Extending WEKA Framework for Learning New Algorithms

Waikato Environment for Knowledge Analysis (WEKA) is a collection of state-of-the-art machine learning algorithms and data preprocessing tools. It is designed so that you can quickly try out existing methods on new datasets in flexible ways. It provides extensive support for the whole process of experimental data mining, including preparing the input data, evaluating learning schemes statistically, and visualizing the input data and the result preprocessing, clustering, classification, regression, visualization, and feature selection. All of WEKA techniques are predicted on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric, or nominal attributes and it also supports other type of attributes). The easiest way to use WEKA is a graphical user interface called the Explorer. The

All of WEKA techniques are predicted on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric, or nominal attributes and it also supports other type of attributes).

of learning. WEKA was developed at the University of Waikato in New Zealand and is an open source software issued under General Public License[2] written in java. It runs on almost any platform and has been tested under Linux, Windows, and Macintosh operating systems. Recently an article was published in CSI which showed application of WEKA in Bio-inspired algorithm[1]. The authors emphasized on MLP classifier using genetic algorithm and fuzzy logic. They gave information about the existing framework. In this article, we extend the existing framework of WEKA in which we can add new classifier and cluster and then can trend the dataset from new algorithms.

The key features of WEKA’s success are as follows:

1. It is open source and freely available;
2. It provides many different algorithms for data mining and machine learning;
3. It is platform-independent; and
4. It is up-to-date, with new algorithms being added as they appear in the research literate.

WEKA[3] supports several standard data mining tasks, more specifically, data other user interfaces to WEKA are Experimenter, KnowledgeFlow, and Simple CLI. The Experimenter gives access to all of its facilities using menu selection and form filling. The KnowledgeFlow provides an alternative to Explorer for showing how data flows through the system. It also allows the design and execution of configurations for streamed data processing. The Simple CLI is a command line interface for executing WEKA commands.

The main interface Explorer has several panels that give access to the main components of the workbench. The “Preprocess” panel has facilities for importing data from a database, a comma-separated values (CSV) file etc., and for preprocessing this data uses a so-called filtering algorithm. These filters can be used to transform the data (e.g. turning numeric attributes into discrete ones) and make it possible to delete instances and attributes according to specific criteria. The “Classify panel” enables the user to apply classification and regression algorithms (indiscriminately called classifiers in WEKA) to the resulting dataset; to estimate the accuracy of the resulting predictive model; and to visualize erroneous predictions, ROC curves, or the model itself (if the model is amenable to visualization, e.g. a decision tree). The “Associate panel” provides access to association rule learners that attempt to identify all important interrelationships between various attributes in the data. The “Cluster panel” gives access to the clustering techniques in WEKA, e.g. the simple k-means algorithm. There is also an implementation of the expectation maximization algorithm for learning a mixture of normal distributions. The next panel, “Select attributes”, provides algorithms for identifying the most

Fig. 1: Existing snapshot of WEKA

* Assistant Professor, computer applications in SATI Degree, Vidisha (MP)
** Assistant Professor, computer applications in SATI Degree, Vidisha (MP)
predictive attributes in a dataset. The last panel, “Visualize”, shows a scatter plot matrix, where individual scatter plots can be selected and enlarged and analyzed further using various selection operators. WEKA can handle a number of file formats, including the ever-popular CSV (which can be exported from any spreadsheet program). WEKA prefers, however, to work with ARFF files, which are basically CSV files with some header information tacked on.

Suppose we want to implement a special-purpose learning algorithm i.e. adding a new classifier or a cluster which is not included in existing WEKA GUI, want to investigate a new learning scheme, or want to learn more about the inner workings of an induction algorithm by actually programming it yourself then integrate new workspace in WEKA.

WEKA can be extended to include the elementary learning schemes for research and educational purposes. Fig. 1 shows the existing framework of WEKA. Now we represent the method to add a new classifier in WEKA, we follow the following steps:

1. Create a new folder in a window directory hierarchy.
   Ex. C:\SmWork\classifiers
2. To enable or disable dynamic class discovery, the relevant file to edit is GenericPropertiesCreator.props (GPC). This file can be obtained from the weka.jar or weka-src.jar archive. These files can be opened with an archive manager that can handle ZIP files and navigate to the weka/gui directory, where the GPC file is located. All that is required is to change the Use Dynamic property in this file from false to true (for enabling it) or the other way round (for disabling it). After changing the file, just place it in home directory. For generating the GOE file, we need to execute the following steps:
   Java weka.gui.GenericPropertiesCreator
3. Remove WEKA.JAR from the CLASSPATH.
4. Edit the GenericPropertiesCreator.props file in the home directory and set UseDynamic to false.
5. Add SmWork/classifiers in Generic PropertiesCreator.props and Generic ObjectEditor.props.
6. Run the command
   java –classpath c:\progra~1\weka-3-6\weka.jar;c:\SmWork\classifiers\weaka\ gui\GUIChooser

Now we can write our new java code, compile it, and then copy the class file into a specified folder. Fig. 2 shows snapshot of newly added classifier. Similarly, we can extend WEKA for cluster and association as well.

References