Machine Learning & Data Mining
Caltech CS/CNS/EE 155
Homework 1

Changhao Xu
UID: 2103530
January 13th, 2020

1 Basics

Question A: Hypothesis set is a set of hypothesis/functions that approximates an unknown target function.

Question B: Hypothesis set of linear model is in the form of f(x|w, b) = w^{T}x + b.

Question C: Overfitting occurs when the model function fits too closely to a limited set of data points, which results in a large out-of-sample error while in-sample error is small.

Question D: Validation & Regularization.

Question E: Training data is the data set that is used for training a model. Testing data is the data set that is used for evaluating the trained model's performance. We should never change the model based on test data because that will make test data become training data, and cannot indicate out-of-sample performance any more.

Question F: i.i.d., i.e. Data point is independent and identically distributed.

Question G: Input space X can be a 'bag of words' feature vector extracted from the email contents, and Y can be the result whether this email is spam or not (let +1 when the email is spam and -1 when the email is not spam).

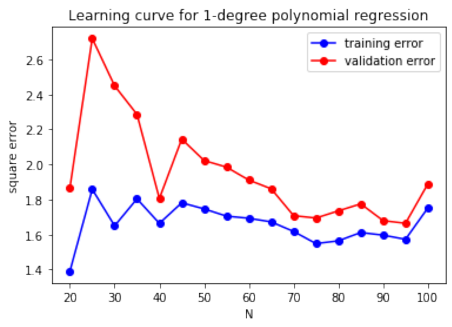
Question H: K-fold cross-validation procedure: 1) Split data into k equal partitions, 2) Train on (k-1) partitions and test one 1 partition, 3) Repeat the process for k times. This procedure allows re-using training data as test data and using all data as validation.

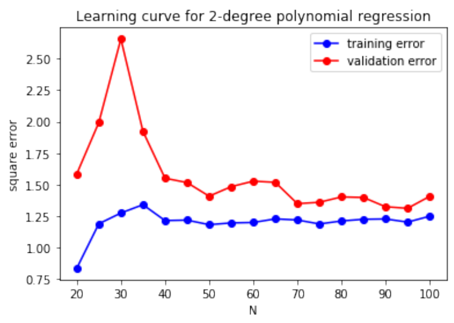
2 Bias-Variance Tradeoff

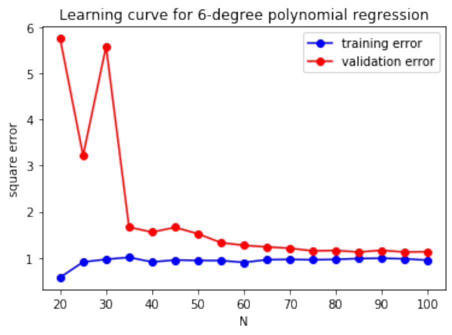
Question A:

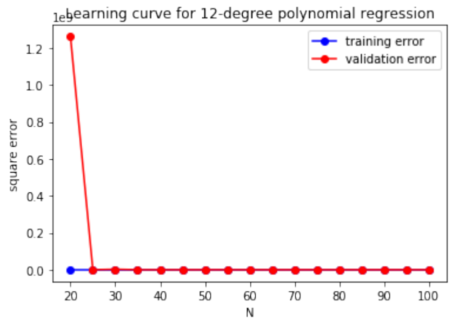
$$\begin{aligned}
\mathbb{E}_s\left[E_{out}(f_s)\right] &= \mathbb{E}_s\left[\mathbb{E}_x\left[(f_s(x)-y(x))^2\right]\right] \\
&= \mathbb{E}_x\left[\mathbb{E}_s\left[(f_s(x)-y(x))^2\right]\right] \\
&= \mathbb{E}_x\left[\mathbb{E}_s\left[(f_s(x)-F(x)+F(x)-y(x))^2\right]\right] \\
&= \mathbb{E}_x\left[\mathbb{E}_s\left[(f_s(x)-F(x))^2\right]\right]+\mathbb{E}_x\left[\mathbb{E}_s\left[(F(x)-y(x))^2\right]\right]+2\mathbb{E}_x\left[\mathbb{E}_s\left[(f_s(x)-F(x))(F(x)-y(x))\right]\right] \\
&= \mathbb{E}_x\left[Var(x)\right]+\mathbb{E}_x\left[Bias(x)\right]+2\mathbb{E}_x\left[(F(x)-y(x))\mathbb{E}_s\left[(f_s(x)-F(x))\right]\right] \\
&= \mathbb{E}_x\left[Var(x)\right]+\mathbb{E}_x\left[Bias(x)\right]+0 \\
&= \mathbb{E}_x\left[Bias(x)+Var(x)\right]
\end{aligned}$$

Question B:









Question C: 1st-degree polynomial has the highest bias, since it has the highest square error for both training and validation on average, which is a sign of underfitting.

Question D: 12th-degree polynomial has the highest variance, since it has the biggest validation error and small training error initially when data set is small, which is a sign of overfitting.

Question E: The model will not improve much with additional training points due to limited model complexity. As shown in the learning curve, the slope of both training error and validation error becomes small after reaching certain number of data points.

Question F: Training error is the error on the data set that has been used by the model. Since the model has used these data to minimize error of loss function, while validation data has not been used by the model, training error is generally lower than validation error.

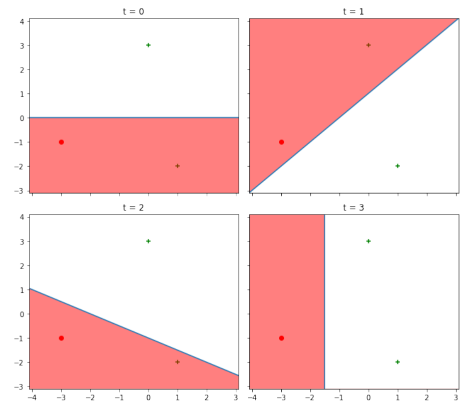
Question G: 6th-degree polynomial is expected to perform best on unseen data, since it has the lowest validation error and optimal bias-variance tradeoff, which is close to out-of-sample performance.

3 The Perceptron

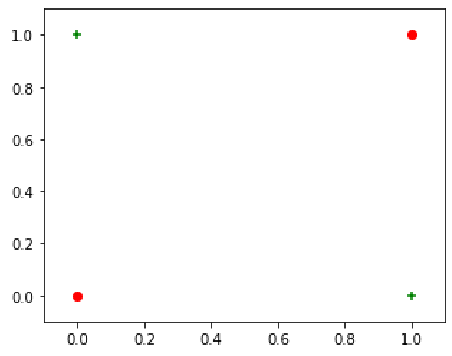
Question A: Code output as follows:

```
t = 0, b = 0, w = [0. 1.], [x1, x2, Y] = [array([ 1, -2]), 1]
t = 1, b = 1, w = [ 1. -1.], [x1, x2, Y] = [array([0, 3]), 1]
t = 2, b = 2, w = [1. 2.], [x1, x2, Y] = [array([ 1, -2]), 1]
t = 3, b = 3, w = [2. 0.], [x1, x2, Y] = []

final w = [2. 0.], final b = 3.0
```



Question B: In a 2D data set, 4 points are linearly inseparable:



In a 3D data set, 5 points are linearly inseparable: suppose any 3 points form a plane, and let these 3 points be +1; let one point at one side of the plane be -1, and one point at the other side of the plane be -1. The resulting feature is linear inseparable.
In a N-dimensional data set, (N+2) points will be linearly inseparable.

Question C: When data set is not linearly seperable, PLA will keep finding a hyperplane that classifies all data points correctly, which doesn't exist, so PLA will not converge.

4 Stochastic Gradient Descent

Question A: We can simply let w_0 =b, and x_0 = 1, so that **w** = (w_0, w_1, w_2, ..., w_d), **x** = (1, x_1, x_2, ..., x_d).

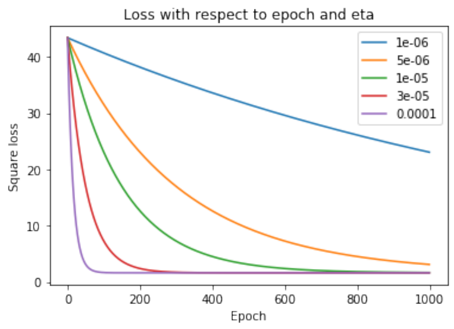
Question B:

$$\partial_w \sum_{i=1}^{N} (y_i - \mathbf{w}^T x_i)^2 = \sum_{i=1}^{N} -2x_i(y_i - \mathbf{w}^T x_i)$$

Question C: Please see Jupyter Notebook.

Question D: SGD converges at the same minimum point regardless of varying starting point, though the converging rate varies between different starting points. Different datasets have different converging rate, but they all converge towards a minimum point.

Question E: As η increases, SGD converges faster, but when η becomes very big, SGD cannot converge

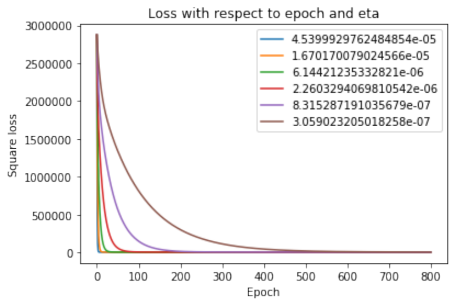


any more.

Changhao Xu
UID: 2103530
January 13th, 2020

Question F: Final weights **w** = [-0.22717707 -5.94209998 3.94391318 -11.72382875 8.78568403]

Question G: As η increases, SGD converges faster. (When η becomes very big, SGD cannot converge



any more. But here η overall is small.)

Question H: Final weights **w** = [-0.31644251 -5.99157048 4.01509955 -11.93325972 8.99061096]. The result is not the same, but quite similar with SGD.

Question I: SGD may not give the exact same answer as closed form solution, but it is very efficient to get an approximate solution, especially when dataset is large in reality.

Question J: Stopping condition: store the losses after each epoch. When the relative loss decrease becomes very small (e.g. $<$0.0001), stop the epoch.

Question K: SGD is smoothier compared with PLA. SGD uses gradient descent to guide weight modification, while perceptron simply goes from one point to another and may change weight abruptly.