CSE 151: Introduction to Machine Learning Winter 2017

Problem Set 4

Instructor: Kamalika Chaudhuri **Due on:** Feb 24, 2017

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 $y_t \langle w_t, x_t \rangle < 0$ 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 $t = 1, 2, 3, 4, \ldots, T$

      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 $S' = \{(z_1, y_1), \dots, (z_n, y_n)\}$. 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:

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

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 by topic posts made in six different internet newsgroups – comp.windows.x, rec.sport.baseball, sci.med, misc.forsale, talk.politics.mideast and talk.religion.misc – that correspond to labels $1, \dots, 6$ respectively.

For your convenience, we have already pre-processed the posts and converted them to feature vectors, where each feature or coordinate corresponds to the count of a single word. Download the files `hw4train.txt` and `hw4test.txt` from the class website. These files contain your training and test data sets respectively. Each line of the training or test set is a feature vector of length 819, followed by a label $(1, \dots, 6)$.

A dictionary is also provided in the file `hw4dictionary.txt`; the first line in the dictionary is the word that corresponds to the first coordinate, the second line to the second coordinate, and so on.

1. First, we will learn a linear classifier that can predict if a post belongs to class 1 or class 2. For this purpose, your training data is the subset of `hw4train.txt` that has label 1 or 2, and your test data is the subset of `hw4test.txt` that has label 1 or 2.

   Assume that data is linearly separable by a hyperplane through the origin. Run two, three and four 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 two, three and four passes? [Hint: If your code is correct, the training error after a single pass of perceptron, voted perceptron and averaged perceptron would be about $0.04, 0.07, 0.08$.]

2. Consider the averaged perceptron classifier w_{avg} that you built by running three passes on the data. We will now try to interpret this classifier.

   Find the three coordinates in w_{avg} with the highest and lowest values. What are the words (from `hw4dictionary.txt`) that correspond to these coordinates? The three highest coordinates are those words whose presence indicates the positive class most strongly, and the three lowest coordinates are those words whose presence indicates the negative class most strongly.

3. For the third part of the question, we will build a one-vs-all multi-class classifier with a *Don't Know* option.

   For each class $i = 1, \dots, 6$, 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 $i = 1, \dots, 6$, then predict label i. If $C_i(x) = i$ for more than one i in $1, \dots, 6$, or if $C_i(x) = i$ for no i, then report *Don't Know*.

   We will build a confusion matrix, that indicates how well a multiclass classifier can distinguish between classes. Recall from lecture that a confusion matrix is a 6×6 matrix, where each row is labelled $1, \dots, 6$ and each column is labelled $1, \dots, 6$. 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 7×6 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 `hw4train.txt` based on the test data in `hw4test.txt`.

   Looking at the confusion matrix, what are the i and j in the following statements?

   (a) The perceptron classifier has the highest accuracy for examples that belong to class i.

   (b) The perceptron classifier has the least accuracy for examples that belong to class i.

   (c) The perceptron classifier most often mistakenly classifies an example in class j as belonging to class i, for $i, j \in \{1, 2, 3, 4, 5, 6\}$ (i.e., excluding *Don't Know*).