Module: Weka for Information Retrieval

Last Update: December 10, 2010

1 Module Name

Weka

2 Scope

This module stresses the methods of text classification used in information retrieval. We focus on the usage of Weka a data mining toolkits, in data processing with three classification algorithms: *Naive Bayes* [1], *k Nearest Neighbor* [2], and *Support Vector Machine* [3]) mentioned in the textbook [7]

3 Learning Objectives

After finish the exercises, students should be able to:

- Explain the basic methodology of the three classification algorithms
- Use weka and understand its ARFF data file format
- Use Weka to analyze data with the three classification algorithms

4 5S Characteristics of the Module

- **Streams**: Weka is a data processing tools. The input stream is a data file with format of ARFF; The output file includes different types of results, such as text file, graph.
- **Structures**: Weka is a collection of machine learning algorithms and data preprocessing tools.
- **Spaces**: Components in weka can be worked together for any objectives in data processing, while they can also independently work on specific goals such as evaluation of learning schemes.
- **Society**: Weka can be used by those who need to use data mining methods to analysis their data, and also can be expanded by people who design and implement new data mining algorithms.
- **Scenarios**: Weka is designed to provide the whole process of experimental data mining, from data preparing, learning schemes evaluation, to data visualization. Weka can also work as a specific tools in data process such as a text classification tool with support vector machine classifier.

5 Level of Effort Required(in-class and out-of-class time required for students

- In-class: 20 Minutes presentation and demonstration
- Out-of-class:
  1. 2-3 hours for reading
  2. Approximately 1 hour for exercises

6 Relationship with other Modules

Weka is a collection of data processing tools, including feature selection, classification, clustering, and visualization. It has close relationship with CLUTO and R and its results from classification and clustering can be used by NLTK and Wordnet.

7 Prerequisite Knowledge Required

The students need to know basic concept of text classification in information retrieval and usage of unix command line.

8 Introductory Remedial Instruction

Text Classification, also called text categorization, is a problem in information retrieval. The general goal of text classification is to assign a given object to topic(s) which it belongs to based on previously created rules, for example, to assign an email into 'spam' or 'non-spam' or to assign a book into 'fiction' or 'history'. An algorithm that implements classification, especially in a concrete implementation, is known as a *classifier* [4].

Classification normally refers to a *supervised* procedure, e.g. a procedure that learns to classify new instances based on learning from a *trainnng set* of instances that have been properly labeled manually with the correct classes. The corresponding *unsupervised* procedure is known as *clustering*, and involves grouping data into classes based on some measure of inherent similarity [4].

You can use weka for both classification and clustering purpose, but we only include classification in this module with respect to chapter 13, 14, and 15 in the textbook [7]. We recommend readers to CLUTO module for clustering methods.

9 Body of Knowledge

9.1 Text classification Algorithms

In this section, we review principals of three text classification algorithms mentioned in the text book [7]. We refer students to chapter 13, 14, and 15 for details of specific algorithm.

- **Naive Bayes text classification** is a supervised and probabilistic learning methods. It calculates the probability of a document d being in class c by the following formular. $P(t_k|c)$ is the conditional probability of term t_k occurring in a document of class c. $P(c)$ is the prior probability of a document occurring in class c.

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

  The goal of classification is to find the *best* class for the document. The best class in *naive bayes* classification is the most likely or *maximum a posteriori*(MAP) class c_{map}:

$$c_{map} = argmax_{c \in C} \widehat{P}(c|d) = argmax_{c \in C} \widehat{P}(t_k|c)$$

- **k Nearest Neighbor text classification** is an unsupervised and vector spaced classification method. kNN assigns the majority class of the k nearest neighbors to a test document based on the unprocessed training set. It is less efficient than *Naive Bayes* but can work well when the training set is large. k represents the number of neighbor that are used to classification. We choose k based on experience or knowledge with respect to the classification problem. Decision boundaries are usually based on the property of data. An example of decision boundaries is to use cosine similarities to compute a class's score as the below formular.

$$score(c, d) = \sum_{d' \in S_k(d)} I_c(d') cos(\vec{v}(d'), \vec{v}(d))$$

- **Support Vector Machine(SVM) text classification** is also a vector spaced classification method with a kind of large-margin classifier. The goal of SVM is to find a decision boundary between two classes that is maximally far from any point in the training data. The linear classifier is shown below:

$$f(\vec{x}) = sing(\vec{w}^T \vec{x'} + b)$$

  A value of -1 indicates one class, and a value of +1 the other class. $\vec{w}$ is decided by support vectors on the margin of decision hyperplane.

The **Header** of the ARFF file looks like the following:

```
% 1. Title: Iris Plants Database
%
% 2. Sources:
%      (a) Creator: R.A. Fisher
%      (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
%      (c) Date: July, 1988
%
@RELATION iris

@ATTRIBUTE sepallength  NUMERIC
@ATTRIBUTE sepalwidth   NUMERIC
@ATTRIBUTE petallength  NUMERIC
@ATTRIBUTE petalwidth   NUMERIC
@ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}
```

The **Data** of the ARFF file looks like the following:

```
@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
```

Figure 1: An example of ARFF file

9.2 Introduction to Weka [7, 5]

Weka was developed by the University of Waikato in New Zealand, and the name stands for *Waikato Environment for Knowledge Analysis*, with pronouncation of rhyme of *Mecca*. Weka is written in Java and distributed under the terms of the GNU General Public License. It is platform independent and has been tested under Linux, Windows, and Macintosh operating systems. It implement many different learning algorithms along with methods for pre- and postprocessing and for evaluation the result of learning schemes on any given dataset. You can preprocess a dataset, feed it into a learning scheme, and analyze the resulting classifier and its performance.

Before we start the tour of weka application, we first look into the file format that used by weka, which is called ARFF. ARFF files [6] have two distinct sections. The first section is the **Header** information, which is followed the **Data** information. The **Header** of the ARFF file contains the name of the relation, a list of the attributes, and their types. An example **Headers** and the **Data** of the ARFF file looks like in figure 1.

Weka provides a graphical user interface with three main sub-interface.

- **Explorer** gives access to all of its facilities using menu selection and form filling. For example, you can build a decision tree from an ARFF file; you can compare different text classification model by analyzing the output result from the same file. Explorer allow you to finish all the task by moving mouse to choose related options and adjust them quickly until they are applicable to your data.

- **Experimenter** enables Weka users to compare a variety of learning techniques automatically with different parameter settings on a corpus of

datasets, collect performance statistics, and perform significance test.

- **KnowledgeFlow** allows Weka users to design configurations for streamed data processing.

However, in this report, we focus on how to use weka in command line which is a preferred way running applications in IBM cloud. We will provide introduction of how to run weka through command line in 10

The three three mentioned text classification methods: *Naive Bayes*, *k Nearest Neighbor*, *Support Vector Machine(SVM)* are implemented by weka into corresponding corresponding classifier.

- *Naive Bayes classifier* is used the package of *weka.classifiers.bayes.NaiveBayes* in weka.
- *k Nearest Neighbor classifier* is used the package of *weka.classifier.IBK* in weka.
- *Support Vector machine* is used the package of *weka.classifiers.SMO* in weka

10 Exercises/Learning Activities

10.1 A Brief Tutorial to Weka

Installation is very simple for weka. You only need to download it from its homepage`http://www.cs.waikato.ac.nz/ml/weka/index.html` according to your machine's operating system, and then unzip.

To **Use** weka under command line, you have to set up several environment variables.

1. Set `WEKA_HOME` as your weka's home directory
2. Add `$WEKA_HOME`/weka.jar into the `CLASSPATH`

An example command to use a classifier is as follows:

java weka.classifiers.trees.J48 -t weather.arff

Task Description
The objective of the exercises is to apply above learning methods to a dataset and compare the performance of them by analyzing their output to learn more about the data and the learning methods. Here are four tasks in this exercises for each learning methods.
Students are required to submit their report on all the tasks, including any screenshot, source code, and graphs.

- **A:** Ran *weka.classifiers.bayes.NaiveBayes* on *iris* data, explain the results.
- **B:** Ran *weka.classifier.IBK* on *iris* data, explain the results.

- **C:** Ran *weka.classifiers.SMO* on *iris* data, explain the results.
- **D:** Make a table to record all the results, and describe your conclusion on three classifier.

11 Evaluation of Learning Outcomes

In the report of exercise, students need to show their understanding of basic concept of the classification algorithms mentioned previously, and demonstrate their ability to run weka with proper classifier and explain the meaning of results from weka.

12 Resources

Please refer to *Reference* section at the end of this report.

13 Glossary

i. **Classification:** Process to assign a given object into class(es).

ii. **Supervised Learning:** A machine learning methods when learning from a training set of instances that have been properly labeled manually with the correct classes.

iii. **Unsupervised Learning:** A machine learning methods when grouping data into classes based on some measure of inherent similarity.

iv. **Training Set:** A set of data with label.

v. **Classifier:** An implementation of a specific classification algorithms.

vi. **Rules:** A group of features used for classification.

vii. **Vector Space Classification:**Classification works on vector represented dataset.

viii. **Decision hyperplanes:** Boundaries that used by vector space classification methods for a test document. classification

14 Contributors

Authors: Team 5 at CS_5604 Information Storage and Retrieval

- Bhanu Peddi
- Huijun Xiong
- Noha Elsherbiny

Reviewer: Dr. Edward Fox

References

[1] Naive Bayes Classifier: http://en.wikipedia.org/wiki/Naive_Bayes_classifier.

[2] k Nearest Neighbour: http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm.

[3] Support Vector Machine: http://en.wikipedia.org/wiki/Support_vector_machine.

[4] Classification: http://en.wikipedia.org/wiki/Classification_(machine_learning).

[5] Weka: http://www.cs.waikato.ac.nz/ml/weka/index.html.

[6] ARFF file: http://weka.wikispaces.com/ARFF+(book+version)).

[7] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, CA, 2. edition, 2005.