CS 445/545

Machine Learning

Winter 2016

Homework 3: SVMs and Feature Selection

Due Tuesday, February 14, 2016, 2pm

In this homework you will do experiments with linear SVMs and feature selection.

This assignment will require writing a number of scripts to pre-process the data and to create new versions of the training and test sets. For people not familiar with writing text-processing scripts, we will schedule a tutorial session on using Python for this.

Please follow the steps below. Read carefully!

- **Data processing:**
  - Download data from https://archive.ics.uci.edu/ml/datasets/Spambase
  - Put data into format needed for the SVM package you're using
  - Split data into ~ ½ training, ½ test (each should have approximately the same proportion of positive and negative examples as in the entire set).
  - Scale training data using standardization: Compute the mean and standard deviation of each feature (i.e., over a column in the training set). Then subtract the mean from each value and divide each value by the standard deviation. You can either write your own code or use a library function to do this, e.g., the "scale" function in scikit.learn in Python. See http://scikit-learn.org/stable/modules/preprocessing.html for details.
  - Scale test data: for each feature subtract mean and standard deviation of that feature, where the mean and standard deviation were calculated *for the training set, **not** the test set.*

- **Experiment 1:** Run your SVM learner on the training data, and test the resulting model on the test data. You can use the default value of the C parameter.
  - Use SVM^{light} (http://svmlight.joachims.org/) or any SVM package
  - Use linear kernel (default in SVM^{light})
  - Test the learned SVM model on test data. Report accuracy, precision, and recall.
  - Create an ROC curve for this SVM on the test data, using 200 or more evenly spaced thresholds. You can either write your own code to do this or use a library function to create the ROC curve (e.g., plot_roc.py in scikit.learn in Python).

- **Experiment 2:** Feature selection with linear SVM

  - Using your learned SVM model from Experiment 1:
    - Obtain weight vector **w**. (For SVM^{light}, see https://www.cs.cornell.edu/people/tj/svm_light/svm_light_faq.html)

  Select features:
  - For m = 2 to 57
    - Select the set of m features that have highest $|w_m|$
    - Train a linear SVM, SVM_m , on all the training data, only using these m features (see ***Note on Feature Selection** at the end of this document).
    - Test SVM_m on the test set (using the same m features) to obtain accuracy.
  - Plot accuracy vs. m

- **Experiment 3:** Random feature selection

  Same as Experiment 2, but for each m, select m features at random from the complete set. This is to see if using SVM weights for feature selection has any advantage over random.

What to include in your report:

- **Experiment 1:**
  - Which SVM package you used
  - Accuracy, Precision, and Recall on the test data, using learned model
  - ROC Curve

- **Experiment 2:**
  - Plot of accuracy (on test data) vs. m (number of features)
  - Discussion of what the top 5 features were (see https://archive.ics.uci.edu/ml/machine-learning-databases/spambase/spambase.names for details of features)
  - Discussion of the effects of feature selection (about 1 paragraph).

- **Experiment 3:**
  - Plot of accuracy (on test data) vs. m (number of features)
  - Discussion of results of random feature selection vs. SVM-weighted feature selection (about 1 paragraph).

What code to turn in:
- Your code / script for performing SVM-weighted feature selection in Experiment 2
- Your code / script for performing random feature selection in Experiment 3

How to turn it in (read carefully!):

- Send these items in electronic format to mm@pdx.edu by 2pm on the due date. No hard copy please!
- The report should be in pdf format and the code should be in plain-text format.

- Put "MACHINE LEARNING HW 3" in the subject line.

If there are any questions, don't hesitate to ask me or e-mail the class mailing list.
Policy on late homework: If you are having trouble completing the assignment on time for any reason, please see me before the due date to find out if you can get an extension. Any homework turned in late without an extension from me will have 5% of the grade subtracted for each day the assignment is late.

***Note on Feature Selection:**
Once you have selected the *m* features you are going to use, create new training and test data files that contain only those *m* features.

For example, each row in my original training data file for SVM_light, called "spam.train" looks like this:

1 1:x_1 2:x_2 3:x_3 ... 57:x_{57}

where x_1 is the value of the first feature, x_2 is the value of the second feature, etc. (The 1 at the beginning denotes that this example is positive, i.e., spam.)

Now, suppose you have selected $m = 3$ features, and say the ones you selected are x_3 , x_{27} , and x_{41}. What you need to do is to create a new file, called something like "spam3.train", where each row contains only these three features:

1 1:x_3 2:x_{27} 3:x_{41}

Thus, this example is now represented by three features instead of 57. (SVM_light makes you put an index before each feature; thus the "1:", "2:", "3:". This will be different for other SVM packages.)

You would do the same for the test file for this value of *m*—that is, get rid of all features on each row except for the ones you have selected. The file would be called something like "spam3.test".

Then you would train an SVM on "spam3.train" and test it on "spam3.test", and record the accuracy.

You need to do this for $m = 2, ..., 57$.