CS 5350/6350: Machine Learining Spring 2015

Homework 4

Handed out: Mar 9, 2015
Due date: Mar 30, 2015

General Instructions

- You are welcome to talk to other members of the class about the homework. I am more concerned that you understand the underlying concepts. However, you should write down (and program) your own solution. Please keep the class collaboration policy in mind.
- Feel free ask questions about the homework with the instructor or the TAs.
- Your written solutions should be brief and clear. Your assignment should be **no more than 10 pages**. X points will be deducted if your submission is X pages beyond the page limit.
- Handwritten solutions will not be accepted.
- The homework is due by midnight of the due date. Please submit the homework on Canvas.
- Some questions are marked **For 6350 students**. Students who are registered for CS 6350 should do these questions. Of course, if you are registered for CS 5350, you are welcome to do the question too, but you will not get any credit for it.

1 Margins

The margin of a set of points $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$ with respect to a hyperplane is defined as the distance of the closest point to the hyperplane $\mathbf{w} \cdot \mathbf{x} = \theta$. The margin γ is thus:

$$\gamma = \min_i \left| \frac{\mathbf{w} \cdot \mathbf{x}_i - \theta}{||\mathbf{w}||} \right|$$

Suppose our examples are points in $\{0,1\}^{20}$ (that is, 20 dimensional binary vectors). For all the questions below, we wish to learn the following concept in this space using examples:

$$f(\mathbf{x}) = \mathbf{x}_2 \vee \mathbf{x}_4 \vee \mathbf{x}_6 \vee \mathbf{x}_{10} \vee \mathbf{x}_{12} \vee \mathbf{x}_{14} \vee \mathbf{x}_{16} \vee \mathbf{x}_{18}.$$

The variables that are in the disjunction are referred to as relevant variables and the others as irrelevant.

1. [3 points] Represent f as a linear threshold function. That is, find $\mathbf{w}$ and θ such that $sgn(\mathbf{w}^T\mathbf{x} - \theta)$ is equivalent to f for all $\mathbf{x} \in \{0,1\}^{20}$.

2. [6 points] Consider a dataset D_1 that is generated as follows:

- Positive examples: All possible points that have **one relevant variable** and **six irrelevant variables** set to *one* and all others *zero*.
- Negative examples: All possible points that have **no relevant variables** and **six irrelevant variables** set to *one* and all others *zero*.

Compute the margin of D_1 with respect to your $\mathbf{w}$.

3. [6 points] Consider a dataset D_2 that is similar to D_1, except that for positive examples **six relevant variables** are set to one in addition to the six irrelevant variables. Compute the margin of D_2 with respect to your $\mathbf{w}$.

4. [6 points] Now, consider a dataset D_3 that is similar to D_1. The only difference is that the number of irrelevant variables that are seen in both positive and negative examples is increased to **ten**. Write the Perceptron mistake bound for D_1, D_2 and D_3.

5. [4 points] Rank the datasets in terms of "ease of learning". Justify your answer.

2 Boosting and Perceptron

Consider the following set of training examples (i is the index of the example):

i	x	y	label
1	1	10	-
2	4	4	-
3	8	7	+
4	5	6	-
5	3	16	-
6	7	7	+
7	10	14	+
8	4	2	-
9	4	10	+
10	8	8	-

In this problem you will use two learning algorithms, Boosting and Perceptron to learn a hidden Boolean function from this set of examples.

1. [10 points] Use two rounds of AdaBoost to learn a hypothesis for this Boolean data set. As weak learners use a hypothesis of the form $[x > \theta_x]$ or $[y > \theta_y]$, for some integer θ_x or θ_y. Each round, choose the weak learner that minimizes the error ϵ. There should be no need to try many θs, appropriate values should be clear from the data.

   Start the first round with a uniform distribution D_0. Place the value for D_0 for each example in the appropriate column of Table 1. Find the hypothesis given by the weak learner that minimizes the error ϵ for that distribution. Place this hypothesis as the heading to the fourth column of Table 1, and give its prediction for each example in that column.

Now compute D_1 for each example, and select hypothesis that minimizes error on this distribution, placing these values and predictions in the appropriate columns of Table 1.

		Boosting				**Perceptron updates**
		Hypothesis 1		Hypothesis 2		
i	Label	D_0		D_1		Start with $(0, 0)$
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						

Table 1: Table for Boosting and Perceptron

2. [5 points] Write the final hypothesis produced by AdaBoost.

3. [6 points] Use the Perceptron learning algorithm to train a hypothesis for this Boolean data set, using as features the weak learners chosen by the boosting algorithm in (a).

   Train the Perceptron one cycle through the data, using 0 as initial weights, threshold of 3, and learning rate of 1. Fill in the *weight vector* column of Table 1.

4. [4 points] Did the two algorithms converge to the same hypothesis? If both hypotheses were used to predict labels for the training set, would the set of predictions be the same? Explain.

3 Kernels

(a) [10 points] If $K_1(\mathbf{x}, \mathbf{z})$ and $K_2(\mathbf{x}, \mathbf{z})$ are both valid kernel functions, with positive α and β, prove that

$$K(\mathbf{x}, \mathbf{z}) = \alpha K_1(\mathbf{x}, \mathbf{z}) + \beta K_2(\mathbf{x}, \mathbf{z}) \tag{1}$$

is also a kernel function.

(b) [10 points] Given two examples $\mathbf{x} \in \Re^2$ and $\mathbf{z} \in \Re^2$, let

$$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T\mathbf{z})^3 + 9(\mathbf{x}^T\mathbf{z})^2 + 81\mathbf{x}^T\mathbf{z}. \tag{2}$$

Prove that this is a valid kernel function.

4 Learning Decision Lists (For CS 6350 students)

In this problem, we are going to learn the class of k-decision lists. A decision list is an ordered sequence of if-then-else statements. The sequence of if-then-else conditions are tested in order, and the answer associated to the first satisfied condition is output (see Figure 1).

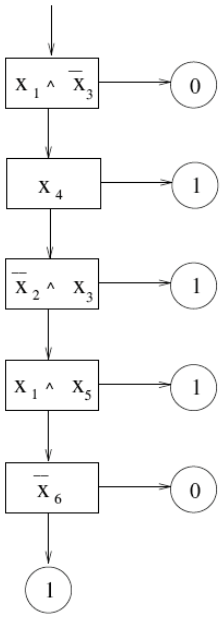

Figure 1: A 2-decision list.

A *k-decision list* over the variables $x_1, \ldots, x_n$ is an ordered sequence $L = (c_1, b_1), \ldots, (c_l, b_l)$ and a bit b, in which each c_i is a conjunction of at most k literals over $x_1, \ldots, x_n$. The bit b is called the *default* value, and b_i is referred to as the bit *associated* with condition c_i. For any input $x \in \{0,1\}^n$, $L(x)$ is defined to be the bit b_j, where j is the smallest index satisfying $c_j(x) = 1$; if no such index exists, then $L(x) = b$.

We denote by *k-DL* the class of concepts that can be represented by a k-decision list.

1. [8 points] Show that if a concept c can be represented as a k-decision list so can its complement, $\neg c$. You can show this by providing a k-decision list that represents $\neg c$, given $c = \{(c_1, b_1), \ldots, (c_l, b_l), b)$.

2. [9 points] Use Occam's Razor to show:
   For any constant $k \geq 1$, the class of k-decision lists is PAC-learnable.

3. [8 points] Show that 1-decision lists are a linearly separable functions. (Hint: Find a weight vector that will make the same predictions a given 1-decision list.)

The experiments question is removed