CSE 151: Introduction to Machine Learning Winter 2021

Programming Assignment 3

Instructor: Kamalika Chaudhuri **Due on:**

Instructions

- This is a 20 point homework. The assignment should be done individually.
- You are free to use any programming language that you wish.
- The programming assignment should be submitted as a single pdf file through Gradescope. Please enter your answers first, followed by the code.

Problem 1: 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 $1, \ldots, 6$ 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 `pa3train.txt` and `pa3test.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 $(1, \ldots, 6)$.

A dictionary is also provided in the file `pa3dictionary.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 with perceptron that can predict if a post belongs to class 1 or class 2. For this purpose, your training data is the subset of `pa3train.txt` that has label 1 or 2, and your test data is the subset of `pa3test.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 on the training dataset to find classifiers that separate the two classes. What are the training errors and the test errors of perceptron after two, three and four passes? [Hint: If your code is correct, the training error after a single pass of perceptron would be about 0.04.]

2. We will again learn a linear classifier that predicts if a post belongs to class 1 or class 2, but this time it will be with logistic regression. For this purpose, your training data is the subset of `pa3train.txt` that has label 1 or 2, and your test data is the subset of `pa3test.txt` that has label 1 or 2.

   Again, the classifier is a hyperplane through the origin. Starting with the initial point w_0 set to the all zeros vector, run 10, 50 and 100 iterations of gradient descent on the following logistic regression loss function with learning rate $\eta = 0.001$:

   $$L(w) = \sum_{i=1}^{n} \log\left(1 + e^{-y_i \mathbf{w}^T \mathbf{x}_i}\right),$$

   where $\{\mathbf{x}_i, y_i\}_{i=1}^n$ is the dataset, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$ and $\mathbf{w} \in \mathbb{R}^d$ is the parameter vector. What are the training and test errors of logistic regression after 10, 50 and 100 iterations of gradient descent? [Hint: If your code is correct, the training error after two gradient descent steps would be about 0.497.]

3. Consider the perceptron classifier w that you built by running three passes on the data. We will now try to interpret this classifier.

Find the three coordinates in w with the highest and lowest values. What are the words (from `pa3dictionary.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.

4. Repeat Part (3) of the question on the logistic regression classifier that you got after 50 iterations of gradient descent in part (2).

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

   For each class $i = 1, \ldots, 6$, 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 $i = 1, \ldots, 6$, then predict label i. If $C_i(x) = i$ for more than one i in $1, \ldots, 6$, 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 $1, \ldots, 6$ and each column is labelled $1, \ldots, 6$. 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 `pa3train.txt` based on the test data in `pa3test.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*).