ISYE 6740 Homework 5
Summer 2020

Total 100 points.

Arjun Singh

July 19, 2020

1. **AdaBoost.** (25 points)

   Consider the following dataset, plotting in Figure 1. The first two coordinates represent the value of two features, and the last coordinate is the binary label of the data.

   $$X_1 = (-1, 0, +), X_2 = (-0.5, 0.5, +), X_3 = (0, 1, -), X_4 = (0.5, 1, -),$$
   $$X_5 = (1, 0, +), X_6 = (1, -1, +), X_7 = (0, -1, -), X_8 = (0, 0, -).$$

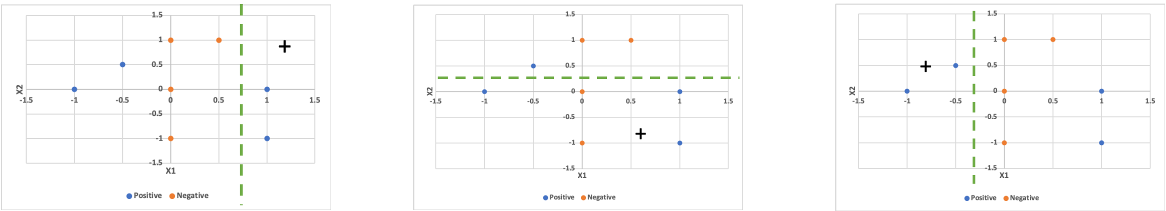
   In this problem, you will run through $T = 3$ iterations of AdaBoost with decision stumps (axis-aligned half planes) as weak learners.

   (a) (15 points) For each iteration $t = 1, 2, 3$, compute ϵ_t, α_t, Z_t, D_t by hand (i.e., show all the calculation steps) and draw the decision stumps on Figure 1. Recall that Z_t is the normalization factor to ensure that the weights D_t sum to one. (*Hint: At each iteration, you may specify a reasonable decision rule h_t as you would like, e.g., the decision stump in our lecture.*)

   - Answer:

     For iteration 1, I decided to place the stump at $X1 = 0.75$. Anything to the right of the stump was classified as positive and to the left was classified as negative. Similarly, stump 2 was chosen at $X2 = 0.25$ with the above the stump being negative and below the stump being positive. Lastly, stump 3 was chosen at $X1 = -0.25$ with the left being positive and right being negative. The decision stumps can be seen in Figure 1 along with a + sign denote the positive area.

     For iteration 1, $D_1 = \frac{1}{m} = \frac{1}{8}$ since all points were assigned the same weight. Using the first decision stump, there were only 2 misclassified points. Hence, $\epsilon = D_1(1) + D_2(2) = 0.25$.

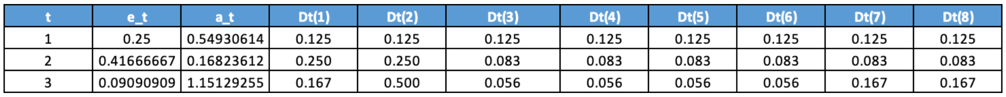


(a) Iteration 1

(b) Iteration 2

(c) Iteration 3

Figure 1: AdaBoost Iterations



t	e_t	a_t	Dt(1)	Dt(2)	Dt(3)	Dt(4)	Dt(5)	Dt(6)	Dt(7)	Dt(8)
1	0.25	0.54930614	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
2	0.41666667	0.16823612	0.250	0.250	0.083	0.083	0.083	0.083	0.083	0.083
3	0.09090909	1.15129255	0.167	0.500	0.056	0.056	0.056	0.056	0.167	0.167

(a) Values after 3 iterations

Figure 2: AdaBoost Results

After calculating ϵ, α was calculated using

$$\alpha = \frac{1}{2}\ln\left(\frac{1-\epsilon}{\epsilon}\right)$$

$$\alpha = \frac{1}{2}\ln\left(\frac{1-0.25}{0.25}\right) = 0.5493$$

D_2 for all points was calculated using

$$D_2(i) = D_1(i)\exp(-\alpha_t y_i h_i)$$

Lastly, D_2 for all points was normalized by dividing by $\sum D_2$. These steps were repeated for the remaining iterations. The results can be seen in Figure 2. The calculations for each iteration can be viewed in the accompanying spreadsheet.

(b) (10 points) What is the training error of AdaBoost? Give a one-sentence reason for why AdaBoost outperforms a single decision stump.

- Answer:
  AdaBoost outperforms a single decision stump because it combines multiple weak learners into a single strong classifier. The training error of AdaBoost decreases with the addition of multiple stumps because the weights give more importance to stumps that have more correct predictions and less weight to incorrect stumps. Hence, the final classifier boosts the stumps that have more correct classifications as compared to stumps that predict incorrectly and combines them to produce a superior classifier.

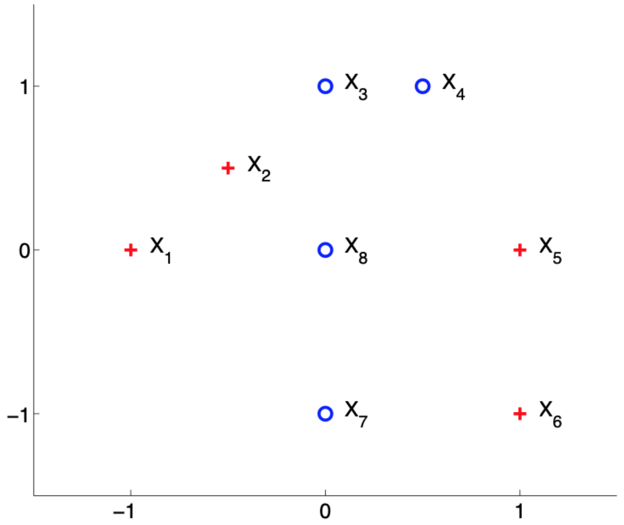


Figure 3: A small dataset, for binary classification with AdaBoost.

Table 1: Values of AdaBoost parameters at each timestep.

t	ϵ_t	α_t	Z_t	$D_t(1)$	$D_t(2)$	$D_t(3)$	$D_t(4)$	$D_t(5)$	$D_t(6)$	$D_t(7)$	$D_t(8)$
1											
2											
3											

2. **Random forest for email spam classifier** (25 points)

   Your task for this question is to use random forest to build a spam classifier using the UCR email spma dataset `https://archive.ics.uci.edu/ml/datasets/Spambase` came from the postmaster and individuals who had filed spam. The collection of non-spam e-mails came from filed work and personal e-mails, and hence the word '`george`' and the area code '`650`' are indicators of non-spam. These are useful when constructing a personalized spam filter.

   You will see there are total of 4601 instances, and 57 features.

   (a) (5 points) How many instances of spam versus regular emails are there in the data? How many data points there are? How many features there are?

   Note: there may be some missing values, you can just fill in zero.

   - Answer:
     There are a total of 4601 data points with 57 features.
     There are 1813 spam and 2788 regular emails.

   (b) (10 points) Build a classification tree model (also known as the CART model). In our answer, you should report the tree models fitted similar to what is shown in the "Random forest" lecture, Page 16, the tree plot.

   - Answer:
     Figure 4 shows a classification tree built on the dataset.

   (c) (10 points) Also build a random forrest model.

   Now partition the data to use the first 80% for training and the remaining 20% for testing. Your task is to compare and report the AUC for your classification tree and random forest models on testing data, respectively. To report your results, please try different tree sizes. Plot the curve of AUC versus Tree Size, similar to Page 15 of the Lecture Slides on "Random Forest".

   Background information: In classification problem, we use AUC (Area Under The Curve) as a performance measure. It is one of the most important evaluation metrics for checking any classification model?s performance. ROC (Receiver Operating Characteristics) curve measures classification accuracy at various thresholds settings. AUC measures the total area under the ROC curve. Higher the AUC, better the model is at distinguishing the two classes. If you want to read a bit more about AUC curve, check out this link `https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5`

   - Answer:
     Figure 5 displays a plot of the tree size vs AUC. For this particular dataset, the Random Forest model outperforms the CART model. This can be attributed to the fact the Random Forest model builds multiple trees and averages across them, whereas the CART model only constructs a single tree.

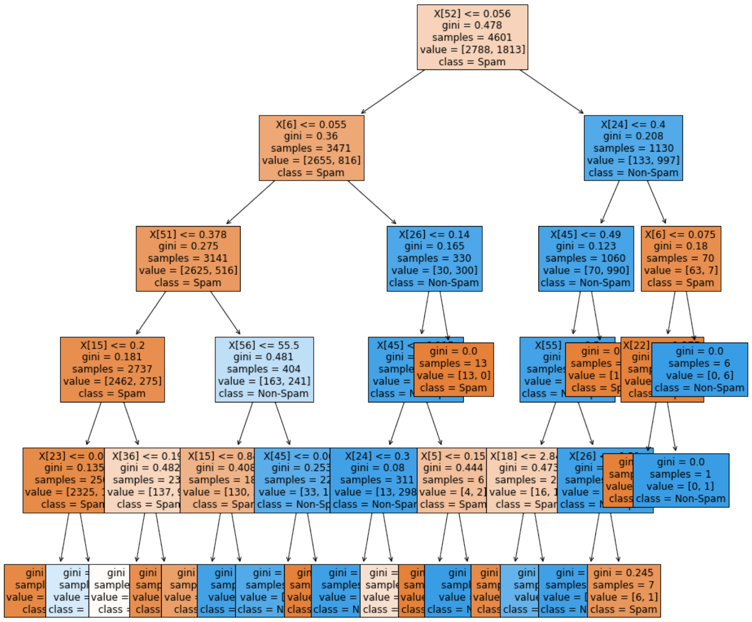


Figure 4: CART built on the dataset

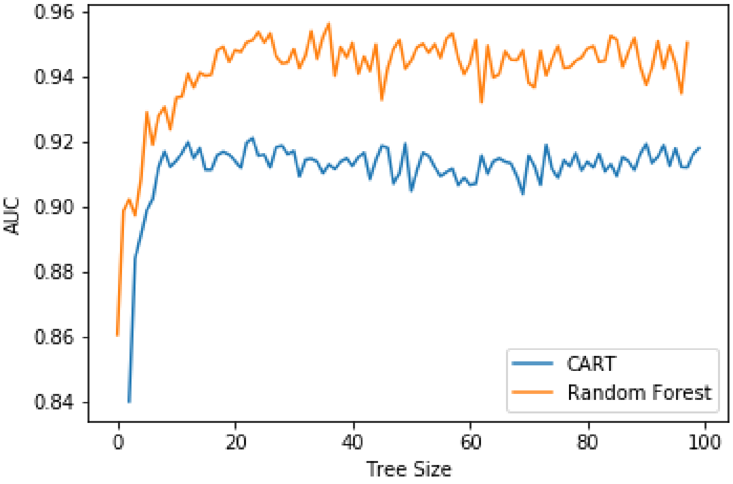
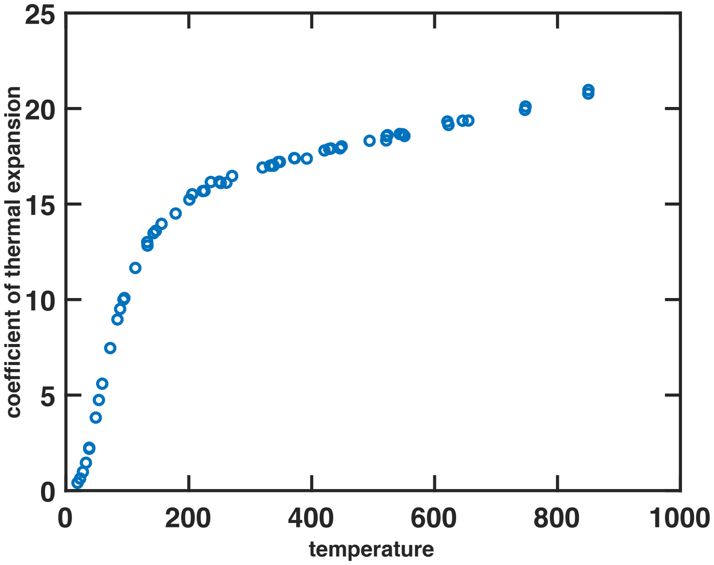


Figure 5: AUC vs Tree size for Random Forest and CART

3. **Nonlinear regression and cross-validation** (25 points)

The coefficient of thermal expansion y changes with temperature x. An experiment to relate y to x was done. Temperature was measured in degrees Kelvin. (The Kelvin temperature is the Celsius temperature plus 273.15). The raw data file is copper-new.txt.



(a) (5 points) Perform linear regression on the data. Report the fitted model and the fitting error.

- Answer:
  After fitting a linear regression model to the dataset, the results were:
  The MSE of the model is 12.48
  The intercept is 7.7156
  The coefficient is 0.0205
  Therefore, the linear equation is $y = 0.0205x + 7.7156$

(b) (5 points) Perform nonlinear regression with polynomial regression functions up to degree $n = 4$ and use ridge regression (see Lecture Slides for "Bias-Variance Tradeoff"). Write down your formulation and strategy for doing this, the form of the ridge regression.

- Answer:
  Using sklearn's PolynomialFeatures package, I first constructed the polynomial matrix using the input dataset with $n = 4$. Then, I built a ridge regression classifier and fit it to the polynomial matrix. Lastly, I used the ridge regression classifier to predict the coefficient of thermal expansion values. Additionally, we can also perform cross validation and use the average

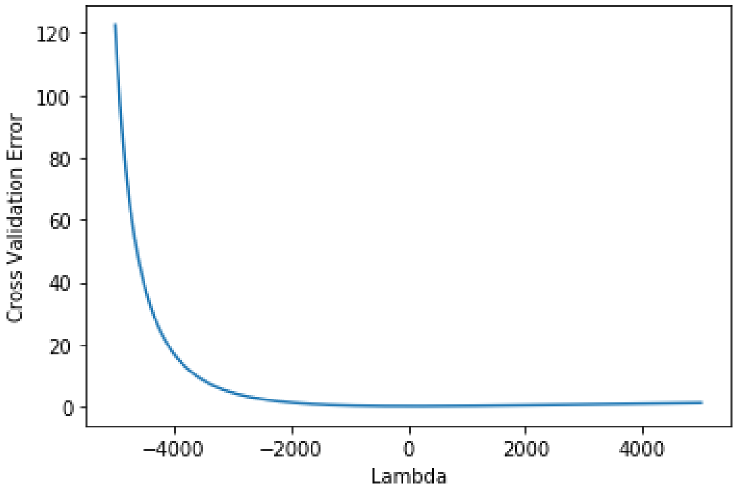


Figure 6: Cross Validation Error vs Lambda

MSE to determine the average error across the polynomial models. Please see attached code for the layout of the approach.

(c) (10 points) Use 5 fold cross validation to select the optimal regularization parameter λ. Plot the cross validation curve and report the optimal λ.

- Answer:
  Using 5 fold cross validation, I obtained the plot shown in Figure 6. From this plot, the optimal value λ occurs at $\lambda = 53$.

(d) (5 points) Predict the coefficient at 400 degree Kelvin using both models. Comment on how would you compare the accuracy of predictions.

- Answer:
  Using the linear regression model, the predicted value of the coefficient at 400 degree Kelvin is 15.90. Using the ridge regression model, the coefficient is 17.48.
  Using the mean squared error, the MSE for the ridge model is 0.22, which is significantly lower than the MSE of the linear regression model.

  Also, by observing the original dataset, the closest data points to 400 degree kelvin are 422 and 393. The coefficient values for these 2 data points are 17.765 and 17.339, respectively. Hence, given that the ridge regression model predict 17.48, which lies in between these 2 values, shows that it is more likely to the closer value. Hence, we can conclude that the ridge regression model has a higher accuracy in this case.

4. **Regression, bias-variance tradeoff** (25 points)

Consider a dataset with n data points (x_i, y_i), $x_i \in \mathbb{R}^p$, drawn from the following linear model:

$$y = x^T\beta^* + \epsilon,$$

where ϵ is a Gaussian noise and the star sign is used to differentiate the true parameter from the estimators that will be introduced later. Consider the regularized linear regression as follows:

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \left\{ \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda\|\beta\|_2^2 \right\},$$

where $\lambda \geq 0$ is the regularized parameter. Let $X \in \mathbb{R}^{n\times p}$ denote the matrix obtained by stacking x_i^T in each row.

(a) (10 points) Find the closed form solution for $\hat{\beta}(\lambda)$ and its distribution.

- Answer:
  To find the closed form solution of $\hat{\beta}(\lambda)$, we first differentiate w.r.t β and set it equal to 0

$$\frac{d\hat{\beta}(\lambda)}{d\beta} = \frac{2}{n}(y_i - x_i^T\beta)(-x_i^T) + 2\lambda\beta = 0$$

$$\frac{d\hat{\beta}(\lambda)}{d\beta} = \frac{2}{n}(-x_i^Ty_i + x_i^Tx_i\beta) + 2\lambda\beta = 0$$

$$0 = 2(-x_i^Ty_i + x_i^Tx_i\beta) + 2n\lambda\beta$$

$$x_i^Ty_i = x_i^Tx_i\beta + n\lambda\beta$$

$$\beta = \frac{x_i^Ty_i}{x_i^Tx_i + n\lambda}$$

$$\implies \hat{\beta}(\lambda) = (x_i^Ty_i)(x_i^Tx_i + n\lambda)^{-1}$$

Using the affine transformation property

$$y \sim N(x^T\beta^*, \sigma^2 I)$$

$$\implies \hat{\beta}(\lambda) \sim N((x^Tx + n\lambda)^{-1}x^Tx\beta^*, (x^Tx + n\lambda)^{-1}x^Tx(xx^T + \lambda I)^{-1})$$

(b) (5 points) Calculate the bias $\mathbb{E}[x^T\hat{\beta}(\lambda)] - x^T\beta^*$ as a function of λ and some fixed test point x.

- Answer:
  To calculate the bias, we can make use of our result from the previous problem:

$$\mathbb{E}[x^T\hat{\beta}(\lambda)] - x^T\beta^* = x^T(\mathbb{E}[\hat{\beta}(\lambda)] - \beta^*)$$

$$x^T(\mathbb{E}[\hat{\beta}(\lambda)] - \beta^*) = x^T((x^Tx + n\lambda)^{-1}x^Tx\beta^*) - \beta^*)$$

$$x^T((x^Tx + n\lambda)^{-1}x^Tx\beta^*) - \beta^*) = x^T((x^Tx + n\lambda I)^{-1}x^Tx) - I)\beta^*$$

$$\therefore Bias = x^T((x^Tx + n\lambda I)^{-1}x^Tx) - I)\beta^*$$

(c) (5 points) Calculate the variance term $\mathbb{E}\left[\left(x^T\hat{\beta}(\lambda) - \mathbb{E}[x^T\hat{\beta}(\lambda)]\right)^2\right]$.

- Answer:

$$\mathbb{E}\left[\left(x^T\hat{\beta}(\lambda) - \mathbb{E}[x^T\hat{\beta}(\lambda)]\right)^2\right] = \mathbb{E}\left[\left(x^T(\hat{\beta}(\lambda) - \mathbb{E}[\hat{\beta}(\lambda)])\right)^2\right]$$

$$= x^T\mathbb{E}\left[\left((\hat{\beta}(\lambda) - \mathbb{E}[\hat{\beta}(\lambda)])\right)^2\right]x$$

  Using the result for variance from part a:

$$Variance = x^T(x^Tx + n\lambda)^{-1}x^Tx(xx^T + \lambda I)^{-1}x$$

(d) (5 points) Use the results from parts (b) and (c) and the bias-variance decomposition to analyze the impact of λ in the squared error. Specifically, which term dominates when λ is small, and large, respectively?

- Answer:
  Plugging in values for λ in bias and variance, we see that if we set λ to a very small value, the bias is also very small whereas the variance is very high. On the contrary, if we set λ to a large value, the bias increases but variance decreases.
  Reference: ESL, pg 64

(Hint.) Properties of an affine transformation of a Gaussian random variable will be useful throughout this problem.