ELEN 4903: Machine Learning

Columbia University, Spring 2016

Homework 3: Due March 25, 2016 by 11:59pm

Please read these instructions to ensure you receive full credit on your homework. Submit the written portion of your homework as a *single* PDF file through Courseworks (less than 5MB). In addition to your PDF write-up, submit all code written by you in their original extensions through Courseworks (e.g., .m, .r, .py, etc.). Any coding language is acceptable. Do not wrap your files in .rar, .zip, .tar and do not submit your write-up in .doc or other file type. Your grade will be based on the contents of *one* PDF file and the original source code. Additional files will be ignored. We will not run your code, so everything you are asked to show should be put in the PDF file. Show all work for full credit.

Late submission policy: Late homeworks will have 0.1% deducted from the final grade for each minute late. *Your homework submission time will be based on the time of your **last** submission to Courseworks. I will not revert to an earlier submission!* Therefore, do not re-submit after midnight on the due date unless you are confident the new submission is significantly better to overcompensate for the points lost. Submission time is non-negotiable and will be based on the time you submitted your last file to Courseworks. The number of points deducted will be rounded to the nearest integer.

Problem 1 (boosting) – 120 points total

This homework will focus on boosting. You will boost two classifiers: (1) The Bayes classifier with *shared* covariance, and (2) the logistic regression classifier learned "online" similar to the Perceptron. The version of AdaBoost you will implement involves bootstrap sampling as discussed in class. The general form of boosting you should implement is given below.

Algorithm: AdaBoost (with sampling)

```
Given labeled data $(x_1, y_1), \dots, (x_n, y_n)$, where $y \in \{-1, +1\}$. Set $p_1(i) = \frac{1}{n}$ for $i = 1, \dots, n$
• For $t = 1, \dots, T$
    1. Sample a bootstrap data set $\mathcal{B}_t$ of size $n$ using the distribution $p_t$ on $x_1, \dots, x_n$
    2. Learn a classifier $f_t$ on data in $\mathcal{B}_t$. (Treat duplicated data as if they are different observations.)
    3. Set $\epsilon_t = \sum_{i=1}^n p_t(i)\mathbb{1}\{y_i \neq f_t(x_i)\}$ and $\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$. (Use the original data for this.)
    4. Update the weights: $\tilde{p}_{t+1}(i) = p_t(i)\exp\{-\alpha_t y_i f_t(x_i)\}$.
    5. Normalize to obtain a probability: $p_{t+1}(i) = \tilde{p}_{t+1}(i) / \sum_{j=1}^n \tilde{p}_{t+1}(j)$.
• Define the classification rule for a vector $x_0$ to be
    $$f_{boost}(x_0) = \text{sign}\left(\sum_{t=1}^T \alpha_t f_t(x_0)\right).$$
```

Data

We will use the breast cancer data set from the University of Wisconsin Hospitals located on the UCI Machine Learning Repository.[1] The data I provide you has been preprocessed, so you must use the data posted on either Courseworks or the class website. The data consists of 10-dimensional observations (including a dimension fixed to 1), and their corresponding labels, $+1$ indicating cancer and -1 indicating no cancer. There are 683 observations in total.

For experiments in Parts 2 and 3 below, set aside the first 183 observations as a testing set and use the remaining 500 observations for training the boosted classifier.

- Part 1 (20 points)

  Write a function that samples a discrete random variable. You will use this to implement Step 1 of the boosting algorithm given above. The function should take in a positive integer n and a k-dimensional probability distribution w. It should return a $1 \times n$ vector c, where each $c_i \in \{1, \ldots, k\}$ and $\text{Prob}(c_i = j|w) = w(j)$. The entries of c should be independently generated. For the distribution $w = [0.1,\ 0.2,\ 0.3,\ 0.4]$, show the histogram of one sampled vector c when $n = 50, 250, 500$.

  Hint: The cumulative distribution function (CDF) of w, and n uniform random variables will be useful.

- Part 2 (50 points)

  In this part you will boost the Bayes classifier with a shared covariance. Recall that we can write this as a linear classifier, where the prediction for x is $y = f(x) = \text{sign}(w_0 + x^T w)$. For the Bayes classifier, $f(x)$ is equal to

$$\ln \frac{p(x|y=1)p(y=1)}{p(x|y=0)p(y=0)} = \underbrace{\ln \frac{\pi_1}{\pi_0} - \frac{1}{2}(\mu_1 + \mu_0)^T \Sigma^{-1}(\mu_1 - \mu_0)}_{= w_0} + x^T \underbrace{\Sigma^{-1}(\mu_1 - \mu_0)}_{= w}.$$

  To make the notation easier to read, I've written the -1 class as a 0 class, but the data is labeled ± 1.

  For $T = 1000$ iterations of boosting, do the following:

  1. Implement a boosted version of this Bayes classifier, where class-specific π and μ, and shared Σ are learned on the bootstrap set $\mathcal{B}_t$. Notice that you only need to store w_0 and w for this problem, as written in the equation above. Since the data already contains a bias dimension equal to 1, you can store a single "augmented" vector where w_0 and w are combined. (When calculating μ_1, μ_0 and Σ, make sure you don't use this extra dimension!)
  2. On a single plot, show the training and testing error as a function of iteration t for $t = 1, \ldots, T$.
  3. What is the testing accuracy for this Bayes classifier *without* boosting?
  4. Plot α_t and ϵ_t as a function of t on different plots.
  5. Pick 3 data points and plot their corresponding $p_t(i)$ as a function of t on the same plot. Select the points such that there is some variation in these values.

[1] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)

- Part 3 (50 points)

  In this part you will perform essentially the same experiments as in Part 2, but with a different classifier. We will focus on an online version of the logistic regression classifier that is similar to the Perceptron.

  Comment: For the classifier below, I am including the offset w_0 within the classification vector w. Therefore, you should use the all 10-dimensions of x in the following algorithm and $w \in \mathbb{R}^{10}$.

```
Algorithm: Logistic regression for round $t$ of boosting (using online learning)

  Input: A bootstrapped data set $\mathcal{B}_t$ and step size $\eta \in (0, 1]$   (e.g., $\eta = 0.1$)

    1. Initialize the vector $w^{(0)} = \vec{0}$
    2. For step $i = 1, \dots, n$ do
         • In the next line, let $(y_i, x_i)$ be the $i$th pair in $\mathcal{B}_t$ and not the $i$th pair in the original data.
         • Update $w^{(i)} = w^{(i-1)} + \eta\{1 - \sigma(y_i x_i^T w^{(i-1)})\} y_i x_i$,   where   $\sigma(y_i x_i^T w) = 1/(1 + e^{-y_i x_i^T w})$
    3. Return the classifier for boosting round $t$, $w_t = w^{(n)}$.

       Comment: It's important that the data in $\mathcal{B}_t$ is randomly permuted so that the $+1$ and $-1$ observations are mixed up together. Your code from Part 1 will most likely do this automatically.
```

  For $t = 1, \dots, 1000$ iterations of boosting, do the following:

  1. Implement the online logistic regression classifier described above.
  2. On a single plot, show the training and testing error as a function of iteration t for $t = 1, \dots, T$.
  3. What is the testing accuracy of the logistic regression model *without* boosting? (You can use the two-class version of your logistic regression code from Homework 2, or your own implementation of binary logistic regression to do this.)
  4. Plot α_t and ϵ_t as a function of t on different plots.
  5. Pick 3 data points and plot their corresponding $p_t(i)$ as a function of t on the same plot. Select the points such that there is some variation in these values.