CSE 151: Introduction to Machine Learning Spring 2018

Problem Set 4

Instructor: Kamalika Chaudhuri **Due on:** never

Problem 1: 8 points

Alice, Bob and Carol have all been asked to implement the perceptron algorithm. They all have the same training and test data, and they make a single pass over the training and test data with all the algorithms (Alice's variant, Bob's variant, and Carol's variant). Carol implements the version of perceptron that we discussed in lecture.

1. Suppose Alice implements the following variant of the perceptron algorithm.

```
(a) Initially: $w_1 = 0$.
(b) For $t = 1, 2, 3, 4, \dots, T$
    i. If $y_t\langle w_t, x_t\rangle < 0$ then $w_{t+1} = w_t + y_t x_t$.
    ii. Otherwise: $w_{t+1} = w_t$.
(c) Output $w_{Alice} = w_{T+1}$.
```

   Is the test error of the classifier output by Alice's algorithm the same as the test error of Carol's algorithm, no matter what the test data is? If your answer is yes, justify your answer. If your answer is no, provide a counterexample or a brief justification.

2. Bob implements a second variant of the perceptron algorithm, as follows.

```
(a) Initially: $w_1 = 0$.
(b) For $t = 1, 2, 3, 4, \dots, T$
    i. If $y_t\langle w_t, x_t\rangle \leq 0$ then $w_{t+1} = w_t + y_t x_t/\|x_t\|$.
    ii. Otherwise: $w_{t+1} = w_t$.
(c) Output $w_{Bob} = w_{T+1}$.
```

   Is the test error of the classifier output by Bob's algorithm the same as the test error of Carol's algorithm, no matter what the test dataset is? If your answer is yes, provide a justification for your answer; if your answer is no, provide a counterexample or a brief justification.

Problem 2: 12 points

In this problem, we will formally examine how transforming the training data in simple ways can affect the performance of common classifiers. Transforming training features by scaling is equivalent to measuring these features in different units; in practice, we frequently have to combine multiple homogeneous or heterogeneous features, and it is important to understand how changing units in which features are measured can affect machine learning algorithms.

Suppose we are given a training data set $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ where each feature vector x_i lies in d-dimensional space. Suppose each $x_i = [x_i^1, x_i^2, \dots, x_i^d]$, so coordinate j of x_i is denoted by x_i^j.

For each x_i, suppose we transform it to z_i by rescaling each axis of the data by a fixed factor; that is, for every $i = 1, \dots, n$ and every coordinate $j = 1, \dots, d$, we write:

$$z_i^j = \alpha^j x_i^j$$

Here α^js are real, non-zero and positive constants. Thus, our original training set S is transformed after rescaling to a new training set $S' = \{(z_1, y_1), \ldots, (z_n, y_n)\}$. For example, if we have two features, and if $\alpha^1 = 3$, and $\alpha^2 = 2$, then, a feature vector $x = (x^1, x^2)$ gets transformed by rescaling to $z = (z^1, z^2) = (3x^1, 2x^2)$.

A classifier $C(x)$ in the original space (of x's) is said to be equal to a classifier $C'(z)$ in the rescaled space (of z's) if for every $x \in \mathbb{R}^d$, $C(x) = C'(z)$, where z is obtained by transforming x by recaling. In our previous example, the classifier C in the original space:

$$C(x) : \textit{Predict } 0 \textit{ if } x^1 \leq 1, \textit{ else predict } 1.$$

is equal to the classifier C' in the rescaled space:

$$C'(z)\textit{: Predict } 0 \textit{ if } z^1 \leq 3, \textit{ else predict } 1.$$

This is because if $C(x) = 0$ for an $x = (x^1, x^2)$, then $x^1 \leq 1$. This means that for the transformed vector $z = (z^1, z^2) = (3x^1, 2x^2)$, $z^1 = 3x^1 \leq 3$, and thus $C'(z) = 0$ as well. Similarly, if $C(x) = 1$, then $x^1 > 1$ and $z^1 > 3$ and thus $C(z) = 1$. Now, answer the following questions:

1. First, suppose that all the α^i values are equal; that is, $\alpha^1 = \ldots = \alpha^d$. Suppose we train a k-NN classifier C on S and a k-NN classifier C' on S'. Are these two classifiers equal? What if we trained C and C' on S and S' respectively using the ID3 Decision Tree algorithm? What if we trained C and C' on S and S' respectively using the Perceptron algorithm? If the classifiers are equal, provide a *brief* argument to justify why; if they are not equal, provide a counterexample.

2. Repeat your answers to the questions in part (1) when the α_is are different. Provide a *brief* justification for each answer if the classifiers are equal, and a counterexample if they are not.

3. From the results of parts (1) and (2), what can you conclude about how k-NN, decision trees and perceptrons behave under scaling transformations?

Problem 3: 10 points

We will now repeat Problem 2 for two forms of solutions to logistic regression.

1. First, suppose that we solve logistic regression exactly to get the solution w^*; in other words, w^* exactly minimizes the logistic regression loss function. Doing this on S gives us classifier C and doing it on S' gives us C'. Are C and C' equal when all the α^i's are equal? What if they are different? Justify your answer.

2. Now, suppose that instead of running gradient descent until we find the exact minimum, we run 1 step of gradient descent starting with an initial point $w^{(0)} = 0$. Doing this on S gives us classifier C and doing it on S' gives us C'. Are C and C' equal when all the α^i's are equal? What if they are different? Justify your answer.

3. Are your answers the same for parts (a) and (b)? If not, what do you think accounts for the difference?

Problem 4: 12 points

Suppose we are given a training set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. Write down the gradient and the gradient descent updates for the following loss functions. In each case, what is the time required to compute a single gradient descent update?

1. $L(w) = \sum_{i=1}^{n} \max(0, 1 - y_i w^\top x_i)^2$.

2. $L(w) = \sum_{i=1}^{n} (y_i - w^\top x_i)^2$.

3. $L(w) = \sum_{i=1}^{n} \log(1 + e^{-y_i w^\top x_i}) + \lambda \|w\|^2$.