ECE-GY 6143: Introduction to Machine Learning
Midterm, Fall 2020

Prof. Sundeep Rangan

Instructions:

- Answer all **eight** questions.
- Total points are 100.
- All problems have partial credit unless otherwise stated.
- Exam is open book. You may do this exam at home with any resources you wish including the class notes, lectures, homework solutions or searching on the web.
- However, you may not speak / converse with anyone else. Copying will be severely punished.
- For any python code, just write the code. You do not need to run or compile it.

Best of luck!

1. (8 points) *Least squares with a transformation.* Consider the following model for a scalar target $\hat{y}$ from features $\mathbf{x} = (x_1, \ldots, x_d)$,

$$\hat{y} = \sum_{j=1}^{d} \alpha_j \exp(-\beta_j x_j).$$

   (a) (2 points) Is the model linear in the parameters α if β is **fixed**?

   (b) (2 points) Is the model linear in parameters (α, β) **together**?

   (c) (4 points) Write a python function `transform`,

```
def transform(X, beta):
    ...
    return Z
```

   The function should take data matrix `X` and `beta` and return a matrix `Z` such that `yhat = Z.dot(theta)`. For full credit avoid for loops.

   No explanations are need for parts (a) or (b). No partial credit will be awarded for these parts. There will be part credit for part (c).

2. (8 points) *Least squares optimization.* Given a model,

$$\widehat{y}_i = \theta x_{i1} + (1-\theta)x_{i2},$$

find the θ to minimize the least squares loss,

$$J(\theta) = \sum_{i=1}^{N} (y_i - \widehat{y}_i)^2.$$

3. (9 points) *K-fold model validation.* A researcher performs K-fold validation for a regression problem and gets the results below.

Model order d	1	2	3	4	5	6	7	8	9	10
R^2 train mean	0.21	0.49	0.69	0.81	0.85	0.86	0.87	0.88	0.89	0.9
R^2 train SE	0.1	0.1	0.1	0.1	0.1	0.09	0.08	0.07	0.06	0.05
R^2 test mean	0.16	0.44	0.64	0.76	0.8	0.78	0.76	0.74	0.72	0.7
R^2 test SE	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

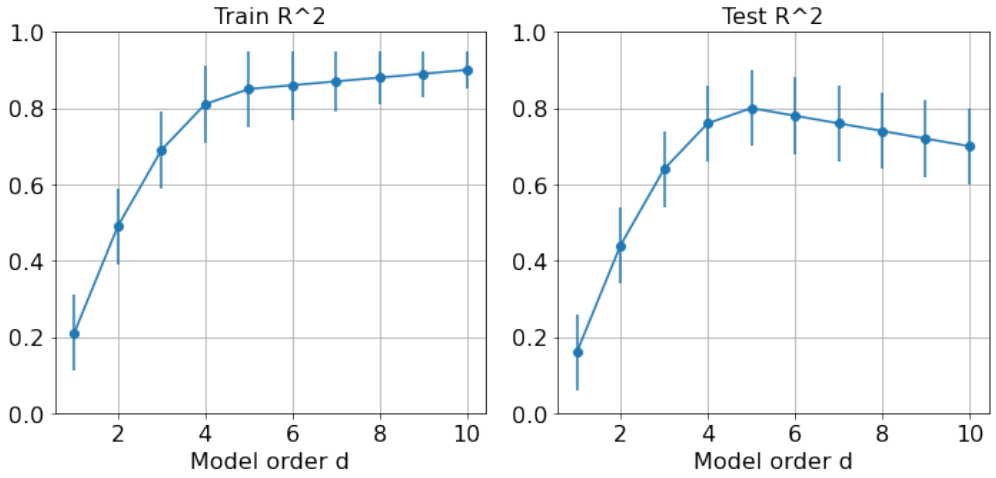


Figure 1: Training and test R^2 values as a function of the model order d

Answer the following. No explanations needed. No partial credit.

(a) (3 points) What is the optimal model order with the normal rule?

(b) (3 points) What is the optimal model order with the one SE rule?

(c) (3 points) At the model order of $d = 8$, the training R^2 is higher than the test R^2 since at $d = 8$ (SELECT ONE):

A. The training model is better than the test model.

B. There is significant over-fitting.

C. There is high bias error.

D. There is significant under-modeling.

4. (15 points) *Model selection.* A researcher wants to perform some binary classification task using data from two cities:

   - `X1,y1`: Data from city 1.
   - `X2,y2`: Data from city 2.

   The outputs `y1` and `y2` are binary labels. For the questions below, assume you have the following methods:

```
# Splits data into training and test
Xtr,Xts,ytr,yts = train_test_split(X,y,test_size)

regr = LogisticRegression()   # Constructs a logistic regression object
regr.fit(Z,y)                 # Fits a model with features Z and outputs y
yhat = regr.predict(Z)        # Predicts outputs given features Z

C = np.vstack((A,B)) # stack two matrices on top of each other
c = np.hstack((a,b)) # join two vectors into one long vector
```

   Write a few lines of python for each of the following tasks:

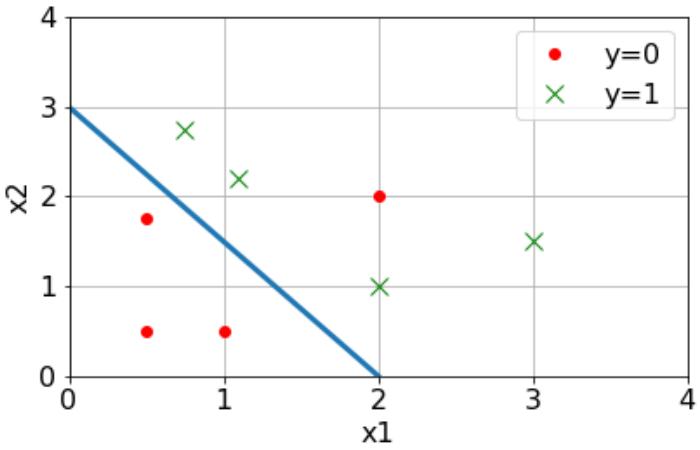
   (a) (5 points) Fit two **separate models** for each city, and measure the test accuracy in each city separately. Use a train-test split with `test_size=0.3` for both cities.

   (b) (6 points) Fit one **combined model** combining the data in both cities and measure the test accuracy on the combined data. Use a train-test split with `test_size=0.3` for both cities.

   (c) (4 points) Select whether using a single model or separate models is better.

   There is no single correct answer. Make any reasonable assumptions.

5. (15 points) *Linear Classification.* A data set has eight data points, $(\mathbf{x}_i, y_i)$ where each data point has two features $\mathbf{x}_i = (x_{i1}, x_{i2})$ a binary label $y_i = \{0, 1\}$. The points are shown below.



(a) (7 points) Write the equations for a binary linear classifier that produces $\widehat{y}$ from $\mathbf{x}$. Find parameters for the classifier such that boundary line of the classifier matches the solid line on the graph. The rule should only make one error.

(b) (8 points) Write the equations for $P(y = 1|\mathbf{x})$ for a logistic classifier. Find parameters for the classifier such that:

- The set of points $\mathbf{x}$ with $P(y = 1|\mathbf{x}) = 0.5$ matches the solid line in the figure
- $P(y = 1|\mathbf{x}) = 0.8$ at $\mathbf{x} = (2, 2)$.

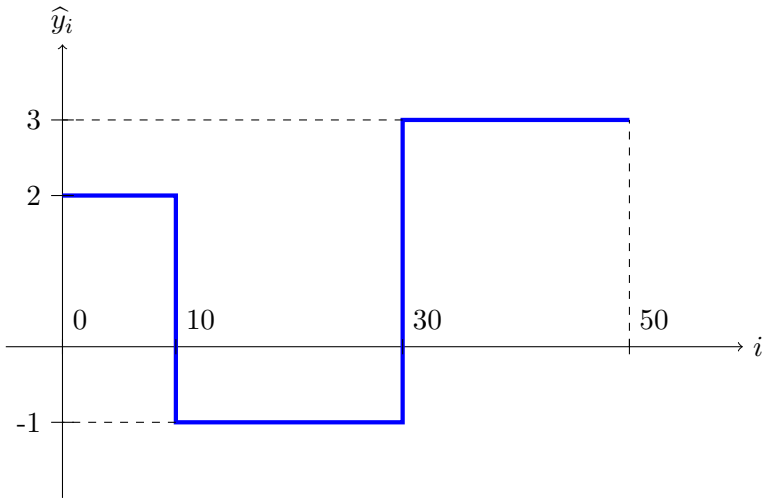
Your answer will have logarithms in them. You do not need to evaluate the logarithms or simplify any expressions. Just provide enough steps that the parameters can be computed.

6. (15 points) *LASSO.* Consider a model $\widehat{\mathbf{y}} = \mathbf{A}\mathbf{w}$ where $\mathbf{A}$ is the $T \times T$ matrix,

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix}.$$

and $\mathbf{w}$ are some unknown coefficients.

(a) (9 points) Find $\mathbf{w}$ for $\widehat{\mathbf{y}} = (\widehat{y}_0, \widehat{y}_1, \ldots, \widehat{y}_{T-1})$ with $T = 50$ shown in the figure below. At the two discontinuities assume $\widehat{y}_{10} = 2$ and $\widehat{y}_{30} = -1$.



(b) (6 points) For a given $\mathbf{y}$, suggest a regularized cost function to find $\mathbf{w}$ such that $\widehat{\mathbf{y}} = \mathbf{A}\mathbf{w}$ is close to $\mathbf{y}$ but $\widehat{y}_i$ is mostly constant and changes values for only a few i.

7. (15 points)*Computing gradients.* Consider the model,

$$\widehat{y}_i = \log\left[1 + \exp\left(\sum_{j=1}^{d} X_{ij}\beta_j\right)\right].$$

The model is trained to minimize the RSS

$$J = \sum_{i=1}^{n}(y_i - \widehat{y}_i)^2.$$

(a) (7 points) Compute the gradient components, $\partial J/\partial \beta_j$.

(b) (8 points) Write a python function to that returns function J and the gradients,

```
def Jeval(...):
    ...
    return J, Jgrad
```

You must determine the arguments for the function. For full credit, avoid for loops.

8. (15 points) *Optimization line search.*

   (a) (7 points) Suppose for $\mathbf{w}_0 = (1, 2, 3)$ we have $f(\mathbf{w}_0) = 4$ and $\nabla f(\mathbf{w}_0) = (5, 6, 7)$. Using a first order approximation, what is $f(\mathbf{w}_1)$ where $\mathbf{w}_1$ is the gradient step,

   $$\mathbf{w}_1 = \mathbf{w}_0 - \alpha \nabla f(\mathbf{w}_0), \quad \alpha = (10)^{-4}.$$

   Your answers will have some additions and subtractions. You do not need to evaluate them or simplify your expressions.

   (b) (8 points) Now suppose you are given a python function `feval` that returns $f(\mathbf{w})$ and its gradient $\nabla f(\mathbf{w})$:

   ```
   f, fgrad = feval(w)
   ```

   For a given $\mathbf{w}_0$, we want to find the α that minimizes the following:

   $$\widehat{\alpha} = \arg\min_{\alpha} f(\mathbf{w}_0 - \alpha \nabla f(\mathbf{w}_0)),$$

   Complete the following function to perform the optimization:

   ```
   def linesearch(feval, w0, alpha_min, alpha_max, nalpha):
       ...
       return alpha_opt
   ```

   The function should try `nalpha` points linearly spaced from `alpha_min` to `alpha_max`.