CSE 446 Machine Learning, Winter 2015
Homework 3

Due: Friday, February 27, beginning of class

1 Fitting an SVM classifier by hand [35 Points]

Consider a dataset with 2 points in 1d:

$$(x_1 = 0, y_1 = -1) \text{ and } (x_2 = \sqrt{2}, y_2 = 1).$$

We will use a feature function $\phi(x) = [1, \sqrt{2}x, x^2]$ (This is equivalent to using a second order polynomial kernel). The max margin classifier has the form

$$\hat{w}, \hat{w_0} = \arg\min_{w,w_0} ||w||_2^2 \quad s.t. \tag{1}$$

$$y_1(w \cdot \phi(x_1) + w_0) \geq 1 \tag{2}$$

$$y_2(w \cdot \phi(x_2) + w_0) \geq 1 \tag{3}$$

1. *(6 Points)* Write down a vector that is parallel to the optimal vector $\hat{w}$. Hint: Recall from Figure 14.12 (page 500 in the Murphy text) that $\hat{w}$ is perpendicular to the decision boundary between the two points in the 3d feature space.

2. *(6 Points)* What is the value of the margin that is achieved by this $\hat{w}$? Hint: Recall that the margin is the distance from each support vector to the decision boundary. Hint 2: Think about the geometry of the points in feature space, and the vector between them.

3. *(6 Points)* Solve for $\hat{w}$, using the fact the margin is equal to $1/||\hat{w}||$.

4. *(6 Points)* Solve for $\hat{w_0}$ using your value for $\hat{w}$ and Equations 1 to 3. Hint: The points will be on the decision boundary, so the inequalities will be tight. A "tight inequality" is an inequality that is as strict as possible. For this problem, this means that plugging in these points will push the left-hand side of Equations 2 and 3 as close to 1 as possible.

5. *(6 Points)* Write down the form of the discriminant function $f(x) = \hat{w_0} + \hat{w} \cdot \phi(x)$ as an explicit function of x. Plot the 2 points in the dataset, along with $f(x)$ in a 2d plot. You may generate this plot by hand, or using a computational tool like Python.

Show your work.

2 Ensemble Methods [65 points]

Your goal of this homework is to implement and run experiments comparing two different ensemble methods: bagging and boosting, on two different base learners: stumps and ID3 decision tree. Our data was provided by UCI Machine Learning Repository and can be found in the Molecular Biology (Promoter Gene Sequences) Data Set. It has 106 instances and 57 features. We randomly split the data set into the training (71 instances) and test (35 instances) sets, which are both well balanced.

You can obtain the data from the provided `training.txt` and `test.txt` files. Each DNA sequence is represented by one line, with the first 57 characters (one of 'a', 'g', 'c' and 't') representing the sequence, and the last character (separated by a space from the sequence) indicating the class ('+' for promoter, '-' for non-promoter). Upon loading the data, you should convert all of these values to numerical equivalents by setting any values with a '+' label as a 1, and every data points with a '-' label as a 0. Additionally, you should come up with a good numerical scheme to assign to your sequence. My personal recommendation would be to do: 'a'=0, 't'=1, 'c'=2, 'g'=3.

Note: There is random sampling in this problem set. When reporting your accuracy, we suggest you to run the experiment 10 times and to take the average. Also, you will get different values from your fellow collaborators.

2.1 Base Learners

We will use two types of decision trees as our base learners: decision stumps and depth two trees. A decision stump is a decision tree of depth one (i.e., it branches on one attribute and makes decisions), while a depth two tree can test at most two attributes on each path from the root to a leaf. These trees should be learned with information gain as the splitting criterion. Because we will use them in Adaboost, you should also make sure that your learning algorithm is flexible enough to accommodate weighted data points.

You are welcome to implement the tree learning yourself, reuse code from assignment 1, or use a toolkit such as scikit-learn. However, you should understand all the choices being made in the learning approach.

1. (5 points) Briefly summarize how you will do the learning for both base classifiers, including all decisions you made.
2. (5 points) Write the equation your learning algorithm will use for computing information gain with weighted data, as is required for the Adaboost algorithm.

2.2 Bagging

Given a standard training set D of size N, bagging generates M new training sets D_m, each of the size of the training set , by sampling examples from D uniformly and with replacement. By sampling with replacement, it is likely that some examples will be repeated in each D_m. This kind of sample is known as a bootstrap sample. The M models are trained using the above M bootstrap samples and combined by voting.

1. (10 points) Implement the bagging algorithm on top of your stump learners using the training data in the text files. Use uniform weight distribution amongst your data

points. Run 100 iterations, where for each iteration you sample another set D_m to learn a new classifier to add to the previously trained ones. After each iteration, record your error on the test set. Plot the test error vs. the number of bagging iterations. Comment on this result.

2. (10 points) Repeat this using depth 2 trees.

3. (5 points) If the number of samples you take for your D_m is equal to the number of elements in your training set, then for large N the set D_m is expected to have 63.2% of the unique examples of D, the rest being duplicates. Demonstrate this empirically by repeating this process 100 times and averaging your ratio of unique examples to total examples over all the trials.

2.3 Adaboost

1. (20 points) Implement AdaBoost as it is described in the course slides for both stump learners and trees of depth = 2 and run it for 100 iterations. Plot your test error vs. iteration number.

2. (5 points) Which base learner performs better (or is it a tie)? Briefly explain why this result happened.

3. (5 points) Run your algorithm for different numbers of iterations to try to find the point at which starts to overfit. Briefly describe how you tested for overfitting and when, if at all, it started. Repeat for both decision stumps and depth two trees, and comment on any differences you see.