

CS 445/545
Machine Learning
Winter, 2009

Homework 2: SVMs and Ensemble Learning

Due Monday Feb. 16, 5:00pm.

For this homework you will:

- Use `svm_light`, a support-vector-machine package, to train SVM classifiers on the optdigits training set;
- Use C4.5, a decision-tree package, to train decision trees on the same training set
- Write code (in any language you choose) to implement bagging and boosting of decision trees for this training set
- Create ROC curves and measure ROC area-under-curve values to compare the performance of all these different methods.

Support vector machines

Download `svm_light` from <http://svmlight.joachims.org> . This is a set of C functions implementing an SVM package. See documentation for instructions on how to compile and run this package. You are free to use the matlab interface for this package if you want to.

Using the `optdigits` data, create training and test files for each of the 10 digits, 0–9, setting the class to 1 for instances of that digit, -1 for instances of other digits.

The format for `svm_light` is a list of examples, one per line. Each line begins with 1 or -1 (the binary class of the example) and then gives the features of the example as

1:feature1 2:feature2 ... N:featureN

Here is a sample line I created from the `optdigits` data (this should all be on one line):

```
-1 1:0 2:0 3:0 4:12 5:13 6:5 7:0 8:0 9:0 10:0 11:0 12:11 13:16 14:9 15:0 16:0  
17:0 18:0 19:3 20:15 21:16 22:6 23:0 24:0 25:0 26:7 27:15 28:16 29:16 30:2 31:0 32:0  
33:0 34:0 35:1 36:16 37:16 38:3 39:0 40:0 41:0 42:0 43:1 44:16 45:16 46:6 47:0 48:0  
49:0 50:0 51:1 52:16 53:16 54:6 55:0 56:0 57:0 58:0 59:0 60:11 61:16 62:10 63:0 64:0
```

Using `svm_light` with the linear kernel, create a model (i.e., support vector machine) for each digit from that digit’s version of the training data. Example for digit “4”:

```
svm_learn 4.train 4.model
```

The resulting SVM is stored in the file 4.model.

To classify digit 4's test data according to this model:

```
svm_classify 4.test 4.model 4model-on-4test.predictions
```

The file 4model-on-4test.predictions gives, for each value the value of the linear combination of support vectors on that example, before the `sgn` function is performed.

After training, you will have 10 models (SVMs). To determine the class of each test example, run the test example through all 10 models, and choose the model whose prediction for that example is highest.

Record the classification error (i.e., fraction of misclassified examples) on the complete test data. Give a confusion matrix n which you report for each example in the test data how many times it was classified respectively as each of the 10 digits.

Repeat for the polynomial kernel, the radial basis function kernel, and the sigmoid kernel (all with the default parameters), using the `-t` option (see documentation). For each kernel, record classification error and confusion matrix.

Decision Trees

Download c4.5, a decision-tree package written in C, from

<http://www2.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/c4.5/tutorial.html>

Read the documentation for information on the format of the data files.

Implement (in any language you choose) the Bagging algorithm and the AdaBoost algorithm to create ensemble classifiers using decision trees as base classifiers. You should choose one of the digits (one of the harder ones to classify) and the decision tree should classify examples as positive (instance of that digit) or negative.

For each algorithm, record the classification error and confusion matrix of the final ensemble classifier on the test data for $T = 1, 5, 10, 50, 100$, where T is the number of iterations of Bagging or Boosting. ($T = 1$ means only one decision tree is used.)

Additional Performance Measures

To be added.

Margins

To be added.

What to turn in

- A writeup giving all your results as described above, and a paragraph or two summarizing the results.
- Your *well-commented* code implementing Bagging and Boosting.

Send these items in **electronic format** to karan@pdx.edu by the due date. No hard-copy please!

This assignment has several parts to it, so start early. If there are any questions on this assignment, don't hesitate to ask!

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 be graded down one grade (e.g., "A" to "B").