* 2 late hours used

CS 155 PS 1

1 Basics

Problem A
A hypothesis set is a set of candidate formulas from which the learning algorithm will choose the final hypothesis to approximate the target function.

Problem B
The hypothesis set of a linear model is the set of all formulas in the form $f(x \mid w, b) = w^T x - b$ with any values of w and b.

Problem C
Overfitting is fitting into a specific set of training data too much that the learned hypothesis has bad generalization. In other words, test error would be much larger than the training error when overfitting occurs.

Problem D
We can prevent overfitting by increasing the size of training data or early-stopping using validation.

Problem E
Training data and test data come from the same distribution, but training data is used by the learning algorithm as a sample of data to learn a hypothesis from, while test data is used to check how the learned hypothesis performs. You should never change your model based on information from the test data because fitting to specific points in the test data will result in bad generalization rather than improving the model.

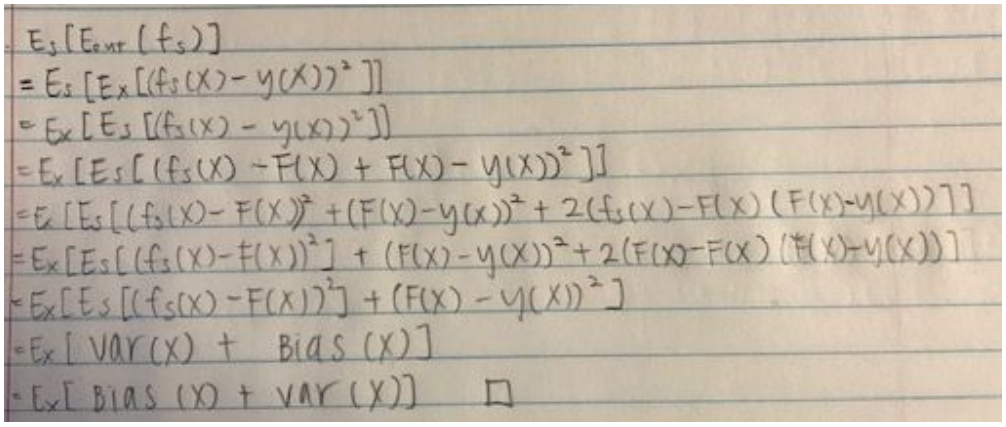
Problem F
We assume that the dataset is sampled randomly and approximates the true distribution of the population we are trying to learn from.

Problem G
The input space could be bags of words, where vectors each have one feature (1 or 0 denoting the presence of the word) for each word in the vocabulary. The output space could be $\{-1,+1\}$ to indicate whether it is spam or not.

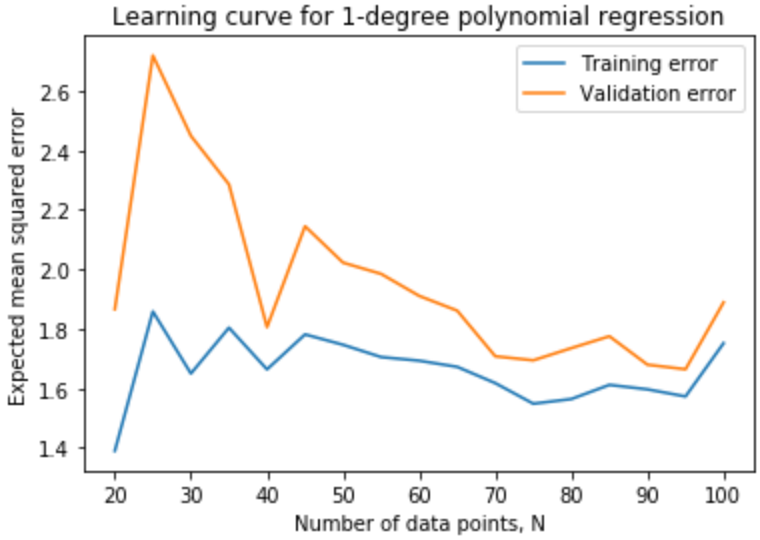
Problem H
The k-fold cross-validation procedure is a validation method in which the data is split into k partitions and each partition is used as a test set while the remaining partitions are used as training set. They are sampled independently, allowing the training data to be reused as test data and the entire dataset to be used as validation.

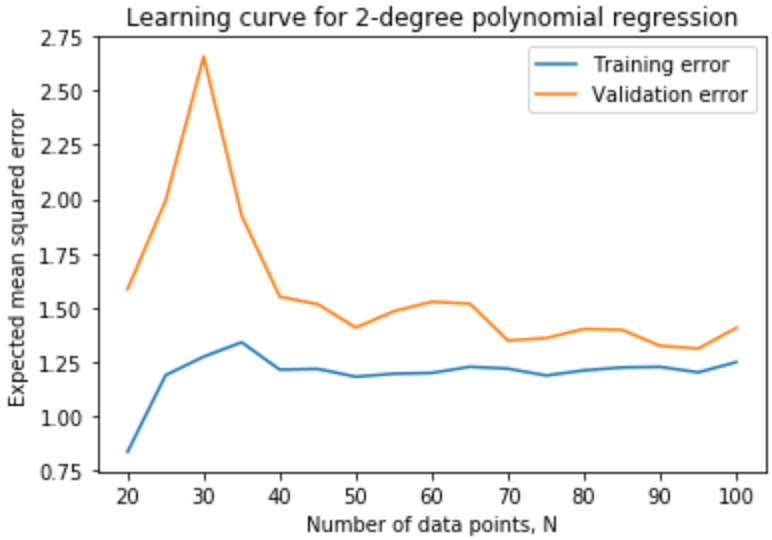
2 Bias-Variance Tradeoff

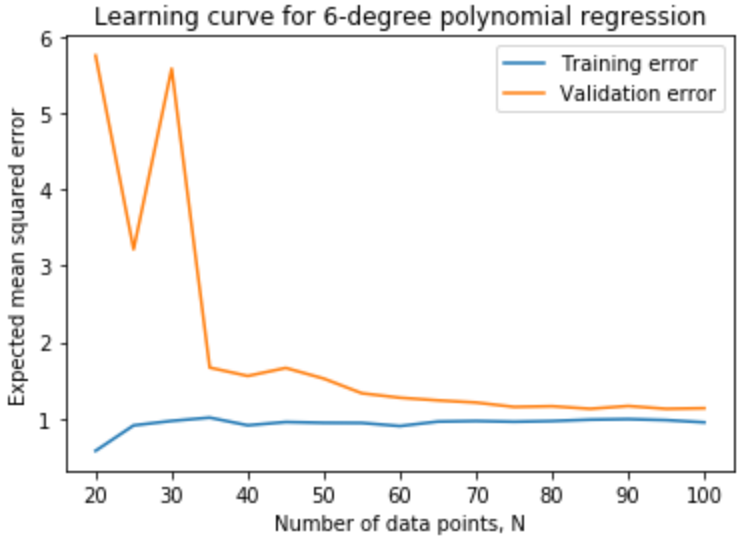
Problem A

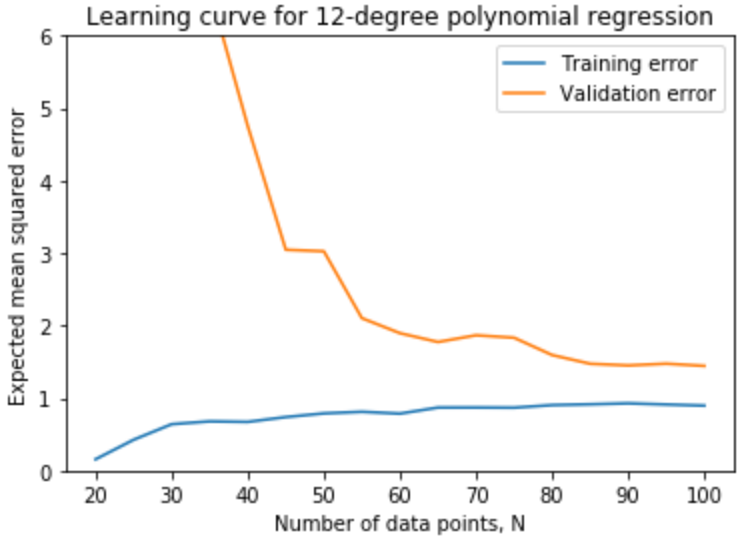


Problem B




Learning curve for 2-degree polynomial regression
Training error
Validation error
Expected mean squared error
Number of data points, N


Learning curve for 6-degree polynomial regression
Training error
Validation error
Expected mean squared error
Number of data points, N


Learning curve for 12-degree polynomial regression
Training error
Validation error
Expected mean squared error
Number of data points, N

Problem C
The 1-degree polynomial regression model has the highest bias, as the training error and the validation error converge to the highest value out of the 4 models. A large value of this inevitable error indicates a high bias, which reflects the limitation of the model in approximating the target function.

Problem D
The 12-degree polynomial regression model has the highest variance, as the validation error is extremely high with a few data points but reduces with more data points.

Problem E
The learning curve of the quadratic model shows that the model won't improve by a lot even if we had more training points, as the curve plateaus after around 50 data points.

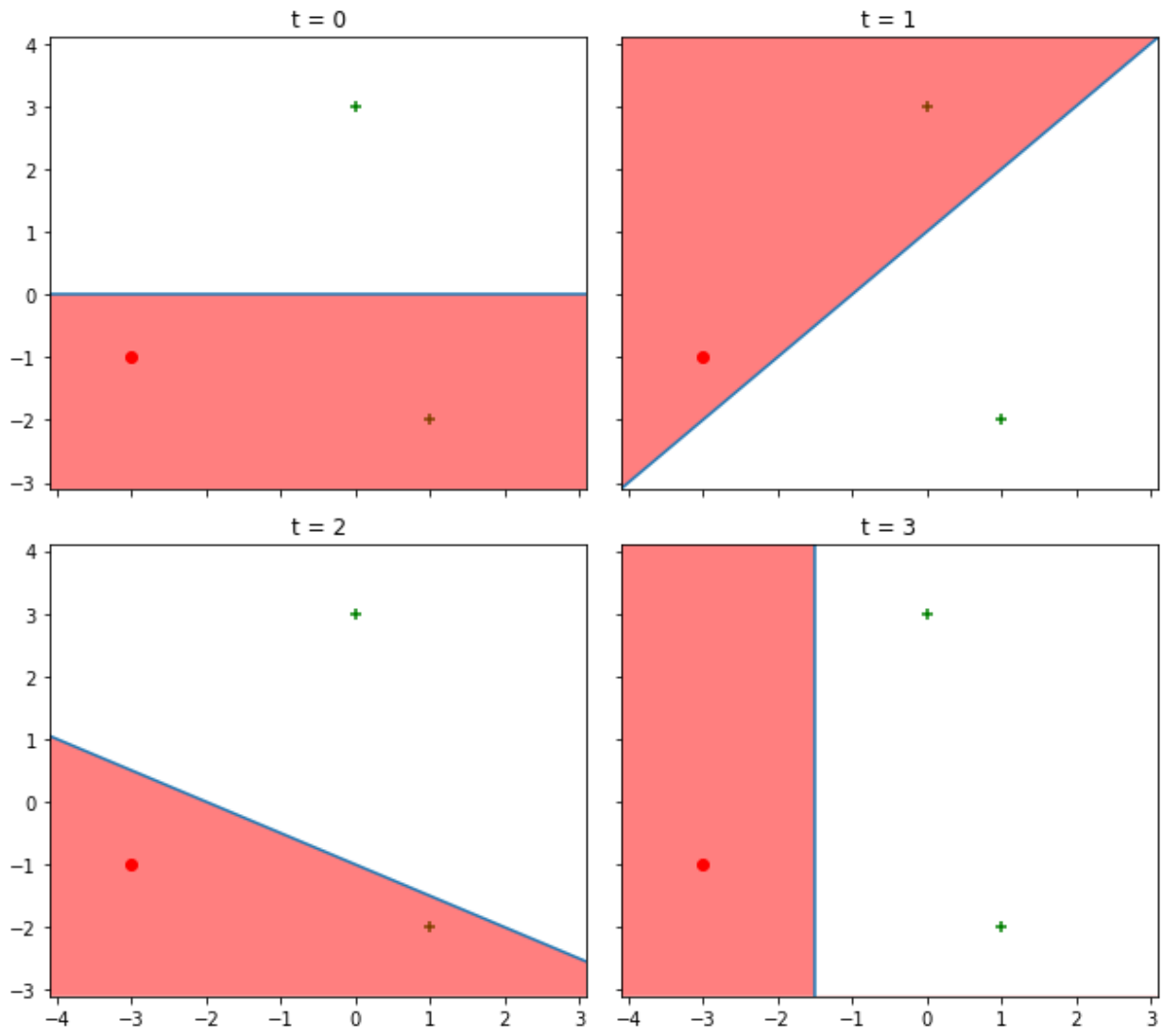
Problem F
Training error is generally lower than validation error because the model is trained on training data (meant to fit training data better), and it is impossible for training data to accurately represent the entire distribution. Validation error is calculated from the uncontaminated test data in each fold, so it will be higher than training error (the extent of which varies by its generalizing ability).

Problem G
I would expect the 6-degree model to perform best, since its validation error is the lowest with a sufficient number of data points meaning it would generalize the best to other sets of data from the same distribution.

3 The Perceptron

Problem A

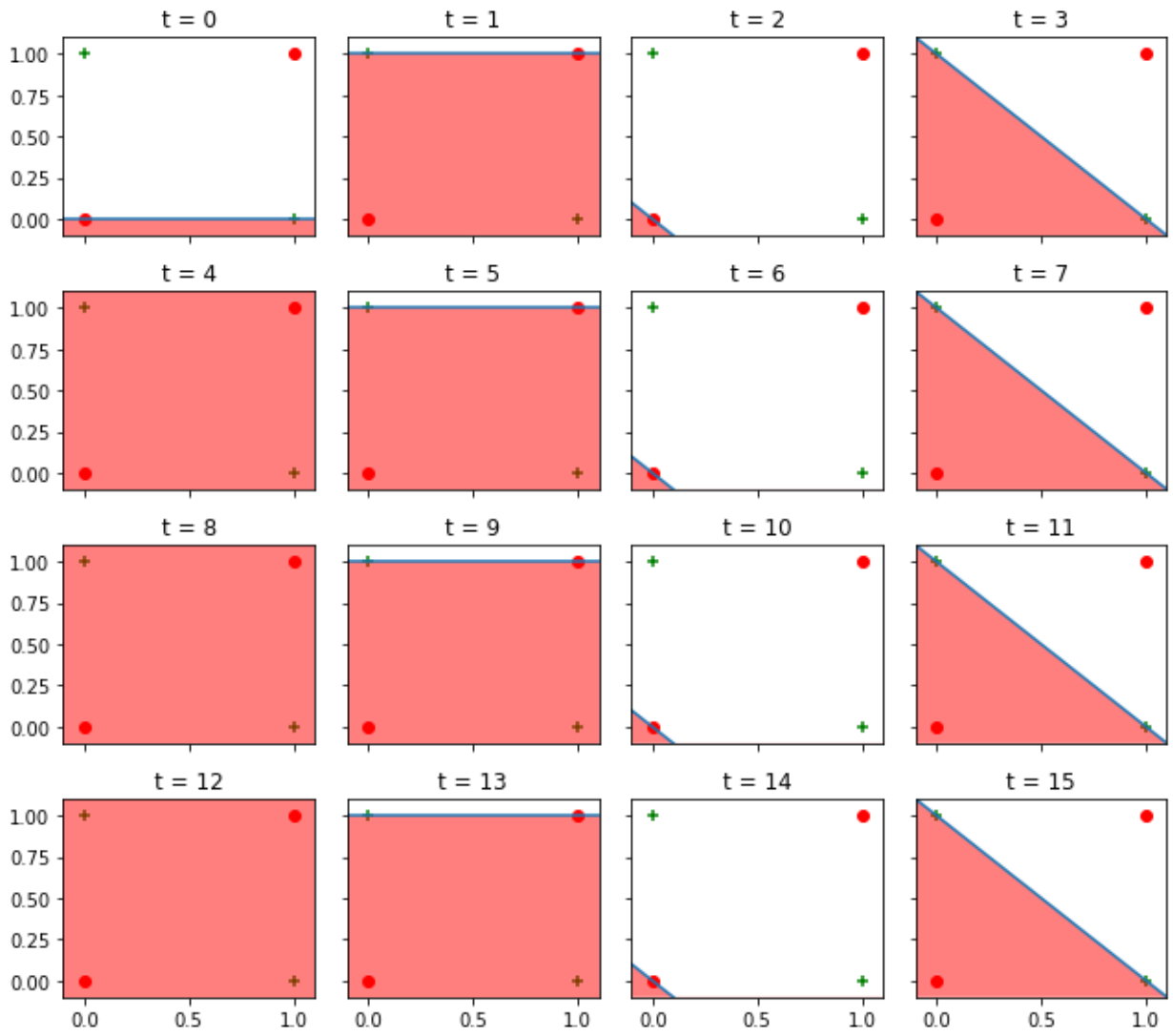
t	b	w1	w2	x1	x2	y
0	0.0	0.0	1.0	1	-2	1
1	1.0	1.0	-1.0	0	3	1
2	2.0	1.0	2.0	1	-2	1
3	3.0	2.0	0.0			



Problem B

A dataset of 4 points is the smallest that is not linearly separable in 2D input space, since any two points in a 2D space would be collinear, and if a line formed by two points of the same sign divides two points of a sign opposite from them, there is no way to separate such four points. A dataset of 5 points is the smallest that is not linearly separable in 3D input space, since any three points in a 3D space would be coplanar, and if a plane formed by three points of the same sign divides two points of a sign opposite from them, there is no way to separate such five points. For an N-dimensional set, a dataset of N + 2 points is the smallest that is not linearly separable.

Problem C



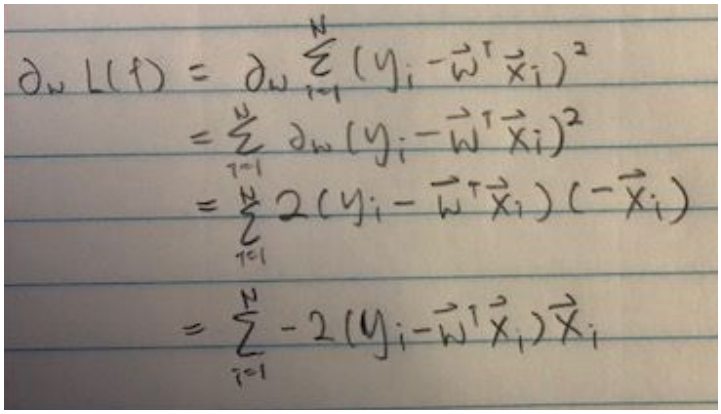
The Perceptron Learning Algorithm will never converge because it is impossible to fit a hyperplane dividing the linearly inseparable points, and there will always be misclassified points that the algorithm tries to classify correctly by infinitely updating the weights and bias.

4 Stochastic Gradient Descent

Problem A

We can add a constant element in **x** (i.e. $x_0 = 1$) and a corresponding element w_0 in **w** such that w_0 represents the bias term b.

Problem B

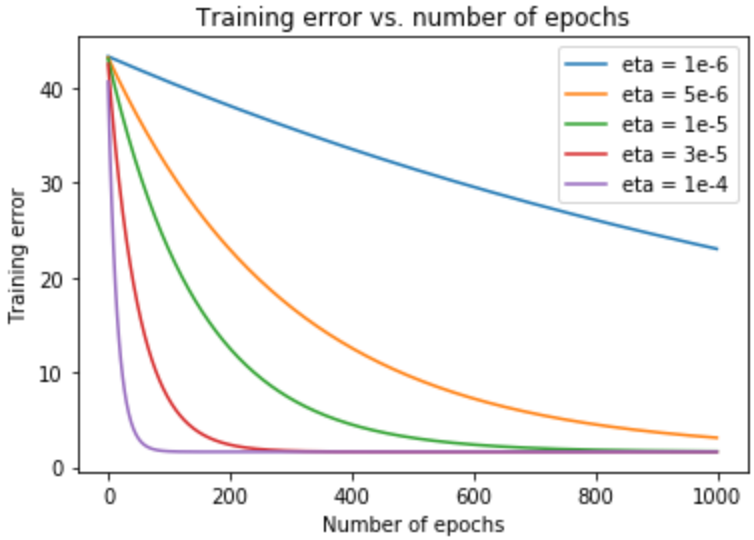


$$\partial_w L(f) = \partial_w \sum_{i=1}^{N} (y_i - \vec{w}^T \vec{x}_i)^2$$
$$= \sum_{i=1}^{N} \partial_w (y_i - \vec{w}^T \vec{x}_i)^2$$
$$= \sum_{i=1}^{N} 2(y_i - \vec{w}^T \vec{x}_i)(-\vec{x}_i)$$
$$= \sum_{i=1}^{N} -2(y_i - \vec{w}^T \vec{x}_i)\vec{x}_i$$

Problem D

Regardless of the starting point, it always converges to the same point (global minimum). It also reaches the global minimum in a straight line without any divergence. This behavior is also consistent between datasets 1 and 2.

Problem E

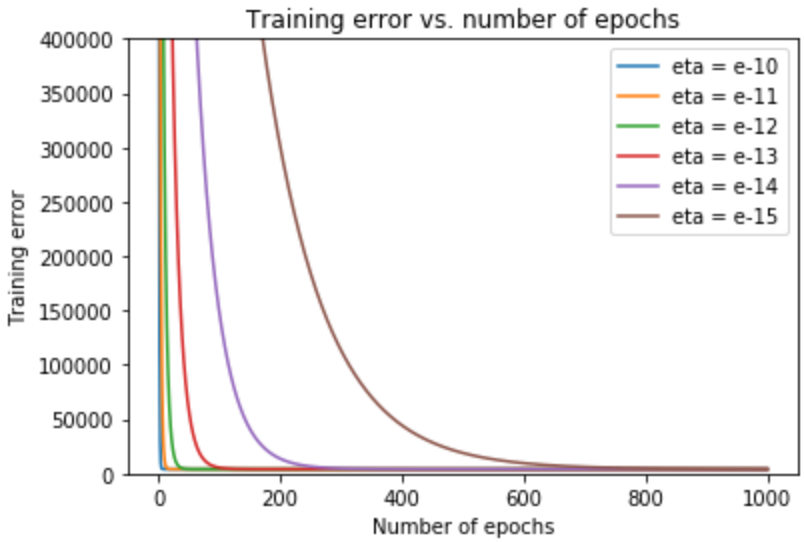


SGD converges faster (takes less epochs) as η increases. With low values of η it doesn't converge within 1000 epochs.

Problem F

The weight vector is [-5.97854231 3.98838757 -11.85700955 8.91129286] and bias is -0.22788575.

Problem G



SGD converges faster (takes less epochs) as η increases. Even the highest eta value isn't too large that it diverges. For small step sizes, it takes longer for the SGD to converge since less progress is made in each epoch.

Problem H

The weight vector is [-5.99157048 4.01509955 -11.93325972 8.99061096] and bias is -0.31644251. This result is very close to the values from SGD.

Problem I

Computing the closed form solution could get very expensive as the size of the data or the input dimension increases. In addition, parallelization can be relatively easily performed with SGD.

Problem J

The SGD convergence plots show that the training error asymptotically reaches 0 as the number of epochs increases. We can stop when the change in training error (or validation error) is sufficiently small, which indicates that progress is sufficiently small.

Problem K

The weight vector fluctuates up and down as it converges (if it does) in the perceptron algorithm, while in SGD it is guaranteed to gradually and smoothly converge to some point without diverging unless the step size is too large.