
B551 Homework 6

You will need the Python and data files contained
in hw6.zip, from the oncourse website.

Directions:
	The problems below will ask you to implement three machine learning
	strategies for a text classification problem on a Reuters news dataset.
	(see readme.txt for more information about the framework code).
	Type or write answers to written questions and hand them in, hard-copy, in class.

Submission instructions:
	When you are done, upload your classifiers.py (and any other Python files you changed)
	to OnCourse. Hand in your written answers in class.

	IMPORTANT: MAKE SURE YOUR ONCOURSE SUBMISSION GOES THROUGH! You may have to click
	"Submit" more than once. You should receive an e-mail from OnCourse
	confirming your submission -- if you do not receive this e-mail, then your
	submission probably has not gone through and you should re-submit or e-mail
	one of the AIs to ask if they can see your submission!

I.

	1. Implement the Naive Bayes classifier in the NaiveBayesClassifier class in classifiers.py
	 by implementing the train() and test_one() functions.
	 For training, use a uniform class prior (rather than performing ML on the class counts).
	 Learn the MAP estimates of each feature probability given a Beta prior of alpha=beta=2
	 (i.e., 1 virtual count for both positive and negative examples).

	 Your test_one() function should work correctly even for

nonuniform class priors and different
parameters for the Beta feature priors.

2. Evaluate your classifier, where the positive examples are given by the
'earn' topic, and the negative examples are given by the 'acq,crude,gold' topics.

Use the --trmax=N command line option to vary the number N of examples in the
training set by increments of 100 from N=100 to 1000. Plot the accuracy on both the training set and
test set against N. Include this plot in your answer document (it should contain
two learning curves, one for training set accuracy and the other for testing set
accuracy). What do you observe?

3. Repeat the process of question 2, but use the --fsize=F option to
classify.py to vary the number F of features (words) extracted from the
documents with F=5, 10, 20, and 50. Plot the accuracy on both the training
set and test set against F as you did before. What do you observe?

II.

1. Implement a decision tree learning algorithm in the DecisionTreeClassifier class
in classifiers.py, by implementing the decision_tree_learning() and choose_feature()
functions (train() and test_one() are already given). To pick attributes to split
on, you may use either the minimum error criterion given in class, or the information
theoretic criterion given in R&N p. 703-704. Do not perform pruning.

2. Same questions as I.2 using this classifier.

3. Same questions as I.3 using this classifier.

III.

1. Implement a third classifier. Implement your learner ThirdClassifier in classifier.py,

and you may optionally implement any new feature extraction techniques in
FeatureExtractor() in utils.py. Describe your new technique and your rationale for
choosing it.

Examples for possible techniques are decision tree pruning, boosting with
decision stumps, Bayesian networks. Or, you could tune your existing
learners.

Extra credit will be awarded for the classifier with the best results on a
new classification task. Note that you will not be tested on the same categories
as in questions I and II, so make sure your techniques work for different categories.
We will also award EC for creative implementations.