ECE M148 Homework 3 Solutions
Introduction to Data Science Due: May 03, 12:00 PM
Instructor: Lara Dolecek TAs: Harish GV, Jayanth Shreekumar

Please upload your homework to Gradescope by May 03, 12:00 PM. You can access Gradescope directly or using the link provided on BruinLearn. You may type your homework or scan your handwritten version. Make sure all the work is discernible.

1. Assume you have a dataset $\mathcal{D}$ with n samples. You want to create bootstrapped datasets of size k using sampling with replacement.

   (a) Assume you create one bootstrapped dataset of size k. Additionally, assume that we fix a data point $x \in \mathcal{D}$. What is the probability that x does not appear in the bootstrapped dataset?

   (b) Now, assume that $k = n$. What does the probability converge to as n goes to infinity? What does this limit imply about the percentage of the original dataset that will not be sampled at n gets large?

   (c) Assume that you create r bootstrapped datasets of size k each. Additionally, assume that we fix a data point $x \in \mathcal{D}$. What is the probability that x does not appear in any bootstrapped dataset?

   Solution:

   (a) Let us consider taking one sample at a time to populate the bootstrapped dataset. The probability that x is not chosen for this first sample is $\frac{n-1}{n}$ since there are $n-1$ other points to choose from.

   Since we are sampling with replacement, all the samples are independent from each other. Thus, the probability that x is not in the bootstrapped dataset is

$$\begin{aligned} P(\text{ x is not sampled by the dataset }) &= P(\cup_{i=1}^{k} \text{x is not the ith sample}) \\ &= \prod_{i=1}^{k} P(\text{ x is not the ith sample}) \\ &= \prod_{i=1}^{k} \frac{n-1}{n} = \prod_{i=1}^{k} (1 - \frac{1}{n}) \\ &= (1 - \frac{1}{n})^k \end{aligned}$$

   (b) For the limit part, $\lim_{n\to\infty}(1 - \frac{1}{n})^n = e^{-1} = \frac{1}{e} \approx 0.36788$. This means that the dataset will not have around e^{-1} percent of the data points from the original dataset.

(c) Again, we can say that the sampling of each dataset is independent of each other. So the probability that x is not in any bootstrapped datasets is

$$\begin{aligned} P(\text{ x is not sampled by any dataset }) &= P(\cup_{j=1}^{r} \text{x is not in the jth dataset}) \\ &= \prod_{j=1}^{r} P(\text{x is not in the jth dataset}) \\ &= \prod_{i=1}^{r} (1 - \frac{1}{n})^k \\ &= ((1 - \frac{1}{n})^k)^r = (1 - \frac{1}{n})^{rk} \end{aligned}$$

2. In this question, let us consider the difference between lasso and ridge regularization. Recall that the lasso regularization of a vector β is $\lambda \sum_{i=1}^{k} |\beta_i|$ and that the ridge regularization is $\lambda \sum_{i=1}^{k} \beta_i^2$. Consider two vectors $x_1 = [4, 5]$ and $x_2 = [-2, 2]$. Additionally, set $\lambda = 1$.

   (a) What is the lasso regularization of x_1 and x_2? What is the change in the lasso regularization when going from x_1 to x_2?

   (b) What is the ridge regularization of x_1 and x_2? What is the change in the ridge regularization when going from x_1 to x_2?

   (c) In your own words, explain the effects of ridge vs lasso regularization.

   Solution:

   (a) Let $L_1(x)$ be the lasso regularization of vector x.

   $$L_1(x_1) = |4| + |5| = 9, L_1(x_2) = |-2| + |2| = 4$$

   $$L_1(x_1) - L_1(x_2) = 5$$

   (b) Let $L_2(x)$ be the ridge regularization of vector x.

   $$L_2(x_1) = (4)^2 + (5)^2 = 41, L_2(x_2) = (-2)^2 + (2)^2 = 8$$

   $$L_2(x_1) - L_2(x_2) = 33$$

   (c) Both regularization techniques aim to prevent overfitting in the model.

   Ridge regression reduces the magnitude of the parameters but never reduces them to zero because it prioritizes decreasing larger parameters over smaller parameters.

   Lasso regression creates a sparse model by setting some model parameters to zero because lasso regression treats all decreases equally regardless if the parameter is large or small. Example: $|2| - |1| = 1$ and $|3| - |2| = 1$ while $2^2 - 1^2 = 3$ and $3^2 - 2^2 = 5$. Lasso regularization is problematic when data samples are highly collinear and it will randomly select which one to zero out.

3. **Coding Question** - Plot the Voronoi regions for $k = 1, 2, 3, 4$ using the k-nearest neighbours classifier on the points: [[1, 1], [4, 1], [2, 3], [3, 3], [3, 4], [5, 4], [6, 5], [4, 5]]. The first 4 points are in class 0 and the rest are in class 1. A .ipynb file has been provided with starter code to get you started. Did you find anything curious about the plots? How do you explain them?

   Solution:

In[1]:
```
x = np.array([[1, 1], [4, 1], [2, 3], [3, 3], [3, 4], [5, 4], [6,
↪  5], [4, 5]])
y = np.array([0, 0, 0, 0, 1, 1, 1, 1])
k_r  = [1,2,3,4]
for k in k_r:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(x, y)
    draw_contour(x,y,knn)

    plt.title(f"K ={k}")
```

   We see that for k = 2, a green point is in the midst of the pink Voronoi region. This can be explained by the fact that there is a tie, and there are many ways to explain this. One way would be to simply randomly assign a class, another would be to default to a class. More complex ways would involve finding the distances and picking the class that had the least average distance. Any reasonable solution would suffice. The same reasoning holds for k = 4.

4. **Coding Question** - Plot the logistic function $\frac{1}{1+e^{-(\beta_0+\beta_1 \times x)}}$ for $x \in [-10, 10]$ and the following parameter values:

   (a) $\beta_0 = 2$ and $\beta_1 = 1$

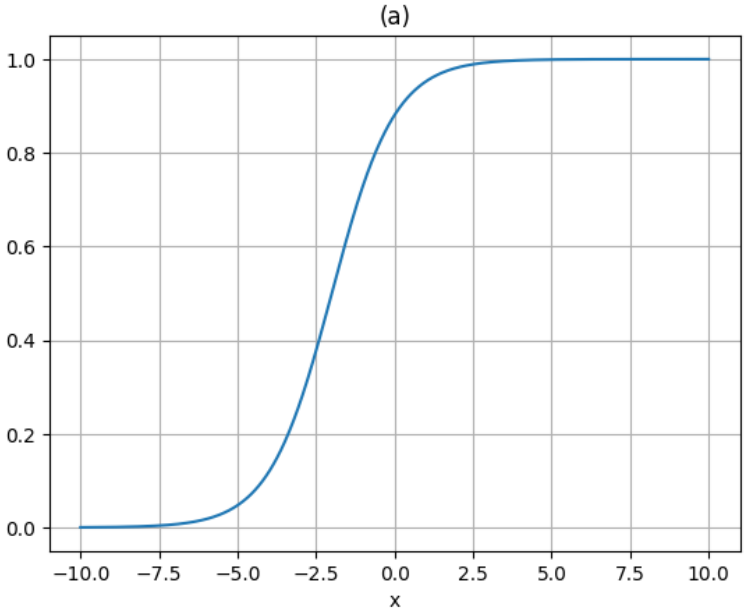
   (b) $\beta_0 = 10$ and $\beta_1 = 2$

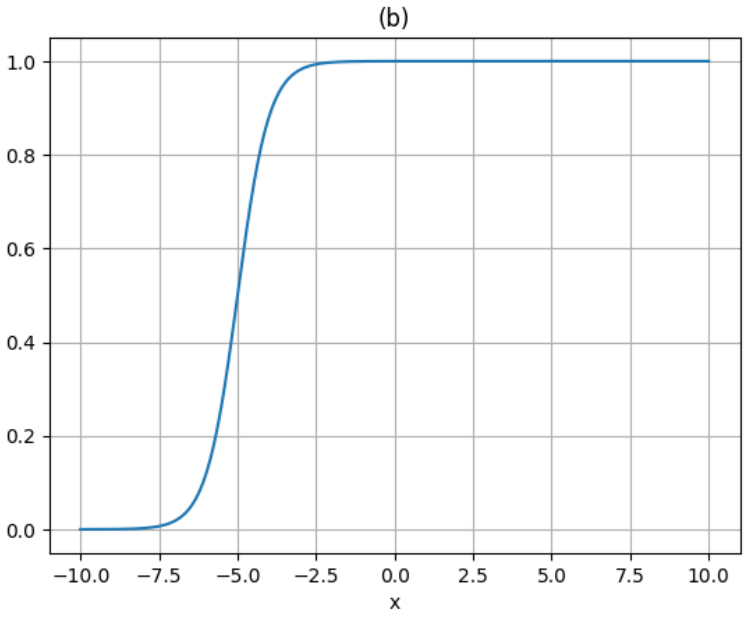
   (c) $\beta_0 = 1$ and $\beta_1 = 10$

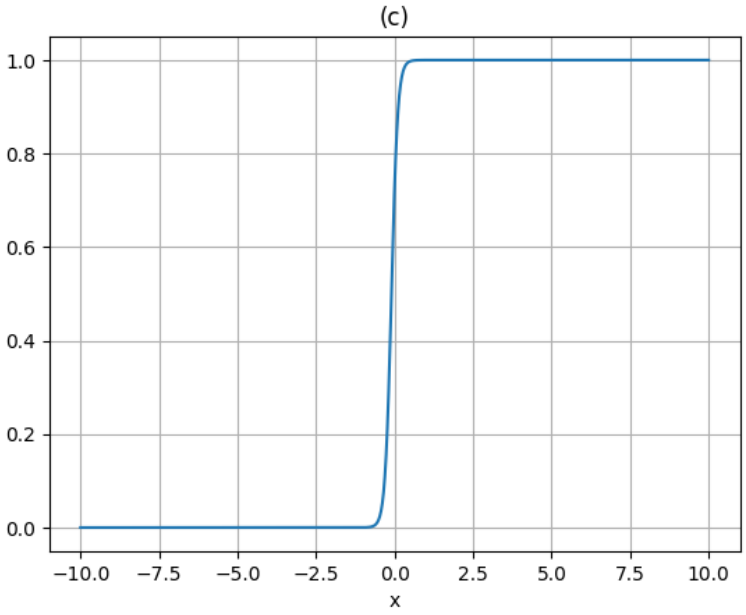
   (d) $\beta_0 = 1$ and $\beta_1 = 5$

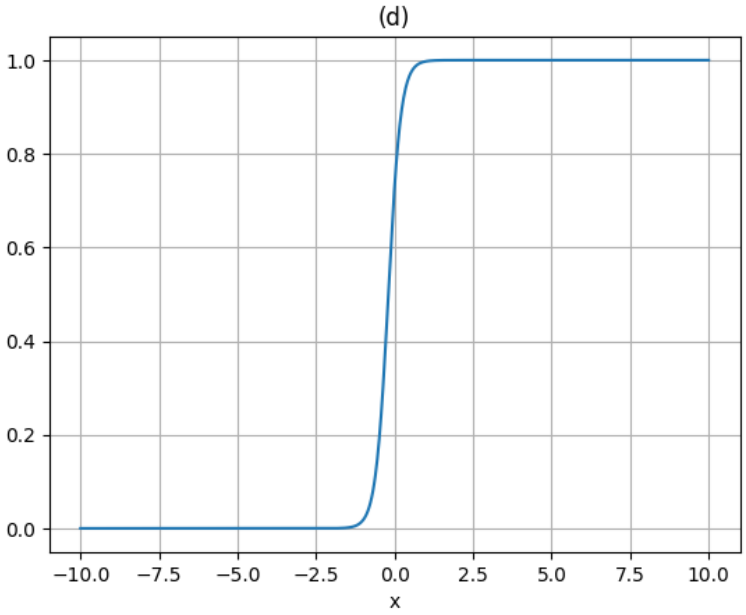
   For what choices of β_0, β_1 does the function become steeper?

   Solution:









As β_1 gets larger, the slope gets steeper. Thus, $\beta_0 = 1$ and $\beta_1 = 5$ has the largest slope.

CODE

In[2]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In[3]:
```
def func(x,beta0,beta1):
    return 1/(1+np.exp(-(beta0+beta1*x)))
```

```
x= np.linspace(-10,10,1000)
```

In[4]:
```
plt.plot(x,func(x,2,1))
plt.grid()
plt.title("(a)")
plt.xlabel("x")
plt.savefig("q2_a.png")
```

In[5]:
```
plt.plot(x,func(x,10,2))
plt.grid()
plt.title("(b)")
plt.xlabel("x")

plt.savefig("q2_b.png")
```

In[6]:
```
plt.plot(x,func(x,1,10))
plt.grid()
plt.title("(c)")
plt.xlabel("x")

plt.savefig("q2_c.png")
```

In[7]:
```
plt.plot(x,func(x,1,5))
plt.grid()
plt.title("(d)")
plt.xlabel("x")

plt.savefig("q2_d.png")
```

5. Recall the problem of ridge linear regression with n points and k features:

$$L_{Ridge}(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{\beta}^T\boldsymbol{x}_i)^2 + \lambda\sum_{j=1}^{k}\beta_j^2$$

where λ is a hyper-parameter. The goal is to minimize $L_{Ridge}(\boldsymbol{\beta})$ in terms of $\boldsymbol{\beta}$ for a fixed training dataset $(y_i, \boldsymbol{x}_i)$ and parameter λ.

(a) In your own words, explain the purpose of using ridge regression over standard linear regression.

(b) As λ gets larger, how will this affect $\boldsymbol{\beta}$? What value do we expect $\boldsymbol{\beta}$ to converge on?

(c) Consider parameters $\boldsymbol{\beta}_\lambda$ that were trained using ridge linear regression with a specific lambda. Let us consider the test MSE using $\boldsymbol{\beta}_\lambda$. Note that the test MSE is the following formula for the test data

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{\beta}_\lambda^T\boldsymbol{x}_i)^2$$

and does not include regularization.

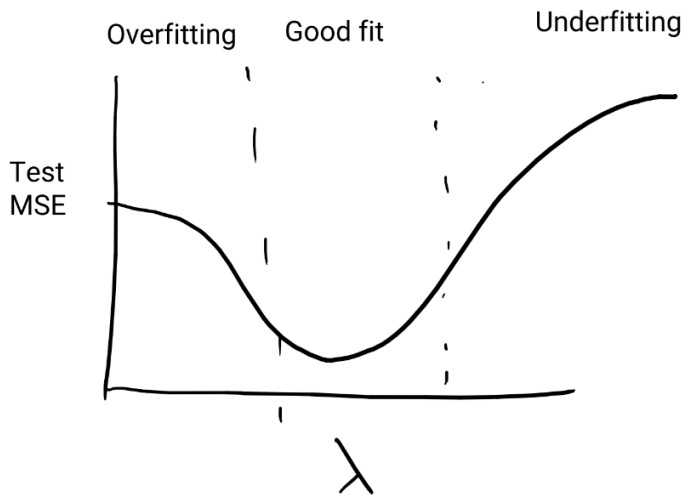
Sketch a plot of how you expect the Test MSE to change as a function of λ. Your sketch should be a smooth curve that shows how the test MSE changes as λ goes from 0 to ∞. Provide justification for your plot. Assume that the right most edge of the graph is where λ is at ∞. Additionally, assume that the linear regression without regularization is overfitting.

Solution:

(a) The purpose of using ridge regression is to avoid overfitting by penalizing the model from having large values for $\boldsymbol{\beta}$.

(b) As λ gets larger, we would expect $\boldsymbol{\beta}$ to get smaller and eventually converge to zero.

(c) The following plot is a sketch of how we would expect the Test MSE to change as a function of λ



When λ is small, it does not affect the model too much and, thus, does not prevent overfitting. When λ is sufficiently large, we expect that overfitting is solved and the model would be a good fit. As λ goes to infinity, ridge regression will heavily bias the model to make $\boldsymbol{\beta}$ be zero which would generally cause underfitting.

6. True of False questions. For each statement, decide whether the statement is True or False and provide justification (full credit for the correct justification).

   (a) In L_2 regularization of linear regression, many coefficients will generally be zero.

   (b) In the leave one out cross validation over the data set of size N, we create and train $N/2$ models.

   (c) 95% confidence interval refers to the interval where 95% of the training data lies.

   (d) If K out of J features have already been selected in Stepwise Variable Selection, then we will train $J - K$ new models to select the next feature to add.

   (e) $P(A|B) = P(B|A)$ if $P(A) = P(B)$ and $P(A)$ is not zero.

   Solution:

   (a) **False**. L_1 regularization, not L_2 regularization, creates sparse model because L_1 norm treats the decrease in magnitude equally regardless of the size of the original value.

   (b) **False**. There would be N models since we create a model by choosing one point as the validation data and the rest as the training data. We can do this $N - 1$ times to create N models.

   (c) **False**. 95% confidence interval refers to the interval estimated from data that 95% of the time will contain the true value of a parameter of interest, e.g. mean of a distribution.

   (d) **True**. Since K features have already been added, there are only $J - K$ features left to add. Thus, if we only add one feature, there are $J - K$ new models to evaluate.

   (e) **True**. This only holds when we know that the events are possible events.