E4ML: EDUCATIONAL TOOL FOR MACHINE LEARNING

Mohd Shamrie Sainin, Fadzilah Siraj
Artificial Intelligence Special Interest Special Group (AISIG)
School of Information Technology, Universiti Utara Malaysia
06010 Sintok Kedah, Malaysia
shamrie@uum.edu.my, fad173@uum.edu.my

Abstract – ***There are various types of machine learning algorithms with certain processes taken by the algorithm. In teaching of the machine learning algorithms, such processes need to be explained especially to the beginner in introductory level. This paper discusses the development the tool that addresses the process by certain algorithm to produce a hypothesis or output based on given data. This tool can also be used in teaching and learning purposes. The explanation of processes by the algorithms is demonstrated through simple simulation. The source of the algorithms was adapted from Mitchell book [1] that cover popular algorithms in machine learning for teaching and learning such as Concept Learning, Decision Tree, Bayesian Learning, Neural Networks, and Instance based Learning. The tool also used several classes of Weka (Waikato Environment for Knowledge Analysis) as a basis for the design and implementation of the new tool that focuses on explaining the processes taken by certain algorithm.***

Keywords:
Machine Learning, Explaining Machine Learning Algorithms, Algorithm Simulation.

I INTRODUCTION

Machine learning is one of Artificial Intelligence areas that focus on studying the algorithms for learning from data so some knowledge can be extracted. There popular learning algorithms in machine learning are: concept learning (CL), decision tree learning (DT), Bayesian learning (BL), inductive logic programming (ILP), instance based learning (IBL), evolutionary learning (EL), artificial neural networks (ANN), analytical learning (AL), and reinforcement learning (RL). Table 1 describes a brief summary of the learning algorithms. Machine learning is a computer program that programmed to learn and improves itself at some task trough experience. The definition of machine learning is that "a computer program is said to learn from experience E with respect to some class of task T and performance measure P, if its performance at task in T as measured by P, improves with experience E" [1].

As described in [5], the *knowledge discovery in databases* [2] [3] [4] was attract researches and enabled various efforts to develop a generic or even specialized tool that can be used in data analysis using machine learning algorithms. Although we have are many tools that freely or commercially available in machine learning, but the problem is most of the tools are complicated to be used by the student to learn the very basic algorithms.

II MOTIVATION

In early motivation to machine learning, [6] reported that the awareness of the use of machine learning is less popular than the accepted traditional statistical analysis. The defined problem why machine learning is less popular because:

- Machine learning is not broadly used, publication of research results might be difficult for other researchers (new) to understand
- Some researchers would be interested in using machine learning in future studies, provided they could learn more about how it should be used.

According to problem above, tool for beginner is the best solution to provide hands-on learning and simulation on machine learning algorithm. Hands-on learning or learning by example becomes more important and useful to teach a beginner in certain application. Most of the existing machine learning or data mining product or tool is more suitable for industrial application. Hence the tool is more complicated to be used as a tool for illustrating the Machine learning algorithm

functionality. The evaluation of fourteen desktop tools that used in Data Mining (using machine learning algorithm) is explained by [5].
Simulation as described by [7] is a very important technique that can be used in any area of application and should be a part of student education. The use of simulation tool in teaching however depends on user model that differentiate the background and their future needs. Stahl explained that there are three factors that affect how simulation tool that can be used in education, a knowledge background (particular in programming and statistics), future undertakings by the user (are they just doing simulation or doing another prototype) and the total teaching time for the simulation tool.

Since we are targeting the novice user or beginner student for the tool, we are dealing with user model indirectly for the introductory level. In tutoring system, student model is very important as a background for knowing the user understanding of the knowledge domain and provided with the appropriate decision by the pedagogical module [8].

Although we are not building the student model in the simulation tool development, we are still observing the student knowledge and basic graphical user interface in order to meet their current ability. In the process to design and develop the simulation tool, we are focusing at Weka that claimed to be comprehensive software resources, full and industrial-strength in Machine learning methods [9].

Apart from Weka, specific machine learning tool known as EGALT (Educational Genetic Algorithm Learning Tool) [10] that specifically developed to help students to facilitate Genetic Algorithm course. In the similar way, the development of general-purpose tool will help students to learn machine learning together with taught course more widely.

The data collection from UCI Machine Learning [11] is the source for sample data that will be used in the tool development. UCI Machine Learning collection provides different types of numerical and nominal data that covers discrete sequence, multivariate, relational, text and time series data. The other data for experimenting the tool is adapted from Weka classifier distribution.

III WEKA CLASSIFIER

Weka is an acronym for "Waikato Environment for Knowledge Analysis" and was written in Java as a multi-platform operating system programming language [9]. Weka software was developed at University of Waikato New Zealand and has been used as a tool for analyzing agricultural data set especially to address lots of data preprocessing and finally answering various questions to the agricultural data [12]. It employs various types of algorithm in Machine learning that consists of three modules known as data preprocessing, machine learning function selection, and output processing. Weka provides many algorithms for implementing classes that can be used based on Open Source concept. To date, the new version of Weka is 3.2.3 for the windows version.

The related projects that used Weka as a basic development include Tertius System [13], Tertuis extension to Weka [14], KDDML-MQL: Knowledge Discovery in Databases Markup Language [15], KEA: Practical Automatic Keyphrase [16] and YALE: Yet Another Learning Environment [17].

The Tertuis system is a tool for learning a first order logic rules and uses Weka as an extension to the system rule production. The system performs an optimal best-first search, finding the most confirmed hypotheses, and includes a non-redundant refinement operator to avoid duplicates in the search. Tertius can be adapted to many different domains by tuning its parameters, and it can deal either with individual-based representations by upgrading propositional representations to first-order, or with general logical rules [18]. KDDM-MQL is the XML based environment for Knowledge Discovery in Database and integrated with Prolog engine together with the wrapper for Weka machine learning classes. KAE is an algorithm that automatically extracting the Keyphrase from text and uses the Naïve Bayes machine learning algorithm. YALE was developed at University of Dortmund, Germany and focusing on environment for machine learning experiments. It includes various machine learning Algorithms such as support vector machines for regression classification, decision tree, clustering algorithms and a wrapper to all Weka classifiers.

IV SYSTEM DESIGN AND IMPLEMENTATION

The educational tool developed for machine learning system uses several Weka classes that has been modified to meet the teaching and learning requirements. The system was designed and implemented using object oriented Java Programming language with Swing UI capable and is named e4ML. The initial design of the tool includes the following components that are grouped into four tabs as Data tab, Data Viewer tab, Simulation tab and Tool Utility tab:

- Data file reader
- Data file construction
- Data file object collection (instances memory)
- Data file viewer
- Machine learning algorithms collection (Algorithms Module)
- User Interfaces and simulation
- Testing utility

Figure 1.0 shows the basic architecture for the educational tool that consists of two main modules, they are:

- Interface and simulation module, and
- Algorithms module.

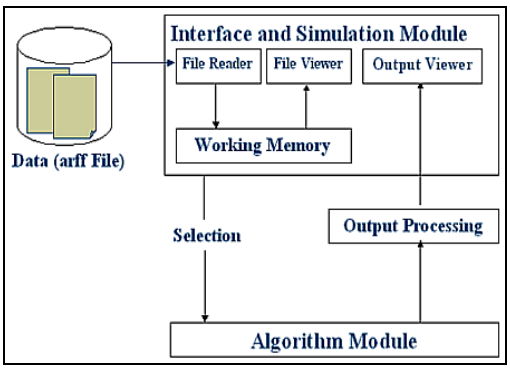

Figure 1.0: Basic Architecture of the Educational Tool for machine learning

An interface and simulation module will interact with user and thus acquiring the data that has been simulated using certain machine learning algorithm from the Algorithm Module. There are two options provided for the simulation (not every algorithm), step-by-step execution or final output oriented simulation. The output simulation of the tool is the explanation process taken by the certain algorithm when producing hypotheses or classification according to the provided data.

A. *Data file*

Data file or database for the tool represented as an arff extension that can be used in Weka classifier. The arff file consists of data file header information and data instances as an input to the tool. The e4ml tool uses Weka class for reading the file and converts the data instances into object representation for other classes or modules. Table 2 depicts the arff file representation that has been used in this tool.

Table 2: The example of Weka and e4ML arff file representation.

```
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

B. *Machine Learning Algorithms Collection (Algorithms Module)*

The current e4ML algorithm collection module is equipped with six algorithms that mainly in Concept Learning (Find-S and Candidate Elimination algorithm), Decision Tree (ID3), Bayesian Learning (Simple Naïve Bayes), Instance Based Learning (k-Nearest Neighbour) and Rule Learning (Simple Covering Algorithm). Most of these algorithms require nominal data representation except for Naïve Bayes and k-Nearest Neighbour. The e4ML is also focuses on supervised learning that is dependent on the class data to produce hypotheses or classification.

C. *User Interfaces and Simulation*

The interface for e4ML is extensively developed using Swing UI from Java for better graphical representation. The initial e4ML startup includes the file browsing, data summary and messages similar to the Weka classifier.

Figure 2.0 shows the main e4ML startup. The user has to input the respective data set and if the data can be processed, the information of the data will be displayed including its attributes and values.

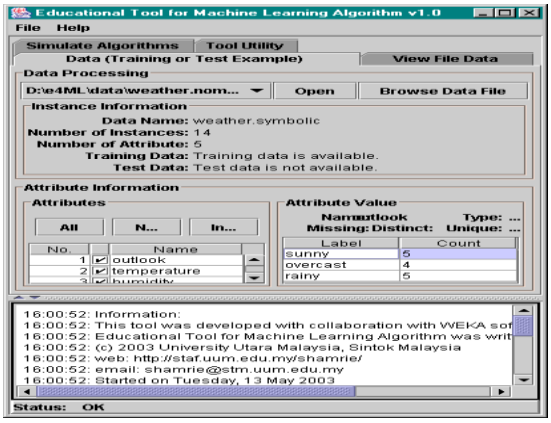

Figure 2.0: Main startup GUI tab for e4ML

The data can be viewed with arff file viewer and simple table representation but at the moment the data cannot be modified due to the current phase of the tool implementation and user has to modify the data entry directly from the arff file using any text editor. Figure 3.0 depicts the arff file viewer and figure 4.0 for table viewer tab.

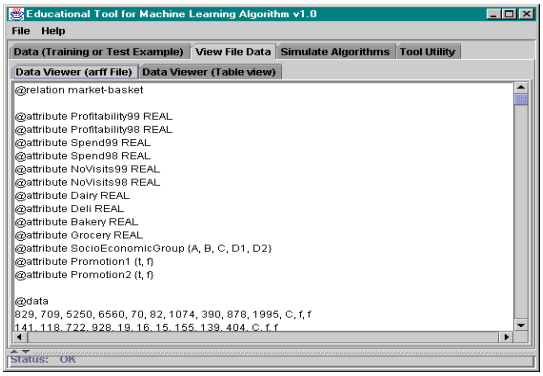

Figure 3.0: arff file Viewer

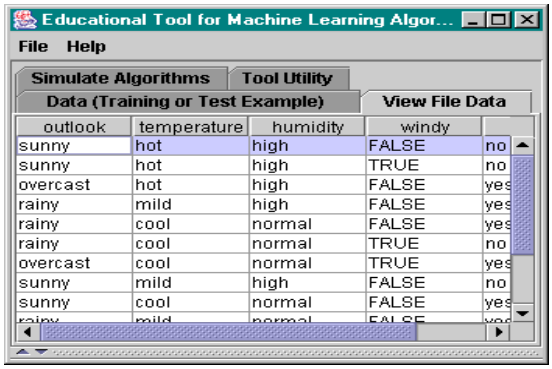

Figure 4.0: Table viewer

The next process in the e4ML is to select certain algorithm implementation to simulate the data that has been provided. Some of the algorithms will proceed if the data is represented in nominal only or other requirements by the algorithms. ID3 learning for Decision tree for example requires nominal data representation only and the data can be simulated using Decision Tree (ID3) algorithm. The data must contain class label for the purpose of supervised learning and classification because the current e4ML tool only support data with class label. Figure 5.0 shows the simulation tab screenshot.

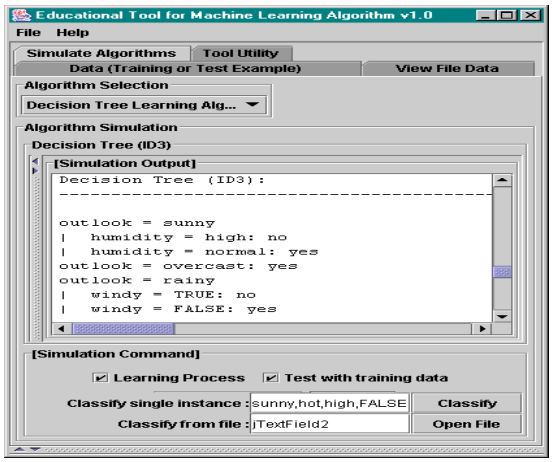

Figure 5.0: Simulation tab

In the simulation tab, output process including steps taken by the algorithms with certain hypothesis will be explained so user will understand the process that lead to the given hypothesis or decision tree rule for ID3. Testing or classifying new data instance is also provided but in the current e4ML tool, testing can be done with single instance only. Figure 6.0 shows the classifying new instance and its output.

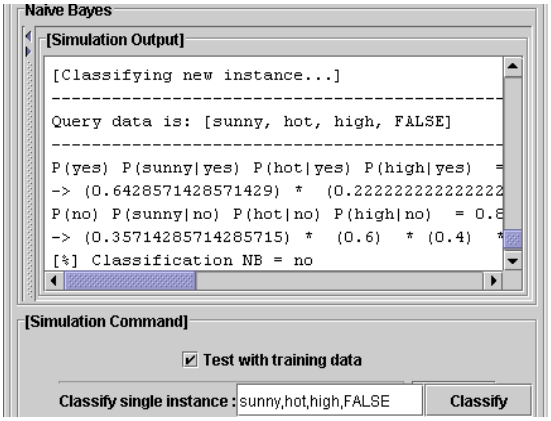

Figure 6.0: Classifying new single instance (Naïve Bayes)

The other tab for this tool is a simulation of computational learning theory (COLT) and other utility simulation such as concept learning version space, continuous value attribute and missing value solution. However, the implementation of the utility that currently provided by the tool is a computational learning theory for version space. COLT is a study of the mathematical power of computer programs. [19] addresses the influence of hypothesis space to the learnability setting of certain algorithms that deals with how computational theory could help when selecting some learning method. Figure 7.0 shows the tab for the utility. The purpose of this utility is to explain the number of training data that is sufficient to some algorithm to be learnable based on the hypothesis space computation.

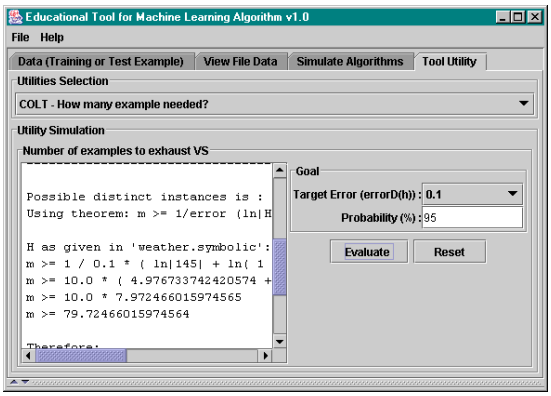

Figure 7.0: COLT utility

V CONCLUSION AND FUTURE WORK

Machine learning has many algorithms that behave differently according to given data set. It is inevitably difficult to explain what is happening to the data and therefore a tool to support teaching of machine learning course is preferable. e4ML is developed with an intention of making the learning of such a course is probably easier to understand. This is important since a beginner has to know how certain machine learning algorithm works and in what condition it can be used. Through simulation, the user can actually experiment how the data representation can affect the learning algorithms. In addition, e4ML serves as a basic work for future research and development mainly in data mining and machine learning.

The e4ML system is currently lack of many machine learning capabilities but will be enhanced in the future. The future enhancement includes the development of the GUIs, preprocessing stage, other machine learning algorithms and visualization utility.

ACKNOWLEDGEMENT

We would like to thank the School of Information Technology and Universiti Utara Malaysia for funding and supporting this project.

REFERENCES

[1] Mitchell, Tom M, 'Machine Learning', McGraw-Hill, 1997.

[2] J. Han, Y. Fu, K. Koperski, G. Melli, W. Wang, O. R. Zaïane, (1995). Knowledge Mining in Databases: An Integration of Machine Learning Methodologies with Database Technologies", Canadian Artificial Intelligence.

[3] U.M Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. (1995). Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press.

[4] G. Piatetsky-Shapiro and W. J. Fraeley. (1991). Knowledge Discovery in Databases. AAAI/MIT Press.

[5] Michel A. King, John F. Elder IV, et al. (1998). Evaluation of Fourteen Desktop Data Mining Tools. IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA.

[6] McQueen R.J. and Holmes G. (1998). User perceptions of machine learning. Proc Association of Information Systems Conference,edited by E.D. Hoadley and I. Benbasat, Maryland, Baltimore, pp 180-182. Association for Information Systems, Atlanta, GA.

[7] Stahl I. (2000). How Should We Teach Simulation?. Proceedings of the 32nd conference on Winter simulation 2000 Pages: 1602 - 1612 Series-Proceeding-Section-Article.

[8] Zakaria, A., Siraj, F., Ahdon, M.F., Mat Hussin, M. Z. A. (2002). PEAGENT: An Interactive Web Based Educational System Using Animated Pedagogical Agent. Proceedings of the International Conference on Artificial Intelligence in Engineering & Technology, Kota Kinabalu Sabah, Malaysia. pp: 344-348.

[9] Witten, I.H., Frank, E., 'Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations', Morgan Kaufmann, 1999.

[10] Y.H. Liao and C.T. Sun. (2001). An Educational Genetic Algorithms Learning Tool, to appear on *IEEE Transactions on Education,* (NSC88-2520-S-009-002).

[11] Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases. [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[12] Garner, S.R, 'WEKA: The Waikato Environment for Knowledge Analysis', In Proc. of the New Zealand Computer Science Research Students Conference, 1995, pp. 57--64.

[13] P. A. Flach and N. Lachiche. The Tertius System. (1999). http://www.cs.bris.ac.uk/Research/MachineLearning/Tertius/.

[14] Amélie Deltour, 'Tertius extension to Weka', Technical Report CSTR-01-001, Department of Computer Science, University of Bristol, 2001.

[15] P. Alcamo, F. Domenichini, F. Turini. An XML based environment in support of the overall KDD process, 2000. Proceedings of the Fourth International Conference on Flexible Query Answering Systems. Physica-Verlag Heidelberg New York, 413-424.

[16] Witten I.H., Paynter G.W., Frank E., Gutwin C. and Nevill-Manning C.G. (1999) "KEA: Practical automatic keyphrase extraction." Proc. DL '99, pp. 254-256. (Poster presentation.)

[17] Fischer, Simon and Klinkenberg, Ralf and Mierswa, Ingo and Ritthoff, Oliver, 'Yale: Yet Another Learning Environment – Tutorial', CI-136/02, Collaborative Research Center 531, University of Dortmund, ISSN 1433-3325, 2002.

[18] Peter A. Flach and Nicolas L., 'Confirmation-Guided Discovery of First-Order Rules with Tertius', Machine Learning. 42(1/2), 2001, pp. 61--95.

[19] John C, S. Jain, and M. Suraj, 'Control Structures in Hypothesis Spaces: The Influence on Learning', Expansion of the version in *Proceedings of the Third European Conference on Computational Learning Theory (EuroCOLT'97)*, Jerusalem, 1997.

[20] D. Zhang, (2000). Applying Machine Learning Algorithms in Software Development, the Proceedings of 2000 Monterey Workshop on Modeling Software System Structures, Santa Margherita Ligure, Italy, pp.275-285.

Table 1: Types of Learning Method (Classification based on [1] and [20]).

Learning Type	Learning Method	Target Function Generation	Bias	Algorithms
CL	Eager, Supervised	Conjunction of attributes constraints with version space search	$c \in H$	Find-S, Candidate Elimination
DT	Eager, Supervised	Decision tree with Information Gain search	Preference for small tree (Occam Razor)	ID3, C4.5
BL	Eager, Supervised	Bayes, Bayesian Network using probabilistic search	Minimum description length (MDL)	Bayes Optimal Classifier, Naïve Bayes, Gibbs
ILP	Eager, Supervised	If-Then rules using statistical and general-to-specific	Rule accuracy, FOIL-Gain	Covering Algorithm, One-R, FOIL
IBL	Lazy, Supervised	Not explicitly defined (using statistical reasoning)	Nearest	k-NN, Case-Based Reasoning
EL	Eager, Supervised	Bit string using simulated evolution	Fitness	Genetic Algorithm
ANN	Eager, Supervised	ANN with gradient descent search	Interpolation	Backpropogation
AL	Eager, Supervised	Horn clause with deductive reasoning	Horn clause set	Prolog-EBL
RL	Eager, Unsupervised	Control strategy, Policy with training episodes search	Action with maximum Q values	Q learning, TD learning