in, for each instance, in the class attribute which will be predicted. The next step implies setting the *Output predictions* option and loading the *ARFF* file created with the aid of the *Supplied test set* command. Next, we either press a right click on the built model and choose the *Reevaluate model on current test set* command, or we rebuilt the model and test it on the instances from the created *ARFF* file. The authors consider that the process presented is unnatural and is not very intuitive, besides, the question ñHow do I make predictions with a trained model?ò appears in the FAQ list on *WEKA's* website and in other sources. In order to simplify the prediction process, the authors modified the lower part of the *Classify* interface, transforming it into a dynamic section as following: the panel *Prediction of an instance* was added, inside which, for each dataset a number of *JLabels* equal to the number of attributes from the data set will be displayed (one *JLabel* for each attribute), for the nominal attributes the *JComboBoxes* with all the possible nominal values will be displayed, and for numeric attributes *JTextField* cassettes will be displayed. The user must select and complete the values of the attributes for the instance he wants to predict and then press the predict command, as the result will be displayed near the button.

5. CONCLUSIONS

In this paper the authors briefly presented the functionalities of the machine learning and data mining application *WEKA*, its architecture as well as the modifications they conceived and implemented on this open source application. The purpose of the modifications that were done was to simplify the interface with the user inside the classification panel on one side, and on the other side to introduce a new functionality, that is the possibility to easily fill in the values of a new instance on a dynamic interface, generated according to the content of the used data set and to apply the built and tested model on this instance in order to predict the class. The developed interface is dynamic according to the considered data set because on this user interface a number labels equal to the number of attributes will be displayed, one label for each attributes, and for the nominal attributes comboboxes will be displayed so that the user can choose a nominal value from the possible nominal values, and for the numeric and string attributes, text cassettes will be displayed. Once the values for the attributes of the instance were selected, the built classifier model to predict the class can be applied. Further on, the authors want to extend WEKA with a genetic algorithm.

6. REFERENCES

Bouckaert, R.; Frank E.; Hall M.; Kirkby R.; Reutemann P.; Seewald A. & Scuse D. (2008). *WEKA Manual for Version 3-6-0*, University of Waikato, Hamilton, New Zealand

Frank R.; Ester M. & Knobbe A. (2009). A Multi-Relational Approach to Spatial Classification. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 309-318 ISBN:978-1-60558-495-9, Paris, France

Hall M.; Frank E.; Holmes G.; Pfahringer B.; Reutemann P. & Witten I.H. (2009). The WEKA Data Mining Software: An Update, *ACM SIGKDD Explorations Newsletter*, Volume 11 , Issue 1, pp. 10-18

Hornik K.; Buchta C. & Zeileis A. (2009). Open-Source Machine Learning: R Meets WEKA, *Computational Statistics*, ISSN 0943-4062, pp 225-232

Tan P.-N.; Steinbach M. & Kumar V. (2005) Introduction to Data Mining, Addison Wesley, US, ISBN 0-321-32136-7

Witten I.H. & Frank E. (2005). Data Mining Practical Machine Learning Tools and Techniques, Second Edition, Elsevier Inc., ISBN-13:978-0-12-088407-0

Wu X.; Kumar V.; Quinlan R.; Ghosh J.; Yang Q.; Motoda H.; McLachlan G.; Ng A.; Liu B.; Yu P.; Zhou Z.; Steinbach M.; Hand D. & Steinberg D. (2007). Top 10 algorithms in data mining, *Knowledge and Information Systems*, Volume 14, pp. 1-37, ISSN:0219-1377