CS155: Set 1

Abraham Hussain

January 9, 2017

1 Basics

Question A: A hypothesis set is the set of all possible hypothesis that one must use to best approximate the unknown target function.

Question B: The hypothesis set of a linear model is in the form y = ax + b.

Question C: Overfitting is the phenomenon when fitting the observed data no longer indicates that one will get a decent out-of-sample error.

Question D: Two ways of preventing overfitting are: 1) Use a validation set by splitting data to training set and validation set and evaluate on the validation set. 2) Regularization which reduces the bias by slightly increasing the variance.

Question E: The training data is the data that we use to train the ML algorithm and the test data is the data we use to determine how accurate the ML algorithm is. The main difference is that the test data is just used to see how well the algorithm works based on the training data. You should never change your data based on information from the test data is because this would be considered data snooping and this would increase the VC dimension.

Question F: The two assumptions we make about how the data in our data set were sampled are that the data was chosen at random and that the training sets are from the same distribution as the validation set.

Question G: The input space could be a word of bags that contain a feature vector of the words in the email that we will look at keywords and phrases and the output would be either -1 if we deny it as spam and +1 if it is spam.

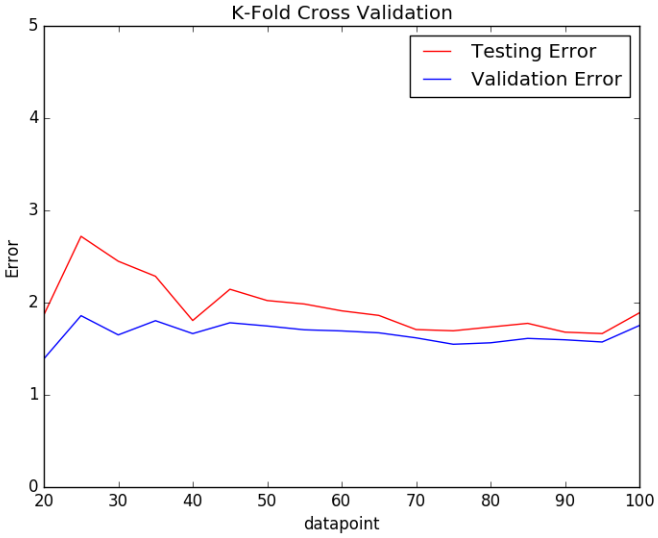
Question H: The K-fold cross-validation process is dividing the data set into k subsets. From this where we use k-1 partitions to train the algorithm and 1 to test the data.

2 Bias-Variance Tradeoff

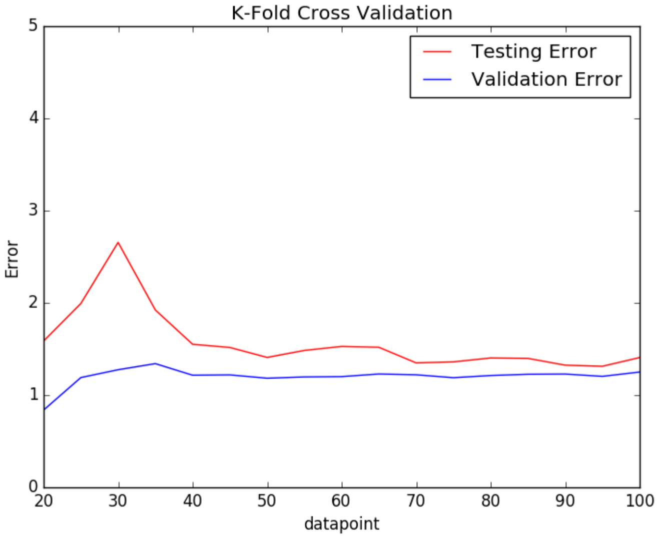
Question A: In order to show this we will $E_S[E_{out}(f_S)] = E_S[E_x[f_S(x) - y(x))^2]]$ and from this we can switch the orders of E_S and E_x such that it will now equal $E_S[E_{out}(f_S)] = E_x[E_S[f_S(x) - y(x))^2]]$ This will now equal $E_x[E_S[f_S(x) - F(x) + F(x) - y(x))^2]] = E_X[E_S[(f_S(x) - F(x))^2 + (F(x) - y(x))^2 + 2(f_S(x) - F(x))(F(x) - y(x))]] = E_x[E_S[(f_S(x) - F(x))^2] + (F(x) - y(x))^2]$ and if we look at the form that the manipulated left hand side takes, it is the same as that of the right hand side since we can now change it to $E_x[Bias(x) + Var(x)]$.

Question B:
The blue represents the validation error while the the green represents the training error.

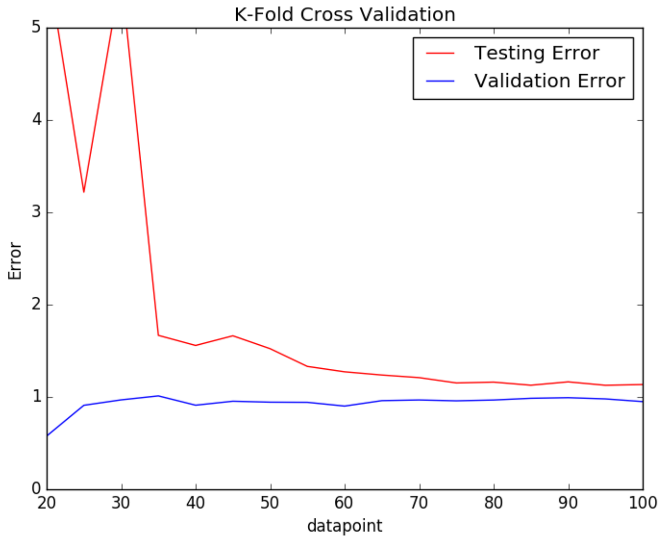
Degree 1



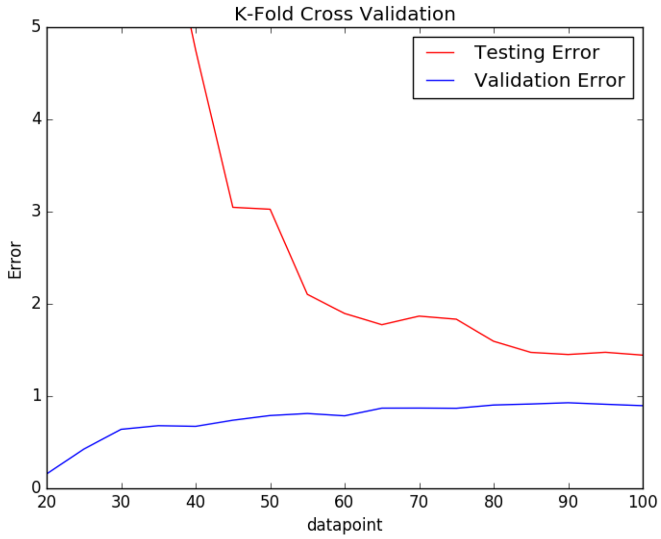
Degree 2



Degree 6



Degree 12



Question C: Based on the learning curves degree 1 has the highest bias because the level of convergence is higher.

Question D: Based on the learning curves degree 12 has the highest variance because the training error is lower than the variation error compared to the other degrees.

Question E: It tells us that adding more training points will not help since the training and validation error have already converged.

Question F: The training error is generally lower than the validation error because we use the training data to make our model so it has more opportunities to tune the model such that the training data produces fits it better causing us to see this trend.

Question G: Degree 6; Because we will want to minimize the sum of the variance and the bias and if we look at the graphs, this has the smallest area between the training and validation error and converges the quickest.

3 The Perceptron

Question A:

t	b	w$_1$	w$_2$	x$_1$	x$_2$	y
0	0	0	1	1	-2	1
1	1	1	-1	1	-2	1
2	2	1	2	0	3	1
3	3	2	0			

Question B: In a 2D data set, the smallest that is not separable would be a data set of 4. For N-dimensional sets it would be N + 2.

Question C: If a data set is not linearly separable then the Perceptron Learning Algorithm will never converge, because it will never stop trying to modify itself such that it could split all of the data. The only way that it can converge is if it is modified so that it would accept some classifications.

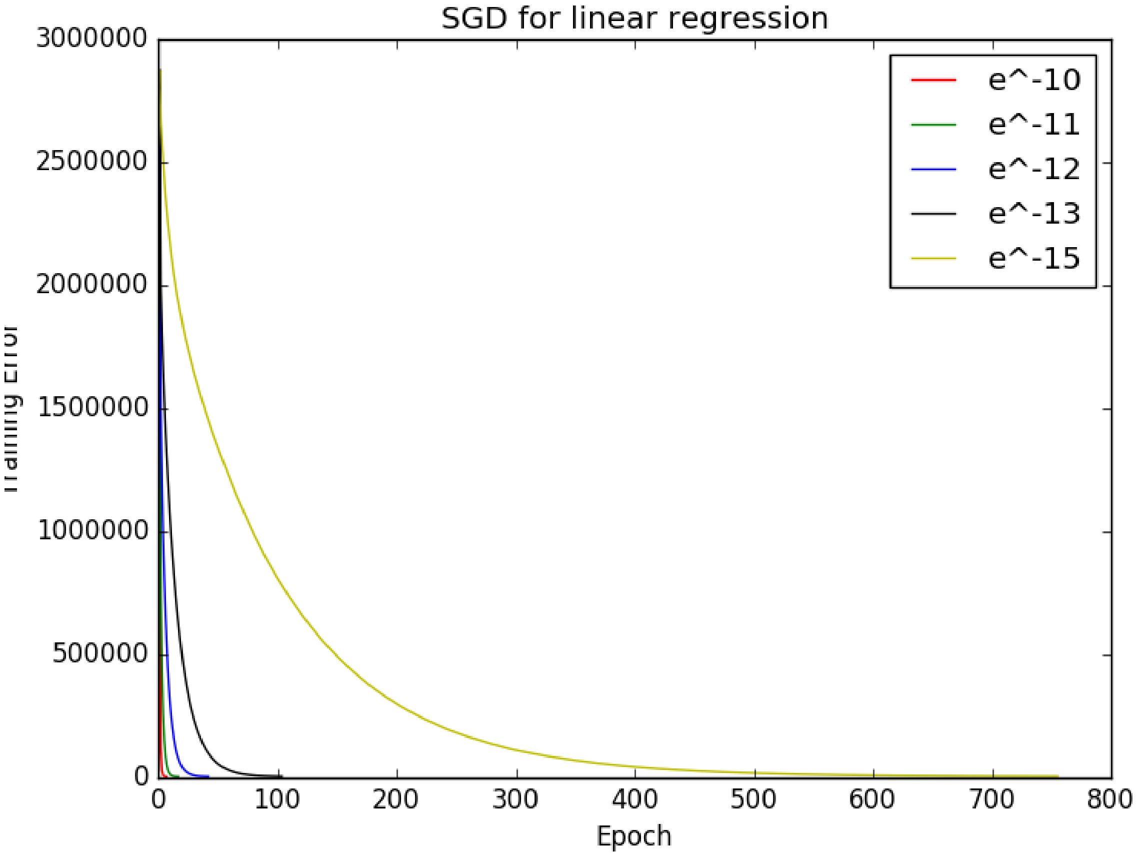
4 Gradient Descent

Question A: In order to include the bias term in this form we must put $w_o = 0$ and $x_o = 1$ so they can be valued as 0 and 1

Question B: We will start by stating the original loss function: L = $\sum_{i=1}^{N}(y_i - w^T x_i)^2$. Now we must find the gradient of the loss function: $\partial_w \sum_{i=1}^{N}(y_i - w^T x_i)^2 = \sum_{i=1}^{N} \partial_w (y_i - w^T x_i)^2 = \sum_{i=1}^{N} -2(y_i - w^T x_i) * \partial_w w^T x_i = \sum_{i=1}^{N} -2x_i(y_i - w^T x_i)$

Question C: Weights: [-0.22917907 -5.92559652 3.92726648 -11.66675541 8.72920006]

Question D:



The reason for this is that the larger step size changes the weight faster given the update function for the weight.

Question E:
$[-0.31644251, -5.99157048, 4.01509955, -11.93325972, 8.99061096]$ The analytical solution roughly matches up with the weights we got from the SGD, however it is off by around 0.1 to 0.2.

Question F: The reason why SGD would be better than the closed form cases is because if the data/matrix it too large, a computer will not be able to compute the closed form formula as well as it would the SGD method.

Question G: If SGD is used on a non-convex learning problem, then it would most likely find a local minimum that will not represent what SGD is suppose to find. Therefore this would not be an accurate representation of the true global minimum.