Sta 561 / Comp 571: Homework 2

1 Classifiers for Basketball Courts

In this problem, you will use linear classifiers and decision trees to classify regions of a basketball court. If you do not know anything about basketball, the left-hand-side of Figure 1 tells you everything you need to know [1]. When a player shoots the ball and it goes in the basket, the shot is worth three points if they are behind the *3-point line* and two points otherwise. There is also a special region called the *paint*. We will use linear classifiers (that pass through the origin, for simplicity) and decision trees to classify these two regions, the area outside the 3-point line and the paint. To make the problem simpler, we will consider only the upper-right quadrant of the court and we will approximate the 3-point line with the square root function (see the right-hand-side of Figure 1).

Recall that at each node, a decision tree splits based on an impurity measure I. One such impurity measure is the *Gini index*, which is defined as follows for 2 classes:

$$I(p, 1-p) = 1 - p^2 - (1-p)^2 = 2p(1-p),$$

where p is the fraction of positives in the node. We choose to split the node with the best reduction in Gini index, averaged across the leaves (children) of the possible split. Denote N as the number of observations in the node we are considering to split, p as the fraction of positives in the node we are considering to split, p_c as the fraction of positives in the cth branch of the potential split, $1 - p_c$ as the fraction of negatives in the cth branch of the potential split, and N_c as the number of observations falling into the cth branch of the potential split. Then:

$$\Delta I = I(p, 1-p) - \sum_{\text{children } c} \frac{N_c}{N} I(p_c, 1-p_c).$$

In this question, use the reduction in the Gini index, ΔI, to grow all decision trees. Throughout this problem, all decision trees are binary decision trees (*i.e.*, each split is binary).

We will start with the 3-point line. For parts (a) and (b), suppose you do not know anything about location of the 3-point line, so you record the coordinates of a shot (*i.e.*, the position of the player when they shot the ball), $(X_1, X_2) \in [0,1] \times [0,1]$, and the label assigned to a shot, $Y \in \{-1, 1\}$. A label of $Y = 1$ means a shot is assigned three points while a label of $Y = -1$ means a shot is assigned two points. Table 1 shows the data you observe while watching a game. What could the 3-point line look like? How would you classify the label of a new shot given its coordinates? In part (a) you will find a linear classifier and in part (b) you will find a decision tree.

(a) Run the Perceptron algorithm to compute a linear classifier that passes through the origin. How many iterations does it take to converge? What is the error of the classifier? Are

[1] Court diagram taken from `http://unmasadalha.blogspot.com/2016/01/basketball-court-diagram.html`

Figure 1: *Left:* Diagram of a full basketball court. A player scores by shooting the ball in the basket. If they are behind the 3-point line when they shoot, the shot is assigned three points, while if they are inside the 3-point line the shot is assigned two points. The paint is shaded in blue. *Right:* Approximation to the upper-right quadrant of a basketball court. This is the layout we will use in the problem. The 3-point line is the graph of $X_2 = \sqrt{X_1}$ and the scale is $[0, 1] \times [0, 1]$.

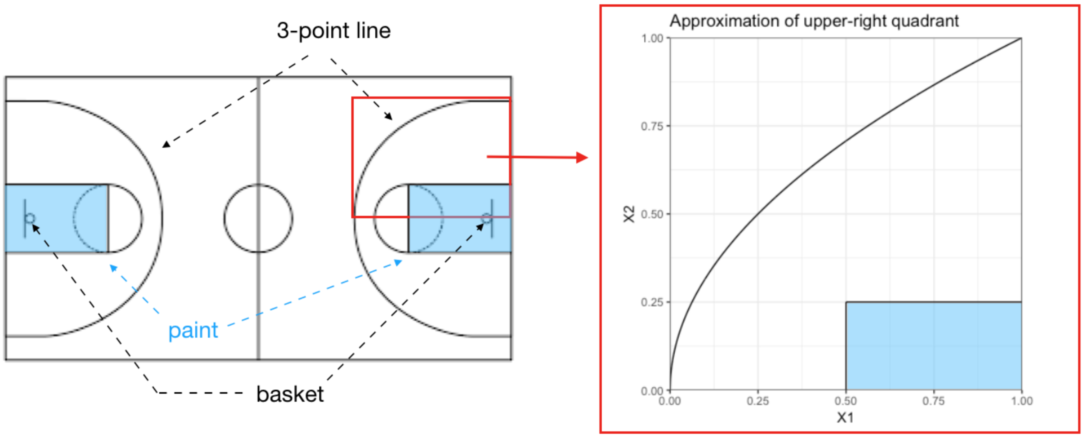


Table 1: Observed shots. (X_1, X_2) are the coordinates of a shot. A shot labeled $Y = 1$ is assigned three points and a shot labeled $Y = -1$ is assigned two points.

X_1	X_2	Y
.75	.10	-1
.85	.80	-1
.85	.95	1
.15	.10	-1
.05	.25	1
.05	.50	1
.85	.25	-1

there any other solutions that give the same error? Plot the observed data, the decision boundary found by the Perceptron algorithm, and describe in the plot any other linear decision boundaries passing through the origin that have the same error[2]. You may assume the Perceptron is initialized at $\boldsymbol{w} = \mathbf{0}$ and that it considers the points in the order given in Table 1. You may write code (or recycle code from Homework 1) or do the calculations by hand as long as you show all of your work.

(b) Grow a fully-grown decision tree using the reduction in the Gini index as the splitting criterion. What is its error? By adjusting the threshold of each split, are there any other solutions that give the same error? Plot the observed data, the decision boundaries of the decision tree, and describe in the plot any other decision trees you found by adjusting the threshold that have the same error.

So far we have estimated the optimal decision boundaries using observed data. For parts (c)-(f)

[2] This question asks you to create a number of plots. These plots do not need to be perfect as long as they are not ambiguous. You can draw a sketch by hand and include a picture, or use PowerPoint, for example. You may show all of the plots on the same page if that is easier.

we will use our knowledge of a basketball court to compute the optimal linear and decision tree classifiers. By an optimal classifier we mean one that minimizes the true risk. Recall from class that for a loss function l, a true joint distribution D, and a classifier f, the true is defined as:

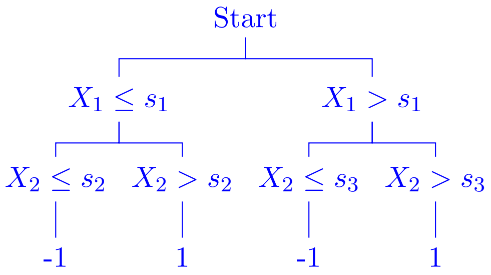
$$R^{\text{true}}(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim D} l(f(\boldsymbol{x}), y) \tag{1}$$

Use the misclassification loss $l(f(\boldsymbol{x}), y) = \mathbf{1}_{[\text{sign}(f(\boldsymbol{x}))\neq y]}$ and assume the position of a shot, $\boldsymbol{x} = (X_1, X_2)$, is uniformly distributed on $[0,1] \times [0,1]$. As a reminder, Y is the label assigned to a shot. Y is 1 (with probability 1) if $(X1, X2)$ is behind the 3-point line and Y is -1 (with probability 1) if $(X1, X2)$ is inside the 3-point line (inside the 3-point line means closer to the basket).

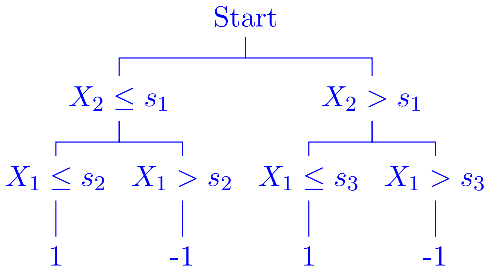
(c) What is the optimal linear classifier that passes through the origin and what is its error? Is this solution among the solutions that acheived the minimum empirical loss in part (a)? Draw the decision boundary on a plot of the court.

(d) Consider the two depth 2 decision trees below. For each tree, find the values of s_1, s_2, and s_3 that minimize the true risk, R^{true}. For the tree with the lower true risk, is this solution among the solutions that achieved the minimum empirical loss in part (b)? Draw the decision boundary on a plot of the court.

Tree 1: Split on X_1 first.



Tree 2: Split on X_2 first.



(e) Transforming the variables X_1 and/or X_2 by applying a function might improve the misclassification error. Find a transformation that minimizes the true risk of the optimal linear classifier (that passes through the origin) that uses the transformed variables. What is this classifier and what is its error (*i.e.*, its true risk)?

(f) Using the same transformation, can a decision tree achieve the same error?

(g) [This part was never included.]

The last two questions apply to a different classification problem, this time for classifying the paint. Assume outside the paint is labeled $Y = +1$ and inside the paint is labeled $Y = -1$ (if you already solved the problem the other way that is fine, just state your assumption in your solution). As in parts (c)-(f), we want to find the classifiers that minimize the (true) misclassification error (*i.e.*, the true risk).

(h) What is an optimal linear classifier that passes through the origin and what is its error (*i.e.*, its true risk)?

(i) What is an optimal depth 2 decision tree and what is its error (*i.e.*, its true risk)?

2 Variable importance for trees and random forests

In this question, you will investigate several measures of variable importance for trees and random forests. Software packages often lack detail and/or use slightly different definitions of variable importance for trees and random forests. There is not necessarily a "best" measure. The purpose of this question is for you to calculate a few measures yourself in order to learn about some the issues involved in variable importance for trees and random forests.

Recall that for each node t in a decision tree $\mathcal{T}$, we grow the decision tree by finding the split — defined by a variable X_{j_t} and a vector of cutoffs s_t (*e.g.*, $X_{j_t} \leq s_t$ in the binary tree case) — that maximizes the reduction in the impurity measure, $\Delta I(s_t, j_t, t)$. For decision trees, one measure of variable importance for a variable X_j is the total reduction in the impurity measure attributable to X_j:

$$\mathrm{Imp}^{\mathcal{T}}(X_j) := \sum_{t \in \mathrm{nodes}(\mathcal{T})} \mathbb{1}_{[j=j_t]} \Delta I(s_t, j_t, t). \tag{2}$$

Now, suppose there is a different variable $X_{\tilde{j}_t}$, where $\tilde{j}_t \neq j_t$, and cutoff $\tilde{s}_t$ that results in nearly as large of a reduction in the impurity measure. Should this count towards the variable importance of $X_{\tilde{j}_t}$ even though it is not used for the split at node t of tree $\mathcal{T}$? Well, $X_{\tilde{j}_t}$ is not important to $\mathcal{T}$ at node t in the sense that $\mathcal{T}$ does not require $X_{\tilde{j}_t}$ at node t in order to achieve its predictive performance on the training dataset. However, suppose a tree $\tilde{\mathcal{T}}$ were trained on a second training dataset drawn from the same distribution. The large reduction in the impurity measure due to $X_{\tilde{j}_t}$ on the first dataset suggests a reasonably high likelihood that $X_{\tilde{j}_t}$ would have a higher reduction in the impurity measure on the second dataset than X_{j_t}, in which case $X_{\tilde{j}_t}$ would have a higher variable importance (as calculated by Equation (2)) than X_{j_t} with respect to tree $\tilde{\mathcal{T}}$ at node t. In this sense, we say that X_{j_t} is *masking* the *potential variable importance* of $X_{\tilde{j}_t}$. A measure of variable importance that more effectively captures the variable importance *and* potential variable importance is one that includes the impurity measurement that would occur if $X_{\tilde{j}_t}$ were used as the split, even though it is not. We define this measure as follows:

$$\mathrm{Imp}_s^{\mathcal{T}}(X_j) := \sum_{t \in \mathrm{nodes}(\mathcal{T})} \mathbb{1}_{[j=j_t]} \Delta I(s_t, j_t, t) + \mathbb{1}_{[j=\tilde{j}_t]} \Delta I(\tilde{s}_t, \tilde{j}_t, t), \tag{3}$$

where $X_{\tilde{j}_t}$ and cutoff $\tilde{s}_t$ constitute the best *surrogate split*. In other words, for each node, find the best split and the best surrogate split (which is chosen among all of the variables that are not used in the best split). If the variable used in either split is X_j (note that by construction both splits cannot use X_j), then add the reduction in the impurity measure to the importance of X_j. Note that Equations (2) and (3) are sums over *all* nodes in the tree.

We have not yet discussed the way to define the "best" surrogate split. For a binary tree, we define the best surrogate split by the variable $X_{\tilde{j}_t}$, where $X_{\tilde{j}_t}$ is not the actual best split variable X_j, and cutoff $\tilde{s}_t$ that maximize the following predictive similarity measure:

$$\lambda(j_t, s_t, \tilde{j}_t, \tilde{s}_t) = \frac{\min(p_L, p_R) - (1 - P_{L_{j_t} L_{\tilde{j}_t}} - P_{R_{j_t} R_{\tilde{j}_t}})}{\min(p_L, p_R)} \tag{4}$$

where p_L is the proportion of observations such that $X_{j_t} < s_t$, p_R is the proportion of observations such that $X_{j_t} \geq s_t$, $P_{L_{j_t} L_{\tilde{j}_t}}$ is the proportion of observations such that $X_{j_t} < s_t$ and $X_{\tilde{j}_t} < \tilde{s}_t$, and $P_{R_{j_t} R_{\tilde{j}_t}}$ is the proportion of observations such that $X_{j_t} \geq s_t$ and $X_{\tilde{j}_t} \geq \tilde{s}_t$. Choosing the best surrogate split by maximizing $\lambda(j_t, s_t, \tilde{j}_t, \tilde{s}_t)$ is the method used by `MATLAB`, for example.

For random forests, we can extend the decision tree measures of variable importance by simply averaging over all of the trees. For a random forest $\mathcal{F}$ composed of trees $\{\mathcal{T}_i\}_{i=1}^{M}$, the analog to $\text{Imp}^{\mathcal{T}}$ is given by:

$$\text{Imp}^{\mathcal{F}}(X_j) := \frac{1}{M} \sum_{i=1}^{M} \text{Imp}^{\mathcal{T}_i}(X_j) \tag{5}$$

There is also another method, as discussed in class, that uses permuted "out-of-bag" samples:

$$\text{Imp}^{\mathcal{F}}_{\text{OOB}}(x_j) := \frac{1}{M} \sum_{i=1}^{M} \text{error}_{\text{OOB}}(\mathcal{T}_i, x_j^{\text{perm}}) - \text{error}_{\text{OOB}}(\mathcal{T}_i, x_j) \tag{6}$$

where $\text{error}_{\text{OOB}}(\mathcal{T}_i, x_j)$ is the out-of-bag error of tree $\mathcal{T}_i$ predicting on bootstrap sample i with X_j unadjusted and $\text{error}_{\text{OOB}}(\mathcal{T}_i, x_j^{\text{perm}})$ is the out-of-bag error of tree $\mathcal{T}_i$ predicting on bootstrap sample i with X_j randomly permuted. See the class notes for more discussion of this variable importance measure. We will use the least-squares error. When taking the average in Equations (5) and (6), if a variable is not used in the decision tree, ignore this term in taking the average. Only average over trees in which variable X_j is used in a split.

For this question, use the accompanying training and test data, `train.csv` and `test.csv`. You will find $n = 500$ and $n_{\text{test}} = 100$ observations of a binary outcome Y and five binary covariates, $X_1, \ldots, X_5$. You may use any combination of your own code or an existing package to answer any part of this problem. If you choose to use a package, make sure you read the documentation carefully to understand exactly what it is doing. For simplicity you will use decision stumps (*i.e.*, decision trees with one split)[3]. Use the Gini index as the impurity measure. If a variable is not used in a decision tree or decision forest we cannot calculate its variable importance, so report it as `NA`.

(a) Grow a decision stump on the training dataset and answer the following questions.

  (i) Describe clearly (or draw) the decision stump based on the best split and the decision stump based on the best surrogate split.

  (ii) Report the variable importance measurements from Equations (2) and (3) for the tree based on the best split. Does this suggest any variable(s) are more important than the others?

[3] Random forests are usually composed of fully grown decision trees. In this problem we use decision stumps only. Stumps are just special cases of decision trees (they have only one split) so all of the variable importance measures we defined for decision trees apply to decision stumps.

(iii) Report the mean least-squares error of predictions on the test dataset of both decision stumps from part (a)(i).[4]

(b) Grow a random forest of decision stumps on the training dataset for $K = 1, \dots, 5$ randomly selected variables in each stump. Use $M = 1000$ stumps and $B = 0.8 \times n$ bootstrap samples. The following question parts should be done for each K (ideally with your numerical results summarized in a single table for each part). For each part, discuss any dependence of your answers on K and why this may have occurred.[5].

(i) How many times is each variable the best split? How many times is each variable the best surrogate split? Does this suggest any variable(s) are more important than the others?

(ii) Compute the variable importance measures in Equations (5) and (6). Does this suggest any variable(s) are more important than the others? Recall that when using Equation (2) to compute variable importance for decision stumps, the phenomenon of "masking" can hide the potential variable importance of some variables. When using Equations (5) and (6) to compute variable importance for random forests, can masking similarly hide the potential variable importance of some variables, or is the impact of masking lessened? (On the topic of masking, you do not need to provide a numerical answer, just a brief discussion of the role of masking).

(iii) Compute the mean least-squares loss on the test data using two methods. In the first method, use the majority vote of the stumps as the prediction and compute the loss. In the second method, find the predictions of each stump, compute least-squares loss on each, and average the results. Which method is correct for computing the prediction error of the random forest?

(c) Grow a random forest of decision stumps with $B = q \times n$ bootstrap samples, for each of $q \in \{0.4, 0.5, 0.6, 0.7, 0.8\}$. Use $K = 2$ randomly selected variables in each stump (this is the closest to the default choice of $\sqrt{p} \approx 2.23$) and $M = 1000$ stumps. The following question parts should be answered for each B (ideally with your numerical results summarized in a single table for each part). For each question, discuss any dependence on B and why this may have occurred.

(i) Compute the variable importance measurements in Equations (5) and (6). Does this suggest any variable(s) are more important than the others?

(ii) Compute the standard deviation of the variable importance measurements in Equations (5) and (6). That is, instead of computing the mean over the M stumps in Equations (5) and (6), compute the sample standard deviation.

[4] In other words, if $\{\hat{y}_i\}_{i=1}^{n_{\text{test}}}$ are the predictions of the model and $\{y_i\}_{i=1}^{n_{\text{test}}}$ are the actual labels, then the error you should evaluate is $\frac{1}{n}\sum_{i=1}^{n_{\text{test}}}(\hat{y}_i - y_i)^2$.

[5] Parts (b) and (c) ask you to interpret your answers based on K and B. There is not one correct answer, you will be graded on the overall quality of your response.