Web Page Preprocessing, Learning and Classification

Objectives

Obtain experience in preprocessing raw data for Web mining, learning a Web page classifier from the training data and utilizing it for classifying the Web pages in the test data set.

What to do?

You are given a training data set containing a collection of Web pages from University of Texas and University of Washington, and a test data set containing a collection of Web pages from Cornell University. The pages in these two data sets are hand-classified into the following categories (i.e., classes): Student, Faculty and Course. You will learn a Web page classifier from the provided training data, and then use the classifier to classify the Web pages in the test data set. You can implement the classifier by yourself or you can use a tool like Weka (http://www.cs.waikato.ac.nz/ml/weka/) to learn a classifier.

Data Sets

The training data set consists of 468 Web pages, 232 of which are from a department at the University of Texas and 234 of which are from a department at the University of Washington. As mentioned earlier, each of the Web pages in the data set belongs to one of the three classes: student, faculty and course. The files are organized into a directory structure, one directory for each class. Each directory contains Web pages in plain text files. The file name of each page corresponds to its URL, where '/' was replaced with '^'. Note that the pages start with a MIME-header.

The test data set contains 206 Web pages from Cornell University. The files are also organized into three directories, one directory for each class. Both training and test data sets are available in the gzipped tar format on the course web site.

Steps

The project consists of three parts: (1) preprocessing the data; (2) learning a classifier from the training set and (3) testing the learning classifier on the test set and reporting the testing statistics.

For example, the first question that you may have for the data preprocessing is: how to represent a Web page with features, i.e., what attributes are you going to use for representing a Web page? A simple method for representing a Web page (or a text document in general) is the “bag-of-words” method, which uses a set of words to represent a document. How to choose the set of words to represent documents is an important issue. Generally speaking, we should choose the words that can distinguish Web pages of one class from the pages of other classes. Words that appear frequently in one class of pages but not frequently in the other classes of pages are more useful than words that appear frequently (or infrequently) in all categories. You may conduct the following steps in your data preprocessing: (1) extract words from each page; (2) remove stop words from the extracted words (optional); (3) replace a list of words with their stems (optional); (4) build an inverted index file for the training data set; (5) select a set of words for representing Web pages; and (6) generate the training and test data tables.

What to submit?

You should submit the following items:

1. The assignment report that describes your Web page classification system, the design of your programs, the inverted index file for the training data and the analysis of your design and implementation.
2. The programs for preprocessing the raw data, for learning a classifier from the training data and for testing the learning classifier on the test data.
3. A file called readme.txt where you give a tutorial on how to compile and run your programs.

How will you be graded?

The full mark for this assignment is 25. The following will play a crucial role in your grade for this assignment.

1. Correctness of programs for preprocessing raw data, learning a classifier and testing it on the test data.
2. Your assignment report that should include: the introduction, description of your Web page classification system, description of your implementation, analysis of the results and conclusion. In particular, your report should focus on how to preprocess the raw data, why a specific classification learning algorithm is chosen, how you learn a classifier from the training data and how the classification accuracy is on the test data. If a few classification learning algorithms have been chosen, which one can generate the best result in terms of the classification accuracy and please justify in detail why.
3. Your class presentation. It accounts for 5 marks.
4. Clarity of your programs (comments!) and functionality of your programs.
5. Ease of using the README to test your programs and results.
6. Your group competition mark. Your solution will be compared with the solutions from other groups according to the classification accuracy of your learning model on the test data set.
7. Student performance mark in your group. The student performance in your group is to evaluate your performance in the team work. The group-peer marking will come from the other team members of your group. That is, at the end of the project, each of you will be asked to rate the performance of other members in your group. The ratings on you by the other members of your group will determine your "performance mark". This is to encourage all the students to get involved in the project. It accounts for 5 marks.