CS 5350/6350: Machine Learning Spring 2020

Homework 2

Britton Gaul
u0915408

Handed out: 11 Feb, 2019
Due date: 11:59pm, 25 Feb, 2019

1 Paper Problems [40 points + 8 bonus]

1.

(a)

i.

According to Occam's Razor H_2 should be chosen for the result hypothesis. This is because both of the result hypothesises are consistent with the training data, but H_2 is smaller than H.

ii.

This prinicple is reflected in the PAC guarantee, because the samller hypothesis that was chosen is more likely to be less then the total number of m examples. This is required because it is stated in the given equation.

(b)

$3^{10} \cdot .90 = 53,145$ training examples needed.

2.

3.

(a)

$x_1 - x_2 - x_3 \geq 2$

(b)

$-x_1 - x_2 - x_3 \geq -1$

(c)

$x_1 + x_2 + x_3 + x_4 \geq 2$

(d)

$x_1 + x_2 - x_1 + x_2 \geq 1$

4.

5.

(a)
$j(w,b) = \frac{1}{2}\sum_{n=1}^{m}(y_i - w^T x_i)^2 \cdot b$

(b)

(c)

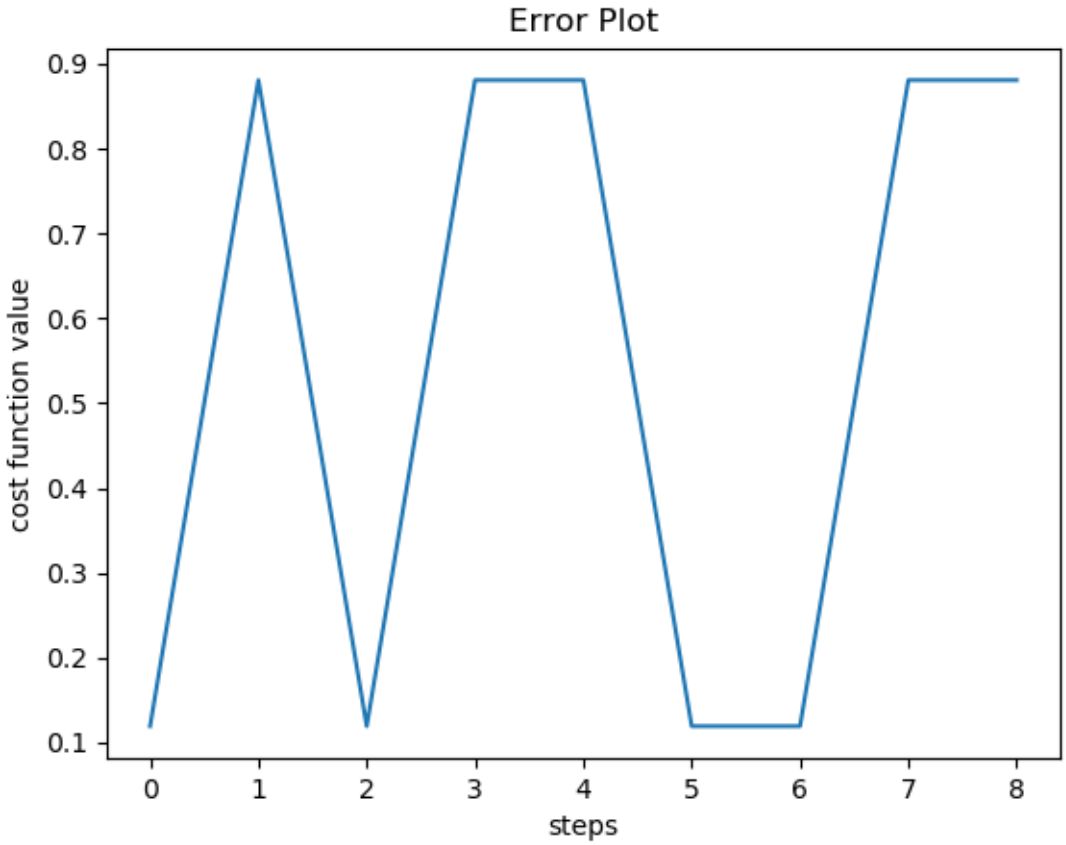
(d)

2 Practice [60 points + 10 bonus]

1. [2 Points]
Repository: https://github.com/BritGaul/CS5350

2.

(a)



When using the Adaboost decision trees the errors are smaller compared to a fully expanded decsion tree, which makes them more accurate.

(b)

Overall the bagged trees seem to be better than a single tree. However, when

compared to the Addaboost trees it seems that the adaboosted trees are more accurate.

(c)

When comparing the results from the single tree learner verses the bagged trees, the bagged trees are more accurate. This is because the bagged trees can learn from the accuracies of previous trees, while the single tree has no other trees to learn from.

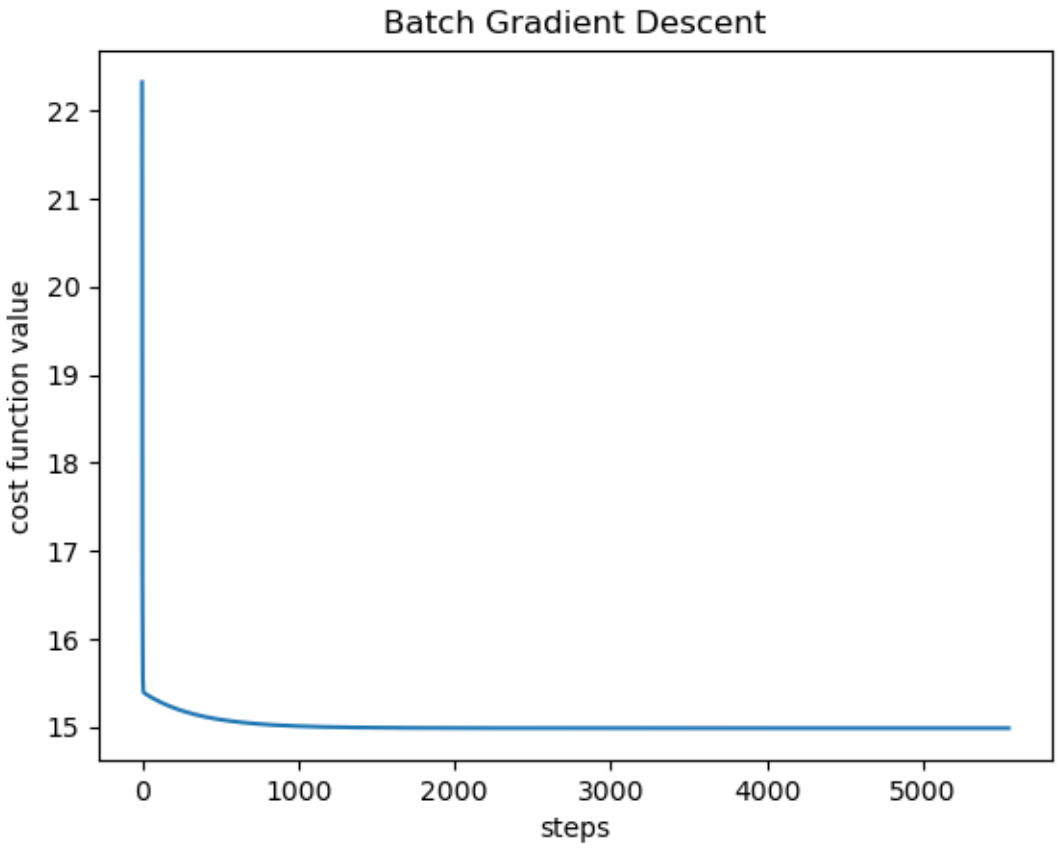
(d)

The random forrest algorithm runs more efficiently then the bagged trees alogorithm, but is less accurate then the bagged trees implementation.

3.

(a)

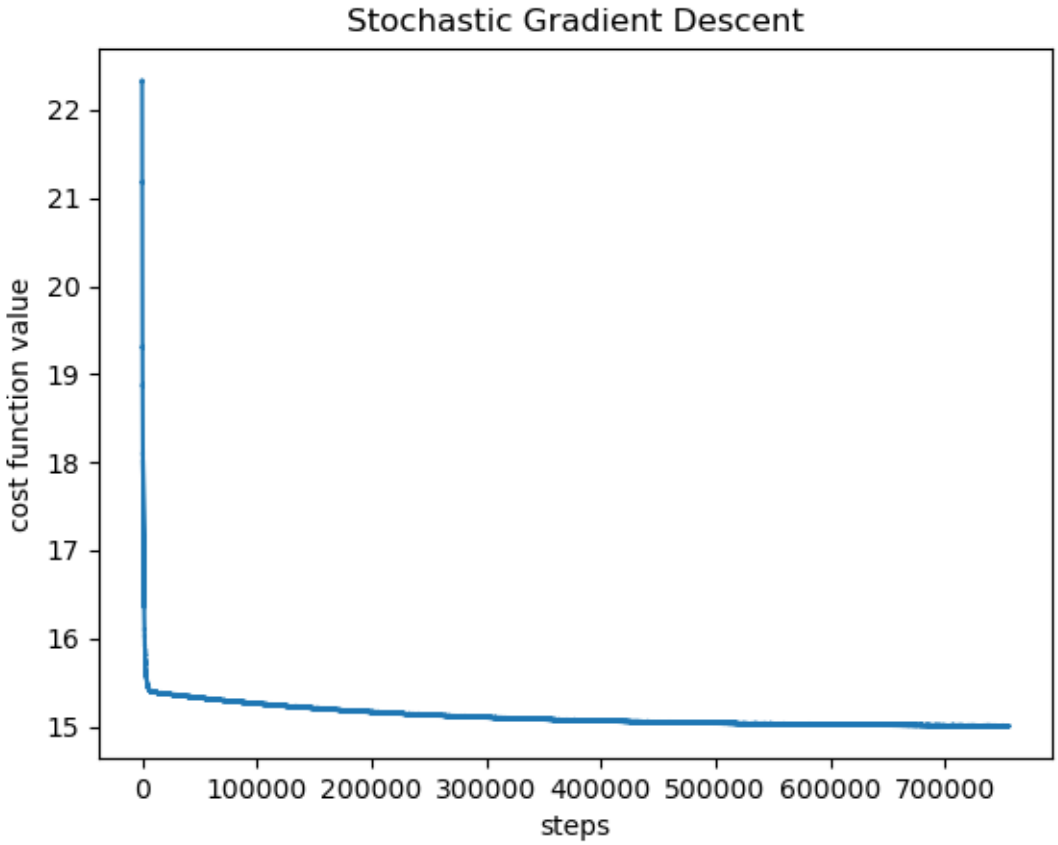
Learned weight vector: [0.92121924, 0.80795412, 0.87360641, 1.31402335, 0.13386879, 1.59860304, 1.01995499]
Learning Rate: 0.0125



(b)

Learned weight vector: [0.72954178 0.60583116, 0.65623542, 1.16051283, 0.10111029, 1.33361719, 0.8242626]

Learning Rate: 0.001



(c)

Learned weight vector: [0.92154947, 0.80829428, 0.87397433, 1.3142877, 0.13392374, 1.59904727, 1.02029192]

The learned weight vector calcualted with an analytical form is very similar to the weight vector from the batch gradient descent method. By comparing all three weight vectors it seems that the stochastic gradient descent weight vector consistently returns the lowest weights. This could be because the other two methods will always return around the same numbers, where the stochastic approach could get lucky and find the most optimal result randomly.