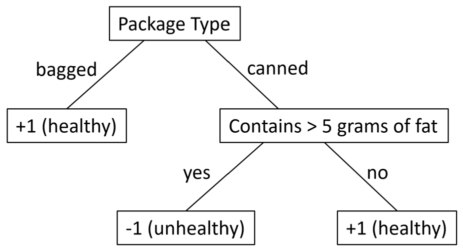
Machine Learning & Data Mining
Caltech CS/CNS/EE 155
Homework 3

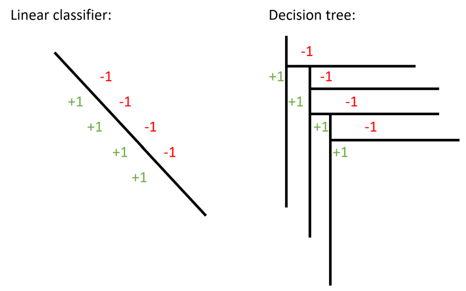
Changhao Xu
UID: 2103530
January 30th, 2020

1 Decision Trees

Question A: At root node, $p_{s'}$ = 0.75 and S' = 4, entropy = 2.249.
At 1st layer:
Package type: bagged = $\{+1,+1\}$, canned = $\{+1,-1\}$, entropy = 1.386
Unit price $>$ \$5: yes = $\{+1,-1\}$, no = $\{+1,+1\}$, entropy = 1.386
Contains $>$ 5 grams of fat: yes = $\{+1,-1\}$, no = $\{+1,+1\}$, entropy = 1.386
Here, since entropy is same, we choose package type as 1st layer.
At 2nd layer:
Unit price $>$ \$5: yes = -1, no = +1, entropy = 0
Contains $>$ 5 grams of fat: yes = -1, no = +1,entropy = 0
Here, since entropy is same, we choose Contains $>$ 5 grams of fat as 2nd layer.



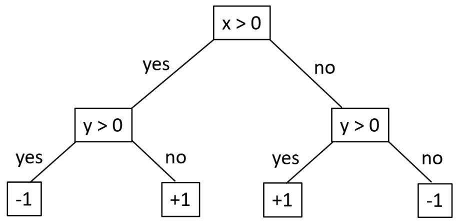
Question B: Decision tree is not always preferred for classification problems compared to linear classifier. For example:



Question C: i. At root node, $p_{s'}$ = 0.5, Impurity via Gini index = $4 \times (1 - 0.5^2 - 0.5^2) = 2$.

Suppose split on X/Y axis, then impurity becomes $2 \times (1 - 0.5^2 - 0.5^2) + 2 \times (1 - 0.5^2 - 0.5^2) = 2$. So further split won't reduce impurity. Therefore, decision tree is just one root node, and classification error = 0.5.

ii.



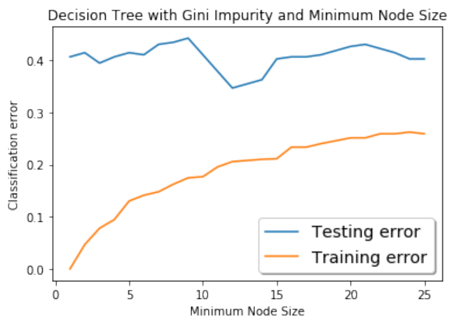
Impurity measure $L(S') = |S'|^2 \times \left[1 - p_{s'}^2 - (1 - p_{s'})^2\right]$. This will lead to an initial error of 8 at root node, then error becomes 4 at 1st layer, and becomes 0 at leaf node. The pros of this error measure is that it can successfully split a data set that have same percentage of correct and incorrect classifications before and after the split. The cons of this error measure is that it highly depends on the number of classified points and misclassified points, and may lead to overfitting due to encouragement of splitting.

iii. The largest number of internal nodes will be 99 to achieve zero classification training error. Suppose 100 points in 1D, which are labeled as 1, -1, 1, -1, ... Then it needs each data point to be in a unique split. Therefore, the decision tree will have (100-1) = 99 nodes.

Question D: As described in C(iii), N data points will need (N-1) internal nodes at most. Then worst-case complexity = O(N*D).

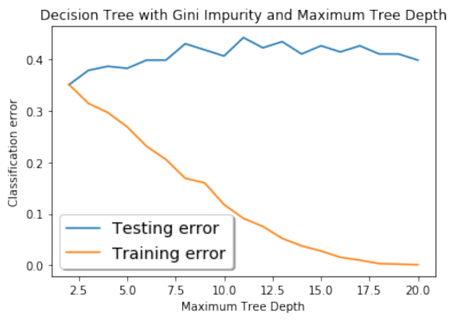
2 Overfitting Decision Trees

Question A:



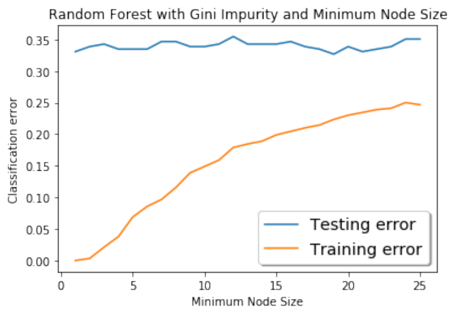
Changhao Xu
UID: 2103530
January 30th, 2020

Question B:

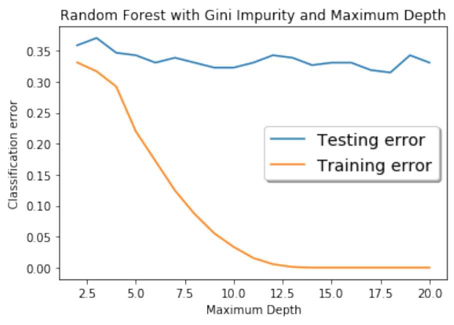


Question C: For the minimal leaf node size, 12 has minimized test error. For maximum depth parameters, 2 has minimized test error. Early stopping will improve the performance of a decision tree model. In Question A, when minimal leaf node size = 1 (no early stopping), test error is high. Similarly, in Question B, when maximum depth = 20 (minimized early stopping), test error is high.

Question D:



Question E:



Changhao Xu
UID: 2103530
January 30th, 2020

Question F: For the minimal leaf node size, 19 minimizes the random forest test error. For maximum depth parameters, 18 minimizes the random forest test error. Early stopping will not improve, or even impair the performance of a random forest model. In Question D, when minimal leaf node size = 1 (no early stopping), test error is stable and early stopping doesn't have much effect on lowering test error. Similarly, in Question E, when maximum depth = 20 (minimized early stopping), test error is low and early stopping increases test error.

Question G: Test error for random forest is lower and the curve is smoother than that of decision tree model. Decision tree is very likely to overfitting, so early stopping is desired; while random forest samples both data and features, and thus reduces variance, so early stopping doesn't have much effect.

3 The AdaBoost Algorithm

Question A:
If y_i is correctly classified, i.e. $-y_i f(x_i) < 0$. Then $exp(-y_i f(x_i)) > 0$, while $\mathbb{1}(H(x_i) \neq y_i) = 0$.
If y_i is incorrectly classified, i.e. $-y_i f(x_i) > 0$. Then $exp(-y_i f(x_i)) > 1$, while $\mathbb{1}(H(x_i) \neq y_i) = 1$.
Therefore, $exp(-y_i f(x_i)) \geq \mathbb{1}(H(x_i) \neq y_i)$ for $\forall i \in N$. And thus

$$E = \frac{1}{N}\sum_{i=1}^{N} \exp(-y_i f(x_i)) \geq \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(H(x_i) \neq y_i)$$

Question B: In lecture 6,

$$D_{t+1}(i) = \frac{D_t(i) exp\left\{-\alpha_t y_i h_t(x_i)\right\}}{Z_t}$$

, so

$$D_{T+1}(i) = D_1(i) \prod_{t=1}^{T} \frac{exp\left\{-\alpha_t y_i h_t(x_i)\right\}}{Z_t}$$

, where $D_1(i) = \frac{1}{N}$. Therefore,

$$D_{T+1}(i) = \frac{1}{N} \prod_{t=1}^{T} \frac{exp\left\{-\alpha_t y_i h_t(x_i)\right\}}{Z_t}$$

, where Z_t is the normalization factor

$$Z_t = \sum_{i=1}^{N} D_t(i) exp\left\{-\alpha_t y_i h_t(x_i)\right\}$$

Question C:

$$E = \frac{1}{N}\sum_{i=1}^{N} \exp(-y_i f(x_i))$$

, where

$$f(x_i) = \sum_{t=1}^{T} \alpha_t h_t(x_i)$$

, so

$$E = \frac{1}{N}\sum_{i=1}^{N}\exp(-y_i\sum_{t=1}^{T}\alpha_t h_t(x_i))$$

Question D: From Question B,

$$D_{T+1}(i) = \frac{1}{N}\prod_{t=1}^{T}\frac{exp\left\{-\alpha_t y_i h_t(x_i)\right\}}{Z_t} = \frac{1}{N}\frac{\prod_{t=1}^{T} exp\left\{-\alpha_t y_i h_t(x_i)\right\}}{\prod_{t=1}^{T} Z_t} = \frac{1}{N}\frac{exp\left\{\sum_{t=1}^{T} -\alpha_t y_i h_t(x_i)\right\}}{\prod_{t=1}^{T} Z_t}$$

Therefore,

$$E = \frac{1}{N}\sum_{i=1}^{N}\exp(-y_i\sum_{t=1}^{T}\alpha_t h_t(x_i)) = \sum_{i=1}^{N}\frac{1}{N}\exp(\sum_{t=1}^{T}-\alpha_t y_i h_t(x_i))$$
$$= \sum_{i=1}^{N} D_{T+1}(i)\prod_{t=1}^{T} Z_t$$

Since $\sum_{i=1}^{N} D_t(i) = 1$,

$$E = \prod_{t=1}^{T} Z_t$$

.

Question E:

$$Z_t = \sum_{i=1}^{N} D_t(i) exp\left\{-\alpha_t y_i h_t(x_i)\right\}$$

If y_i is correctly classified, i.e. $-y_i h_t(x_i) = -1$. $Z_t = \sum_{i=1}^{N} D_t(i)exp(-\alpha_t) = exp(-\alpha_t)$.
In this case, $\epsilon_t = \sum_{i=1}^{N} D_t(i)\mathbb{1}(H(x_i) \neq y_i) = 0$, so $Z_t = (1-\epsilon_t)exp(-\alpha_t) + \epsilon_t exp(\alpha_t)$
If y_i is incorrectly classified, i.e. $-y_i h_t(x_i) = 1$. $Z_t = \sum_{i=1}^{N} D_t(i)exp(\alpha_t) = exp(\alpha_t)$.
In this case, $\epsilon_t = \sum_{i=1}^{N} D_t(i)\mathbb{1}(H(x_i) \neq y_i) = 1$, so $Z_t = (1-\epsilon_t)exp(-\alpha_t) + \epsilon_t exp(\alpha_t)$.

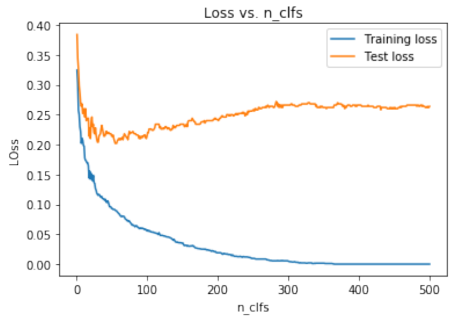
In both cases, $Z_t = (1-\epsilon_t)exp(-\alpha_t) + \epsilon_t exp(\alpha_t)$

Question F:

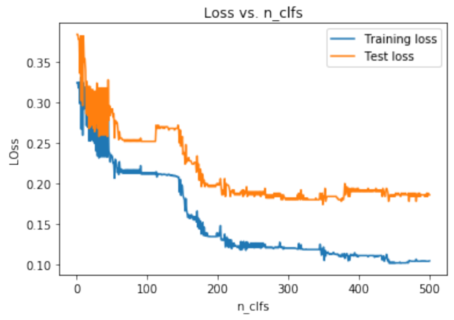
$$\frac{dZ_t}{d\alpha_t} = -\alpha_t(1-\epsilon_t)exp(-\alpha_t) + \alpha_t\epsilon_t exp(\alpha_t) = 0$$
$$\alpha_t\epsilon_t exp(\alpha_t) = \alpha_t(1-\epsilon_t)exp(-\alpha_t)$$
$$\epsilon_t exp(\alpha_t) = (1-\epsilon_t)exp(-\alpha_t)$$
$$\epsilon_t exp(2\alpha_t) = (1-\epsilon_t)$$
$$\alpha_t = \frac{1}{2}ln(\frac{1-\epsilon_t}{\epsilon_t})$$

Question G:

See Jupyter notebook. Loss curve for Gradient Boosting:



Loss curve for Adaboost:



Question H : Gradient boosting has a smoother training and testing loss. Training loss of Gradient boosting approaches 0 steadily, while test loss only decreased initially and then slowly increase, which is due to overfitting. Adaboost has a rough training and testing loss, which is due to the Decision tree classifier used and 0/1 loss. Training loss of Adaboost approaches 0, and test loss also decreases until a steady state.

Question I: Final loss value for Gradient boosting is 0.264, and 0.186 for Adaboost. Adaboost performed better on the classification dataset.

Question J : Dataset weights are the largest at the decision boundary (where red and blue points overlap), and smallest when data point is furthest away from the decision boundary.