WEKA: A Machine Learning Workbench

Geoffrey Holmes, Andrew Donkin, and Ian H. Witten

Department of Computer Science
University of Waikato, Hamilton, New Zealand

Abstract

WEKA is a workbench for machine learning that is intended to aid in the application of machine learning techniques to a variety of real-world problems, in particular, those arising from agricultural and horticultural domains. Unlike other machine learning projects, the emphasis is on providing a working environment for the domain specialist rather than the machine learning expert. Lessons learned include the necessity of providing a wealth of interactive tools for data manipulation, result visualization, database linkage, and cross-validation and comparison of rule sets, to complement the basic machine learning tools.

1. Introduction

Machine learning is a burgeoning new technology with a wide range of applications. It has the potential to become one of the key components of intelligent information systems, enabling compact generalizations, inferred from large databases of recorded information, to be applied as *knowledge* in various practical ways—such as being embedded in automatic processes like expert systems, or used directly for communicating with human experts and for educational purposes. Presently, however, the field is not well placed to do this. Most research effort is directed towards the invention of new algorithms for learning, rather than towards gaining experience in applying existing techniques to real problems. The WEKA* project (Waikato Environment for Knowledge Analysis) is redressing the balance by applying standard machine learning techniques to a variety of agricultural and horticultural problems. Our goal is to discover and characterize what is required for successful applications of machine learning to real-world data.

To support this effort, a workbench has been developed to provide an integrated environment which not only gives easy access to a variety of machine learning techniques through an interactive interface, but also incorporates those pre- and post-processing tools that we have found to be essential when working with real-world data sets.

Other systems for machine learning experimentation exist [1,2,3], but these are libraries of routines that are intended for use by a researcher who is extending and comparing algorithms. One exception—although still a library of modules—is *Consultant* [4], an expert system that allows domain experts to choose a learning algorithm suited to their needs. *Consultant* assumes that a machine learning algorithm exists that can be applied directly to solve the problem at hand. Our experience has been that although a suitable algorithm may well exist, it is unlikely that its direct application on the domain expert's data will produce a meaningful result. Domain experts, in our experience, need an environment in which they can easily manipulate data and run experiments themselves.

The philosophy behind WEKA is to move away from supporting a computer science or machine learning researcher, and towards supporting the end user of machine learning. The end user is someone—typically, in our applications, an agricultural scientist—with an understanding of the data and sufficient knowledge of the capabilities of machine learning to select and investigate the application of different techniques. In order to maintain this philosophy, we have concentrated on ensuring that the implementation details of the machine learning algorithms and the input formats they require are hidden from the user.

This paper describes the workbench and some of the experiences we have had in

*The Weka is a cheeky, inquisitive native New Zealand bird about the size of a chicken.

applying it to real-world data. We describe some of the support tools for viewing, analysing and manipulating data, and discuss our plans for future development of the workbench.

2. The WEKA workbench

The WEKA workbench, shown in Figure 1, currently runs on Sun workstations under X-windows, and gives access to machine learning tools written in a variety of programming languages (C, C++ and LISP). It is not a single program, but rather a set of tools bound together by a common user interface.

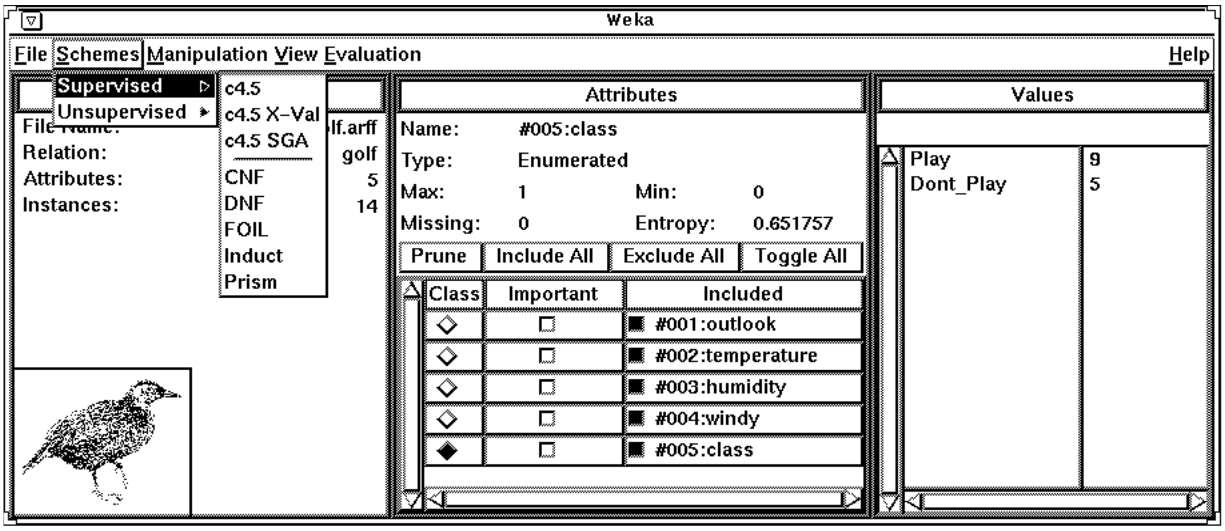
The user interface was implemented using TK/TCL [13], providing rapid prototyping (a first release of the software is currently being established after six months of development by one person) and some portability (all programs other than the learning schemes themselves are written in C).

In designing the interface's "look and feel," we have taken the view that a tool like WEKA will ultimately reside alongside other end-user applications such as spreadsheets, word-processors and databases. This philosophy also extends to files from such programs that are required either as input from or output to WEKA.

WEKA currently includes seven different similarity-based machine learning schemes, summarized in Table 1. Autoclass and Classweb are unsupervised schemes. The rest are supervised learning schemes: C4.5 and OC1 learn decision trees; CNF, DNF, Prism and Induct learn rules, and FOIL learns relations.

2.1. Pre-processing utilities

Real databases invariably contain large quantities of information that must be greatly reduced before processing, and most machine learning schemes only work on comparatively impoverished two-dimensional



File Name: ...lf.arff
Relation: golf
Attributes: 5
Instances: 14

Name: #005:class
Type: Enumerated
Max: 1 Min: 0
Missing: 0 Entropy: 0.651757

Class	Important	Included
◇	☐	■ #001:outlook
◇	☐	■ #002:temperature
◇	☐	■ #003:humidity
◇	☐	■ #004:windy
◆	☐	■ #005:class

Play	9
Dont_Play	5

Figure 1. The WEKA user interface

Scheme	Description	Reference
Autoclass	Unsupervised Bayesian classification	[5]
OC1	Oblique decision tree construction for numeric data	[6]
Classweb	Incremental conceptual clustering	[7]
C4.5 and variants	Supervised decision tree induction	[8]
CNF & DNF	Conjunctive and disjunctive normal form decision trees	[9]
Prism	DNF rule generator	[10]
Induct	Improved Prism	[11]
FOIL	First-order inductive learner	[12]

Table 1: Machine learning schemes supported by the workbench

"flat-file" views of the data. Thus considerable manipulation of a database is invariably necessary before any information can be processed by WEKA. This effort ranges from performing SQL queries on relational databases, through writing macros for spreadsheets, to the invocation of pattern-matching (e.g. Perl) scripts to process text files.

The data set resulting from these operations is then processed on WEKA as follows:

- a data file is selected from the **File** menu;
- statistical characteristics of the data are viewed using XLISPSTAT [15];
- important attributes of the data are selected;
- aggregates of existing attributes are created using the spreadsheet;
- a machine learning scheme is selected from the **Schemes** menu;
- results are viewed as trees, text or three-dimensional plots;
- attribute/aggregate selections are revised;
- the scheme is re-run on the revised data.

In order to maintain format independence, data is converted to an intermediate representation called ARFF (Attribute Relation File Format). Table 2 shows an example ARFF file containing data describing instances of weather conditions and whether or not to play golf (from [8]).

ARFF files contain blocks describing relations and their attributes, together with all the instances of the relation—and there are often very many of these. They are stored as plain text for ease of manipulation. Relations are simply a single word or string naming the concept to be learned. Each attribute has a name, a data type (which must be one of enumerated, real or integer) and a value range (enumerations for nominal data, intervals for numeric data). The instances of the relation are provided in comma-separated form to simplify interaction with spreadsheets and databases. Missing or unknown values are specified by the '?' character.

```
@relation golf
@attribute outlook {sunny, overcast, rain}
@attribute temperature real [0.0, 100.0]
@attribute humidity real
@attribute windy { true, false }
@attribute class { play, dont_play }
% instances of golf games
sunny, 85, 85, false, dont_play
sunny, 80, 90, true, dont_play
overcast, 83, 78, false, play
rain, ?, 96, false, play
```

Table 2: Example ARFF file

When a machine learning scheme is invoked, the data set is converted to the input form appropriate for that scheme using a customized filter. We convert input, and intercept and convert output, individually for each scheme, so that it is not necessary to rewrite a machine learning scheme in order to incorporate it into the workbench. Indeed, the schemes we provide are not all expressed in the same programming language. This policy means that we can capitalize on cleverly optimized implementations that are often available from the scheme's promoter, without needing to worry ourselves about such optimizations. In addition to the filters that are customized for machine learning schemes, a range of filters is available for converting new files to the ARFF format.

For many applications, we have found it necessary to construct new ARFF files from existing ones. Many of the schemes produce results which inform the user that certain attributes do not contribute towards classification, and a user may wish to remove these irrelevant attributes. A basic mechanism for this is provided by WEKA; it in turn encourages further exploration of the dataset. In the "Attributes" column of Figure 1 we see a list of all attributes in the data set, along with information about the currently-selected one. In the list, the checkbox indicates whether or not the attribute will be passed to the learning scheme. The diamond indicates which attribute to classify on when using a supervised learning scheme. If a particular value from the diamond attribute is selected, rules will be formed to differentiate tuples with this value; otherwise, classification rules will be generated for each value. This degree of control is again useful for weeding out unused data items.

A vital requirement, which we discovered only after examining and consulting with the owners of a very large data set [14], is the ability to compute aggregates of attributes. These often lead to a far more satisfactory classification than that obtained with the original attributes. They are typically averages of existing attributes, or differences from an attribute's mean value. They can only sensibly be suggested by people who understand the data (including where and how it came into existence). It is, therefore, imperative to provide within the environment tools for data analysis such as spreadsheets, and basic statistical display tools. WEKA has a built-in spreadsheet capable of generating new ARFF files from existing ones; moreover, it invokes a statistical package (XLISPSTAT [15]) to display histograms of attribute values, scatterplots, boxplots, and three-dimensional views of the data.

2.2. Post-processing utilities

In an environment where many different learning schemes are available, it is important to be able to evaluate and compare the results produced by each one. WEKA provides for cross-validation studies to be performed, and incorporates a new method for comparing classifications and rule sets [16]. This method evaluates classifications by analyzing rule sets geometrically. Rules are represented as objects in *n*-dimensional space, and the similarity of classes is computed from the overlap of the geometric class descriptions. The system produces a correlation matrix that indicates the degree of similarity between pairs of classes. For applications with only a few attributes, the way in which different rule sets cover the data they were inferred from can be viewed and compared.

3. Real world applications

Table 2 lists the data we have collected and the questions we have tried to address using WEKA. Most of the applications are presently at a rather early stage of development.

4. Conclusions and future work

Our experience of processing real-world data sets has enabled us to identify a number of issues that influence future development of WEKA. Most agricultural and horticultural data is characterised by one or more time-dependent attributes that are largely ignored or misused by the schemes in the workbench [14]. This has led us to consider the role of sequence-identification schemes such as hidden Markov models in machine learning. We plan to develop hybrid schemes for WEKA which identify time-dependent data directly, and interpret it correctly.

As mentioned earlier, most data originates from a database which is typically constructed from many relations. Performing many *join* operations on the data, which is generally necessary to convert it into the flat file format expected by most machine learning techniques, inevitably introduces greater duplication and functional dependencies [14]. We are enabling the machine learning schemes to be applied directly to the data in the database in much the same way as systems that perform knowledge discovery in databases [17]. This approach allows owners of the data to use WEKA immediately, rather than waiting for a macro, Perl script or database query to be written.

Finally, as a further encouragement to knowledgeable users, we are investigating the feasibility of adding an interactive scheme such as PROTOS [18] to the workbench.

Data	**Application**	**Reference**
Dairy herds	What rules do farmers use when they decide to cull or retain cows? WEKA processed 19000 records each containing 705 attributes	[14]
Human diabetes	Can machine learning schemes produce a good clinical diagnosis of human diabetes?	[16]
Recumbent cows	Why do some cows lie down and never get up again?	
Grape properties	What factors influence the quality of a prize-winning wine?	
Organic farming	What are the differences between neighbouring conventional and organic farms?	
Mastitis and oestrus in cows	Can these conditions be predicted?	

Table 2: Applications currently under investigation

This will allow users to build applications using both their specialist knowledge of the data and the learning capability of the program. In many ways this is the perfect match of skills needed to build successful intelligent information systems using machine learning.

Acknowledgments

This work is supported by the New Zealand Foundation for Research, Science and Technology. We gratefully acknowledge the stimulating research environment provided by the Waikato machine learning group.

References

[1] R. Kohavi, G. John, R. Long, D. Manley and K. Pfleger, “MLC++: A Machine Learning Library in C++,” *Tech Report*, Computer Science Dept, Stanford University, 1994.

[2] R.J. Mooney, Ml–code machine learning archive. Available by anonymous ftp at cs.utexas. edu:/pub/mooney, 1991.

[3] F. Zandt, Private communication.

[4] D. Sleeman, “The role of CONSULTANT in helping domain experts use machine learning.” *Workshop on Fielded Applications of Machine Learning,* University of Massachusetts, Amherst, 1993.

[5] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor and D. Freeman, “AUTOCLASS: A Bayesian classification system.” *Proc Int Conf on Machine Learning*, pp 54–64, Ann Arbor, MI: Morgan Kaufmann, 1988.

[6] S.K. Murthy, S. Kasif, S. Salzberg and R. Beigel, “OC1: Randomized Induction of Decision Trees.” *Proc National Conf on Artificial Intelligence*, pp. 322–327, Washington, D.C., 1993.

[7] D. Fisher, “Knowledge Acquisition Via Incremental Conceptual Clustering.” *Machine Learning*, 2, pp. 139–172, 1987.

[8] J.R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1992.

[9] R.J. Mooney, “Encouraging Experimental Results on Learning CNF,” *Tech Report*, University of Texas, 1992.

[10] J. Cendrowska, “An algorithm for inducing modular rules.” *Int. J. Man-Machine Studies,* Vol 27, No 4, pp 349–370, 1987.

[11] B.R. Gaines, “The tradeoff between knowledge and data in knowledge acquisition in knowledge discovery in databases,” *AAAI Press*, pp 491–505, 1991.

[12] J.R. Quinlan and R.M. Cameron-Jones, “FOIL: a midterm report,” *Proc European Conf on Machine Learning*, pp 3–20. Springer Verlag, 1993.

[13] J.K. Ousterhout, *Tcl and Tk toolkit*. Addison Wesley, 1994.

[14] R.J. McQueen, D.L. Neal, R. DeWar, S.R. Garner and C.G. Nevill-Manning, “The WEKA machine learning workbench: its application to a real world agricultural database,” *Proc Canadian Machine Learning Workshop*, Banff, Canada, 1994.

[15] Tierney, L., *LISP-STAT: an object-oriented environment for statistical computing and dynamic graphics*. Wiley, New York, 1990.

[16] T.J. Monk, S. Mitchell, L.A. Smith and G. Holmes, “Geometric comparison of classifications and rule sets,” *Proc AAAI workshop on knowledge discovery in databases*, Seattle, Washington, July 1994.

[17] G. Piatetsky–Shapiro and W.J. Frawley (Editors), *Knowledge discovery in databases*. AAAI Press, Menlo Park, CA, 1991.

[18] B.W. Porter, R. Bareiss and R.C. Holte, “Concept learning and heuristic classification in weak-theory domains,” *Artificial Intelligence 45*, 229–263, 1990.