CS 486/686 Introduction to Artificial Intelligence

Fall 2016

Assignment 3 Topic Classification Using Machine Learning Classifiers

Due Date: Monday November 21, 2016 at 5:00pm

This assignment is to be done individually. It is acceptable to share knowledge about Machine Learning algorithms and the WEKA workbench. However, you should do all experiments, interpretation of results, and write-ups on your own.

Purpose: This assignment is intended to further your understanding of learning-from-data using Machine Learning classification for a Natural Language Processing (NLP) task. You will compare several popular classifiers using the "WEKA" Machine Learning workbench.

Where to get help:

If you encounter problems at any step you can post to the Assignment 3 LEARN Discussion Forum, see the Instructor or TAs during their office hours, arrange an individual appointment by emailing the Instructor or TAs, or bring the problem to the Assignment 3 in-class Labs.

Details are given below about how to install and use WEKA. If you should encounter any problems with WEKA please contact edward.chrzanowski@waterloo.ca (our CSCF technical support).

Where to find the Assignment materials:

The Assignment will make use of the "WekaMOOC" videos from the University of Waikato to step you through gaining familiarity with using WEKA, and using its tools and features.
We also reference the WekaManual in the instructions throughout the assignment.

All materials for this Assignment (the assignment itself, sample dataset, WEKA manual, videos) can be found in the "ASSIGNMENT3-MATERIALS" folder in "Assignments" on our CS486/686 course LEARN site.

INTRODUCTION/OVERVIEW OF THE ASSIGNMENT

In this assignment, you will gain experience with an important Natural Language Processing (NLP) task: **Automated Text Categorization.** More precisely, you will test several classification algorithms (Naive Bayes, Support Vector Machine (SVMs), Decision Tree) on a corpus of financial new articles with the WEKA data mining package. Your task is to first download WEKA, experiment with various *pre-processing* techniques to transform the documents in the corpus into a *vector space representation*, and then experiment with the classification algorithms. These classification algorithms will be used to decide the main topic of each financial article, e.g., ship, wheat, corn, acquisitions.

Follow the steps below in order:

STEP 1: Introducing WEKA

a. View the Data Mining with Weka video: 1.1 Introduction) (9mins)
   - (Don't worry about mention of activities and evaluations by the speaker. You will be doing our own activities for this Assignment—these are spelled out below.)

STEP 2: Installing WEKA

a. View the Data Mining with Weka video: 1.2 Exploring the Explorer) (11mins)
    - Skip 3:45-5:00 (we will be using our own dataset)

b. Download WEKA from: http://www.cs.waikato.ac.nz/ml/weka/downloading.html
    - Note: WEKA 3.8 has been installed in the undergraduate Mac labs.

STEP 3: Exploring the "Explorer" (continuing video 1.2: Exploring the Explorer)

a. Launch WEKA and start "Explorer".
b. View the remainder of the video. Then go to STEP 4 below.

STEP 4: Exploring datasets

a. View the Data Mining with Weka video: 1.3 Exploring datasets (10:30mins)
b. We will be using our own dataset for this assignment. Our sample corpus consists of 9981 news articles published by Reuters in 1987 in 10 different categories (crude, acq, interest, grain, earn, money-fx, trade, wheat, corn, ship). This corpus is in the data file "Reuters21578-Apte-10Cat.arff" in the ASSIGNMENT3-MATERIALS folder. The file is already in the "arff" format which can be directly loaded into WEKA.
c. Read section 1.2.1 Dataset in the WekaManual.
d. Open up the file with a text editor to see what the news articles in arff format look like.

STEP 5: Loading the dataset

a. Copy the data file "Reuters21578-Apte-10Cat.arff" into a directory on your computer.
b. Load this corpus into WEKA by clicking on the "Preprocess" tab followed by "Open file" button, then selecting the Reuters data file from its directory on your computer.

STEP 6: Preprocessing: Converting the dataset into a vector space representation

a. Read the WekaManual intro (4.2.1, 4.2.2, 4.2.3) to section 4.2 Preprocessing.
b. Click on the button "Choose" under Filter.
c. Select "weka"→"filters"→"unsupervised"→"attribute"→ "StringToWordVector"

STEP 7: Using filters

a. Working with filters:
    i. Intro: Read section 1.2.3 weka.filters in the WekaManual.
    ii. Read section 4.2.4 in the WekaManual

b. Editing the properties of the filter "StringToWordVector":
    i. Right-click on the filter name "StringToWordVector" for a list of its properties.

ii. Click on the button "More" for a brief explanation of the filter and each property. (More details about terminology will be given in a supplement to the assignment.)

iii. Experiment*** with different values for the following properties:

- IDFTransform
- TFTransform
- lowerCaseTokens
- minTermFreq
- stemmer
- tokenizer
- useStopList

All other properties should remain with their default values.

iv. Once the properties are configured click on "Apply" to run the filter.

***By "experiment" we mean: change the values of the various properties of the filter, apply the filter, build and run the classifier (STEP 8 below), then compare your results given these new property values, e.g., differences in accuracy of the classifier, length of time to build the model (classifier), length of time to test the model on the data, etc.

The idea is that by pre-processing the data differently, the classifier will be using different features and will likely produce different results. This is an opportunity to "get your hands dirty" and see to what extent the pre-processing phase can influence the results.

*****Important: You should re-load the data file before applying a new configuration of properties for the filter.*****

STEP 8: Building a classifier

a. View these Data Mining with Weka videos:
   - 1.5 Using a filter (7:30mins)
   - 2.1 Be a classifier! (11mins)
   - 2.2 Training and testing (5:40mins)
   - 2.3 Repeated training and testing (6:50mins)

b. Click on the tab "Classify" to run a classifier.

c. **Click on the box below the Test options box to make sure the "Class" attribute is "@@class@@" otherwise some classifiers may not be available and the results would be wrong anyway.**
   From the WekaManual 4.3.3 The Class Attribute:
   "The classifiers in WEKA are designed to be trained to predict a single 'class' attribute, which is the target for prediction. Some classifiers can only learn nominal classes; others can only learn numeric classes (regression problems); still others can learn both. By default, the class is taken to be the last attribute in the data. If you want to train a classifier to predict a different attribute, click on the box below the Test options box to bring up a drop-down list of attributes to choose from.

d. Under the "Classifier" button click on "Choose" to find the menu of classifiers.

e. Experiment with the following classifiers (see the instructions below for the parameters of each classifier you should experiment with).
Click "Start" button on left hand side of the WEKA interface to run the classifier.
   i. Naive-Bayes ("weka"→"classifiers"→"bayes"→"naiveBayesMultinomial")
   ii. Support Vector Machine ("weka"→ "classifier"→ "functions"→"SMO")
   iii. Decision Trees ("weka"→classifier"→"J48".
   iv. Some reference videos:
      - Data Mining with Weka 3.1 Simplicity first! (8:20mins)
      - Data Mining with Weka 3.2 Overfitting (8:30mins)
      - Data Mining with Weka 3.4 Decision Trees (9:30mins)
      - Data Mining with Weka 4.5 Support Vector Machines (8:30mins)
      - More Data Mining with Weka 1.3 Comparing classifiers (7:50mins)
      - More Data Mining with Weka 2.4 Document classification (13mins)

*****Important: You should re-load the data file before applying a new configuration of properties for the filter.*****

f. Compare the results when you:
   i. Train and test with the same data (i.e., the original dataset) versus
   ii. Use a 10-fold cross validation.
   iii. Some reference videos:
      - Data Mining with Weka 2.5 Cross-validation (6:50mins)
      - Data Mining with Weka 2.6 Cross-validation results (7:15mins)

Submit your assignment electronically to the LEARN Assignment 3 Dropbox.

What to hand in:

1. **Preprocessing:** Describe in your own words how the properties IDFTransform, TFTransform, lowerCaseTokens, minTermFreq, stemmer, tokenizer and useStopList affect the filter StringToWordVector.
   a. For each property, indicate whether and how different values impact the results you obtain.
      - For example, "stop words" are little words like "the", "and", "an" that occur very frequently in a document but carry little meaning. A "stop list" of these words can be used during the pre-processing stage to filter them out prior to building the classifier—we might expect to see some effect on accuracy of the classifier and possibly other outcomes.
   b. Discuss which joint configuration of the properties is the most suitable for text categorization in your opinion.

2. **Classification:**
   After processing the corpus with the StringToWordVector based on the joint configuration of properties that you recommended, experiment with naiveBayesMultinomial, SMO (support vector machine) and J48 (decision tree). Discuss your results as follows:
   a. Report the results of your experiments for STEP 8 (e) above, e.g.,
      - Comparisons of accuracy for the different classification algorithms.
      - Comparisons of time taken to build the model (i.e., classifier).
      - Comparisons of time taken to process the data.
      - Other observations you might discover.
   b. How much ***overfitting*** (difference between accuracy based on 10-fold cross validation and testing with the training set) occurs for each classifier when using its default configuration? Briefly explain.
   c. **[BONUS QUESTION]** Describe in your own words the effect of the following parameters for J48 and SMO, and recommend a joint configuration of the parameters:
      - J48: unpruned, minNumObj
      - SMO: polyKernel→exponent, RBFKernel→gamma
3. Print the output of each classifier (naiveBayesMultinomial, J48, and SMO) for the parameterization that you recommended in the previous question. Highlight which classifier is best when comparing accuracy and time.

Note: SMO and J48 may be slow depending on the machine you use and parameterization selected. If it is taking too long to run them, reduce the size of the dataset by running the filter "unsupervised"→"resample". Use this filter to sample (without replacement) a subset of the corpus that allows you to run your experiments at a decent pace. Report the size of your sample in your assignment.

You can also use LibSVM instead of SMO since LibSVM is faster than SMO, however you may have to download LibSVM separately if it didn't come with WEKA.

Explanation of Terminology:

Vector Space Representation: By "vector space representation", or "vector space model", we mean that a document is converted into a vector of elements where each element represents the "weight" of a particular term (typically a word) in the document. A term may also be a "keyword", e.g., "investment" for a financial document, or even a phrase. If a certain term does not appear in a given document then a value of **zero** will be assigned for that term's entry in the vector. The *number of terms* in the vector is its *dimensionality*. If words are used as terms then the dimensionality of the vector will be the total number of words in the vocabulary of the document set. Typically therefore in text categorization problems the vectors involved will have *very high dimensionality* (thousands of terms) but can be very *sparse* (i.e., contain many zero values).

*****Additional explanations of terminology will be given in an assignment supplement*****