CSE 151: Introduction to Machine Learning Spring 2014

Problem Set 4

Instructor: Kamalika Chaudhuri **Due on:** May 15, 2015

Instructions

- This is a 40 point homework.
- Homeworks will graded based on content and clarity. Please show your work *clearly* for full credit.
- For Problem 3, you are free to use any programming language that you wish. Please email a copy of your code to cse151homeworks@gmail.com. **Please do not send the solution to Problem 3 to this email addres – instead, write down this solution on your physical HW.**

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 $t = 1, 2, 3, 4, \ldots, T$

      i. If $y_t \langle w_t, x_t \rangle \leq 0$ then $w_{t+1} = w_t + y_t x_t$.

      ii. Otherwise: $w_{t+1} = w_t$.

   (c) Output $w_{Alice} = w_{T+1}/\|w_{T+1}\|$.

   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 $t = 1, 2, 3, 4, \ldots, T$

      i. If $y_t \langle w_t, x_t \rangle \leq 0$ then $w_{t+1} = \frac{w_t + y_t x_t}{\|w_t + y_t x_t\|}$.

      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. Understanding the effect of transformations is important in practice, where we frequently have to combine multiple heterogeneous features.

Suppose we are given a training data set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where each feature vector x_i lies in d-dimensional space. Suppose each $x_i = [x_i^1, x_i^2, \ldots, x_i^d]$, 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 $i = 1, \ldots, n$ and every coordinate $j = 1, \ldots, d$, 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: Programming Assignment: 20 points

In this problem, we look at the task of classifying images of digits again, but this time we will use perceptron instead of k-nearest neighbor classification in Homework 2. Download the files `hw4atrain.txt` and `hw4atest.txt` from the class website. These files contain your training and test data sets respectively.

1. First, we will classify images of 0 vs. 6, which the k-nearest neighbor algorithm found difficult to tease apart in homework 2. For your benefit, we have already converted the images into vectors of pixel colors; each pixel color is a value between 0 and 255. The data files are in ASCII text format, and each line of the files contains a feature vector of 784 features, followed by its label (0 or 6). The coordinates of the feature vector are separated by spaces.

   Assume that the data is linearly separable by a hyperplane through the origin. Run one, two and three passes of perceptron, voted perceptron, and averaged perceptron on the training dataset to find classifiers that separate the two classes. What are the training errors and the test errors of perceptron, voted perceptron and averaged perceptron after one, two and three passes?

2. For the second part of the question, download `hw4btrain.txt` and `hw4btest.txt`. This will be your training and test data respectively.

   For each class $i = 0, \ldots, 9$, run a single pass of the perceptron algorithm on the training dataset to compute a linear classifier separating the training data points in class i from the training data points not in class i. Call this classifier C_i. We will now use these classifiers to construct a *one-vs-all* multiclass classifier.

   Given a test example x, the one-vs-all classifier predicts as follows. If $C_i(x) = i$ for exactly one $i = 0, \ldots, 9$, then predict label i. If $C_i(x) = i$ for more than one i in $0, \ldots, 9$, or if $C_i(x) = i$ for no i, then report *Don't Know*.

   Recall from Homework 2 that the confusion matrix is a 10×10 matrix, where each row is labelled $0, \ldots, 9$ and each column is labelled $0, \ldots, 9$. The entry of the matrix at row i and column j is C_{ij}/N_j where C_{ij} is the number of test examples that have label j but are classified as label i by the classifier, and N_j is the number of test examples that have label j. Since the one-vs-all classifier can also predict *Don't Know*, the confusion matrix will now be an 11×10 matrix – that is, it will have an extra row corresponding to the *Don't Know* predictions.

   Write down the confusion matrix for the one-vs-all classifier on the training data in `hw4btrain.txt` based on the test data in `hw4btest.txt`. Looking at this confusion matrix, and the solutions of Homework 2, what can you say about the performance of perceptron compared with the 3-nearest neighbor classifier on this dataset?