BIOINFO Optimization Theory and Applications
Volume 1, Issue 1, 2011, pp-01-05
Available online at: http://www.bioinfo.in/contents.php?id=292

WEKA: A Dynamic Software Suit for Machine Learning & Exploratory Data Analysis

P.B.Khanale[1], Vaibhav M. Pathak [2]

[1]*Department of Computer Science,Dnyanopasak College,Parbhani 431 401*
e-mail : prakashkhanale@gmail.com
[2]*Department of Computer Science, Shri Shivaji College, Parbhani -431 401.*
e-mail: vaiby_pat@hotmail.com.

ABSTRACT :

This paper has been written on behalf of a popular Java-based software suit primarily developed with a vision to serve as a workbench for the implementation purpose of machine learning algorithms and an efficient tool for the data mining or data analysis tasks. The software got the name, WEKA, the first release was launched in 1993. Since then some rewrites and modifications were done. Through this paper, the review of this particular software has been carried out, including the history, overview of the workbench which consists of the GUI, the advantages and strengths of the application and advanced features.

1. INTRODUCTION

The software, WEKA, has been named with the extension behind it i.e. Waikato Environment for Knowledge Analysis. In 1993, the University of Waikato in New Zealand started development of the original version of Weka (which became a mixture of TCL/TK, C, and Makefiles). It was come in the picture because of the emerging requirement for a unified workbench that would allow researchers easy access to state-of-the-art techniques in machine learning. At the time of the project's inception in 1992, learning algorithms were available in various languages, for use on different platforms, and worked on a variety of data formats.

The operation of gathering together learning schemes for a comparative study on a collection of data sets was daunting at best. It was envisioned that WEKA would not only provide a toolbox of learning algorithms, but also a framework inside which researchers could implement new algorithms without having to be concerned with supporting infrastructure for data manipulation and scheme evaluation.

These days, WEKA is known as a effective system in data mining and machine learning [2]. It has achieved widespread acceptance within academia and business circles, and has become a widely used tool for data mining research. The book that accompanies it [3][5] is a popular textbook for data mining and is frequently cited in machine learning publications. Little, if any, of this success would have been possible if the system had not been released as open source software [4]. Giving users free access to the source code has enabled a thriving community to develop and facilitated the creation of many projects that incorporate or extend WEKA.

In this paper a brief review of the WEKA workbench, the history of project, discussion of new features in the recent 3.6.4 stable release has been given, along with the summary and discussion containing the advantages and drawbacks of the software being discussed.

Figure 1 WEKA GUI Chooser

2. HISTORY OF WEKA

The first release of WEKA was brought in the market in the year, 1993. This has been done by the University of Waikato in New Zealand which had started

development of the original version of Weka (which became a mixture of TCL/TK, C, and Makefiles). In 1997, the decision was made to redevelop Weka from scratch in Java, including implementations of modeling algorithms [6]. The software was very much at beta stage. The first public release (at version 2.1) was made in October 1996. Figure 4 shows the main user interface for WEKA 2.1. In July 1997, WEKA 2.2 was released. It included eight learning algorithms (implementations of which were provided by their original authors) that were integrated into WEKA using wrappers based on shell scripts and data pre-processing tools written in C. WEKA 2.2 also sported a facility, based on Unix Makefiles, for configuring and running large-scale experiments based on these algorithms. In November 2003, a stable version of WEKA (3.4) was released in anticipation of the publication of the second edition of the book [8] [9]. In the time between 3.0 and 3.4, the three main graphical user interfaces were developed In 2005, Weka received the SIGKDD Data Mining and Knowledge Discovery Service Award[9][10]. May 1998 saw the final release of the TCL/TK-based system (WEKA 2.3) and, at the middle of 1999, the 100% Java WEKA 3.0 was released. This non-graphical version of WEKA accompanied the first edition of the data mining book by Witten and Frank [3][4]. In 2006, Pentaho Corporation became a major sponsor of the software and adopted it to form the data mining and predictive analytics component of their business intelligence suite. Pentaho is now an active contributer to the code base, and the first author is currently the maintainer-in-chief of the software. As of this writing, WEKA 3.6 (released in December 2008) is the latest version of WEKA, which, given the even-odd version numbering scheme, is considered to be a feature-stable version

3. EXPLORATION OF WEKA WORBENCH

The main graphical user interface, the "Explorer," is shown in Figure 1. It has six different panels, accessed by the tabs at the top, that correspond to the various data mining tasks supported. In the "Pre- process" panel shown in the Figure 2, data can be loaded from a file or extracted from a database using an SQL query. The file can be in CSV format, or in the system's native ARFF file format. Database access is Weka 3 provided through Java Database Connectivity, which allows SQL queries to be posed to any database for which a suitable driver exists. Once a dataset has been read, various data preprocessing tools, called "filters," can be applied—for example, numeric data can be discretized. In the Figure the user has loaded a data file and is focusing on a particular attribute, normalized-losses, examining its statistics and a histogram.

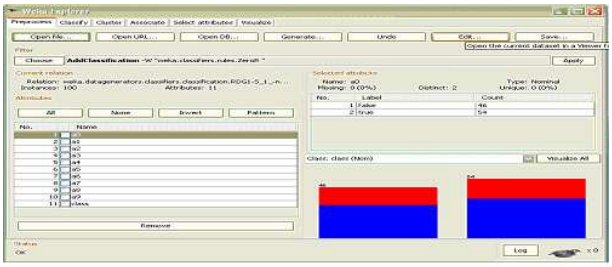

Figure 2 User interface of WEKA 3.6.4 Explorer window

Through the Explorer's second panel, figure 3, called "Classify," classification and regression algorithms can be applied to the preprocessed data. Classification algorithms typically produce decision trees or rules, while regression algorithms produce regression curves or regression trees. This panel also enables users to evaluate the resulting models, both numerically through statistical estimation and graphically through visualization of the data and examination of the model (if the model structure is amenable to visualization). Users can also load and save models.

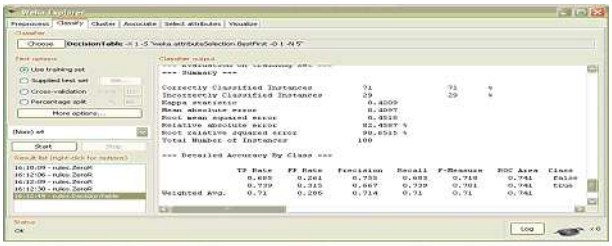

Figure 3 Classify Panel

The third panel, "Cluster," enables users to apply clustering algorithms to the dataset. Again the outcome can be visualized, and, if the clusters represent density estimates, evaluated based on the statistical likelihood of the data. Clustering is one of two methodologies for

analyzing data without an explicit target attribute that must be predicted. The other one comprises association rules, which enable users to perform a market-basket type analysis of the data. The fourth panel, "Associate," provides access to algorithms for learning association rules. Attribute selection, another important data mining task, is supported by the next panel. This provides access to various methods for measuring the utility of attributes, and for finding attribute subsets that are predictive of the data. Users who like to analyze the data visually are supported by the final panel, "Visualize." This presents a color-coded scatter plot matrix, as in Figure 4 and users can then select and enlarge individual plots. It is also possible to zoom in on portions of the data, to retrieve the exact record underlying a particular data point, and so on.

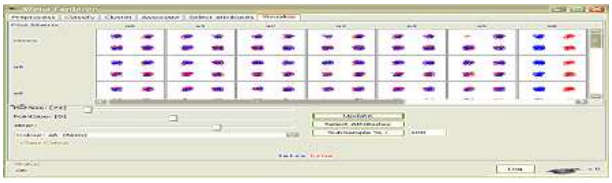

Figure 4 Panel Visualize

The Explorer interface does not allow for incremental learning, because the Preprocess panel loads the dataset into main memory in its entirety. That means that it can only be used for small to medium sized problems. However, some incremental algorithms are implemented that can be used to process very large datasets. One way to apply these is through the command-line interface, which gives access to all features of the system. An alternative, more convenient, approach is to use the second major graphical user interface, called "Knowledge Flow." this enables users to specify a data stream by graphically connecting components representing data sources, preprocessing tools, learning algorithms, evaluation methods, and visualization tool.

4. NEWLY ADDED FEATURES

Many new features have been added to WEKA since version 3.4—not only in the form of new learning algorithms, but also pre-processing filters, usability improvements and support for standards. As of writing, the 3.4 code line comprises 690 Java class files with a total of 271,447 lines of code2; the 3.6 code line comprises 1,081 class files with a total of 509,903 lines of code. In this section, the discussion of these features has been shortlisted.

4.1. Core: The largest change to WEKA's core classes is the addition of relation-valued attributes in order to directly support multi Instance learning problems [6]. A relation-valued attributeallows each of its values to reference another set of instances (typically defining a "bag" in the multi-instance setting). Other additions to WEKA's data format include an XML format for ARFF files and support for specifying instance weights in standard ARFF files.

4.2. Learning Schemes: Many new learning algorithms have been added since WEKA 3.4 and some existing ones have been improved. An example of the latter category is instance-based learning, where

there is now support for pluggable distance functions and new data structures—such as ball trees and KD trees—to speed up the search for nearest neighbors.

4.3. Preprocessing Filters: Some of the new filters in WEKA 3.6 were included. Namely, *Add Classification, Add ID, Add Value, Attribute reorder, Interquartile range, Kernel filter, Numeric cleaner, Numeric to nominal, Random subset* and etc.

4.4. User Interfaces: The "Tools" menu provides two new supporting GUIs as given below:

- ***SQL viewer*:** allows user-entered SQL to be run against a database and the results previewed. This user interface is also used in the Explorer to extract data from a database when the "Open DB" button is pressed.
- ***Bayes network editor*:** provides a graphical environment for constructing, editing and visualizing Bayesian network classifiers.

5. APPLICATIONS

Weka was originally developed for the purpose of processing agricultural data, motivated by the importance of this application area in New Zealand. However, the machine learning methods and data engineering capability it embodies have grown so quickly, and so radically, that the workbench is now commonly used in all forms of data

mining applications—from bioinformatics to competition datasets issued by major conferences such as Knowledge Discovery in Databases. New Zealand has several research centers dedicated to agriculture and horticulture, which provided the original impetus for our work, and many of our early applications. There are countless other applications, actual and potential. As just one example, Weka has been used extensively in the field of bioinformatics. Published studies include automated protein annotation (Bazzan et al., 2002), probe selection for gene expression arrays and classifying gene expression profiles and extracting rules from them. Text mining is another major field of application, and the workbench has been used to automatically extract key phrases from text and for document categorization and word sense disambiguation. The workbench makes it very easy to perform interactive experiments, so it is not surprising that most work has been done with small to medium sized datasets. However, larger datasets have been successfully processed. Very large datasets are typically split into several training sets, and a voting-committee structure is used for prediction. The recent development of the knowledge flow interface should see larger scale application development, including online learning from streamed data. Many future applications will be developed in an online setting. Recent work on data streams has enabled machine learning algorithms to be used in situations where a potentially infinite source of data is available. These are common in manufacturing industries with 24/7 processing. The challenge is to develop models that constantly monitor data in order to detect changes from the steady state. Such changes may indicate failure in the process, providing operators with warning signals that equipment needs re-calibrating or replacing.

6. SUMMARY & DISCUSSION

Weka has three principal advantages over most other data mining software. First, it is open source, which not only means that it can be obtained free, but—more importantly—it is maintainable, and modifiable, without depending on the commitment, health, or longevity of any particular institution or company. Second, it provides a wealth of state-of-the-art machine learning algorithms that can be deployed on any given problem. Third, it is fully implemented in Java and runs on almost any platform—even a Personal Digital Assistant. The main disadvantage is that most of the functionality is only applicable if all data is held in main memory. A few algorithms are included that are able to process data incrementally or in batches. However, for most of the methods the amount of available memory imposes a limit on the data size, which restricts application to small or medium-sized datasets. If larger datasets are to be processed, some form of subsampling is generally required. A second disadvantage is the flip side of portability: a Java implementation is generally somewhat slower than an equivalent in C/C++. Releasing WEKA as open source software and implementing it in Java has played no small part in its success. These two factors ensure that it remains maintainable and modifiable irrespective of the commitment or health of any particular institution or company.

REFERENCES

[1] Bazzan, A. L., Engel, P. M., Schroeder, L. F., and da Silva, S. C. (2002). Automated annotation of eywords for proteins related to mycoplas mataceae using machine learning techniques. Bioinformatics, 18:35S 43S.10

[2] Frank, E., Holmes, G., Kirkby, R., and Hall, M. (2002). Racing committees for large datasets. In Proceedings of the International Conference on Discovery Science, pages 153–164. Springer-Verlag.

[3] Frank, E., Paynter, G.W.,Witten, I. H., Gutwin, C., and Nevill-Manning, C. G. (1999). Domain-specific keyphrase extraction. In Proceedings of the 16th International Joint Conference on Artificial Intelligence, pages 668–673. Morgan Kaufmann.

[4] Holmes, G., Cunningham, S. J., Rue, B. D., and Bollen, F. (1998). Predicting apple bruising using machine learning. Acta Hort, 476:289-296.

[5] Holmes, G. and Hall, M. (2002). A development environment for predictive modelling in foods. International Journal of Food Microbiology, 73:351–362.

[6] Holmes, G., Kirkby, R., and Pfahringer, B. (2003). Mining data streams using option trees. Technical Report 08/03, Department of Computer Science, University of Waikato. Kusabs, N., Bollen, F., Trigg, L., Holmes, G., and Inglis, S. (1998).

[7] Objective measurement of mushroom quality. In Proc New Zealand Institute of Agricultural Science and the New Zealand Society for Horticultural Science Annual Convention, page 51.

[8] Li, J., Liu, H., Downing, J. R., Yeoh, A. E.-J., and Wong, L. (2003). Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (all) patients. Bioinformatics, 19:71–78.

[9] McQueen, R., Holmes, G., and Hunt, L. (1998). User satisfaction with machine learning as a data analysis method in agricultural research. New Zealand Journal of Agricultural Research, 41(4):577–584.

[10] Pedersen, T. (2002). Evaluating the effectiveness of ensembles of decision trees in disambiguating Senseval lexical samples. In Proceedings of the ACL-02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions.

[11] Sauban, M. and Pfahringer, B. (2003). Text categorisation using document profiling. In Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, pages 411–422. Springer.

[12] Taylor, J., King, R. D., Altmann, T., and Fiehn, O. (2002). Application of metabolomics to plant genotype discrimination using statistics and machine learning. Bioinformatics, 18:241S–248S.

[13] Tobler, J. B., Molla, M., Nuwaysir, E., Green, R., and Shavlik, J. (2002). Evaluating machine learning approaches for aiding probe selection for gene-expression arrays. Bioinformatics, 18:164S–171S.