ECE M146 Midterm

[redacted]

TOTAL POINTS

116.5 / 120

QUESTION 1

Short Questions 28 pts

1.1 4 / 4

✓ **- 0 pts** **False and correct reasoning**

- 1 pts False but partially correct reasoning

- 2 pts False but incorrect reasoning or no reason

- 3 pts True but attempted to reason (invalid reason)

- 4 pts True and no reason or not attempted

1.2 4 / 4

✓ **- 0 pts** **Correct (True/false) with correct explanation**

- 1 pts Partially correct expalanation

- 2 pts Attempted but Incorrect explanation

- 3 pts Attempted but no explanation

- 4 pts No attempt

1.3 4 / 4

✓ **- 0 pts** **False with correct explanation**

- 1 pts False with partially correct explanation

- 2 pts False with incorrect explanation

- 3 pts False with no explanation

- 3 pts True with attempted explanation

- 4 pts True with no explanation

- 4 pts Not attempted

1.4 4 / 4

✓ **- 0 pts** **False with correct explanation**

- 1 pts False with partially correct explanation

- 1.5 pts False with incorrect explanation

- 2 pts False with no explanation

- 3 pts True with attempted explanation

- 4 pts True with no explanation

- 4 pts Not attempted

1.5 4 / 4

✓ **- 0 pts** **True statement with correct explanation**

- 1 pts True statement with partially correct explanation (mixing objective function value and training error, simply repeating the question, etc.)

- 1.5 pts True statement with incorrect explanation

- 2 pts True statement with no explanation

- 3.5 pts False statement with attempted explanation

- 4 pts False statement with no explanation / not attempted

1.6 0.5 / 4

- 0 pts False statement with correct explanation

- 1 pts False statement with partially correct explanation

- 1.5 pts False statement with incorrect explanation

- 2 pts False statement with no explanation

✓ **- 3.5 pts** **True statement with attempted explanation**

- 4 pts True statement with no explanation / not attempted.

1.7 4 / 4

✓ **- 0 pts** **True statement with correct explanation**

- 1 pts True statement with partially correct explanation

- 1.5 pts True statement with incorrect explanation

- 2 pts True statement with no explanation

- 3.5 pts False statement with attempted explanation

- 4 pts False statement with no explanation / not attempted

QUESTION 2

Multiple Choice Questions 18 pts

2.1 6 / 6

- 1 pts a wrong

- 1 pts b wrong

- 1 pts c wrong

- 1 pts d wrong

- **1 pts** e wrong
- **1 pts** f wrong

✓ **- 0 pts Correct**

2.2 6 / 6

- **1 pts** a wrong
- **1 pts** b wrong
- **1 pts** c wrong
- **1 pts** d wrong
- **1 pts** e wrong
- **1 pts** f wrong

✓ **- 0 pts correct**

2.3 6 / 6

- **1 pts** a wrong
- **1 pts** b wrong
- **1 pts** c wrong
- **1 pts** d wrong
- **1 pts** e wrong
- **1 pts** f wrong

✓ **- 0 pts correct**

QUESTION 3

Decision Tree 28 pts

3.1 8 / 8

✓ **- 0 pts all correct**

- **1.5 pts** Incorrect or no gain(Y|V)
- **1.5 pts** Incorrect or no gain (Y|W)
- **1.5 pts** Incorrect or no gain (Y|X)
- **2 pts** No attempt to find gain (Y|V)
- **2 pts** No attempt to find gain (Y|W)
- **2 pts** No attempt to find gain (Y|X)
- **2 pts** Incorrect attribute picked
- **0.5 pts** Correct attribute picked corresponding to the incorrectly computed gains
- **1 pts** Incorrect entropy of Y
- **1.5 pts** Gains not simplified

3.2 8 / 8

✓ **- 0 pts Correct**

- **2 pts** If root is not X
- **3 pts** Incorrect overall structure of the tree
- **2 pts** For more than one minor mistakes in the leaves and branch labels
- **5 pts** If no tree drawn but only correct explanation provided
- **6 pts** If no tree drawn and partially correct explanation provided
- **8 pts** No tree drawn and incorrect or no explanation provided
- **1 pts** Single mistake in either a leaf or a branch

3.3 4 / 4

✓ **- 0 pts Correct**

- **0.5 pts** Label for (1,0,0) != 1
- **0.5 pts** Label for (001) != 0
- **0.5 pts** Label for (010) != 1
- **1 pts** Correct answer for Yes/no part with incorrect reasoning
- **0.5 pts** Correct answer for Yes/no part with no reasoning
- **1 pts** Incorrect answer for yes/no part but attempt at reasoning
- **2 pts** Incorrect answer for yes/no part and no reasoning
- **2.5 pts** No attempt for yes/no part

3.4 8 / 8

✓ **- 0 pts Correct**

- **1 pts** Slightly Incorrect tree
- **2.5 pts** Completely incorrect tree
- **3.5 pts** No tree
- **1 pts** Mildly incorrect conclusion
- **2 pts** Incorrect conclusion
- **3 pts** No attempt at conclusion
- **8 pts** Unattempted question
- **2 pts** Conclusion unclear

QUESTION 4

Perceptron and Logistic Regression 30 pts

4.1 4 / 4

- **2 pts** 1) incorrect
- **2 pts** 2) incorrect

✓ **- 0 pts correct**

4.2 8 / 8

✓ **+ 3 pts First sample correct**

✓ **+ 3 pts Second sample correct**

✓ **+ 2 pts Third sample correct**

+ 0.5 pts incorrect but attempted answer

+ 5 pts no bias term update / data augmentation incorrect

+ 0 pts no answer

4.3 4 / 4

- 2 pts First question incorrect

- 2 pts Second question incorrect

✓ **- 0 pts Correct**

4.4 14 / 14

✓ **- 0 pts correct**

- 3 pts Training accuracy curve wrong

- 3 pts Testing accuracy curve wrong

- 2 pts Overfit range wrong

- 2 pts Underfit range wrong

- 4 pts explanation wrong (you may need to see adjust points)

- 14 pts No answer

- 0.5 pts Draw error curve instead of accuracy

QUESTION 5

Linear Regression 16 pts

5.1 6 / 6

✓ **- 0 pts Correct**

- 6 pts No answer

- 6 pts Some attempt but completely incorrect

- 2 pts Correct answer with no explanation.

- 3 pts Issues with answer: e.g.,Confusing linear regression with logistic regression, what is \sigma? Or scalar vector confusion etc.

- 2 pts Minor error.

- 5 pts Major issues.

- 3 pts Confusing the derivatives and incompatability of matrices.

- 1 pts missing factor or small error (dimension issues etc).

5.2 6 / 6

✓ **- 0 pts Correct**

- 6 pts No answer

- 5 pts Major errors

- 1 pts Minor issue, missing/additional factor

- 3 pts Closed form solution not provided but mentioned gradient =0 and tried to simplify

- 3 pts Multiple issues with answer such as treating matrices as scalars and dividing two matrices or using incorrect gradient

- 3 pts Closed form solution not provided however mentioned that the optimal can be found using gradient descent

5.3 4 / 4

✓ **- 0 pts Correct**

- 4 pts No answer

- 2 pts unclear answer, minor issues

- 1 pts accurate but not complete

- 3 pts unclear answer, major issues

Machine Learning M146
Prof. Suhas Diggavi

UCLA Spring quarter 2017-2018
Handout # 16, Wednesday, May 9th 2018

MIDTERM

Wednesday, 9th May 2018, 10am-11:50am
This exam has 5 problems and 120 points in total.

Instructions

- You are allowed to use 1 sheet of paper for reference. No mobile phones or calculators are allowed in the exam.
- You can attempt the problems in any order as long as it is clear which problem is being attempted and which solution to the problem you want us to grade.
- If you are stuck in any part of a problem do not dwell on it, try to move on and attempt it later.
- Please solve every problem in **fixed space right after the question**. It is your responsibility to notify the grader if you use any additional space other than the space reserved.
- You may find the following useful.
  - Consider the vectors $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{a} \in \mathbb{R}^n$ and the symmetric matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$.

$$\nabla_{\boldsymbol{x}} \boldsymbol{a}^T \boldsymbol{x} = \boldsymbol{a} \qquad \nabla_{\boldsymbol{x}} \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} = 2\boldsymbol{A}\boldsymbol{x}$$

  - $\log_2$ look up table

$\log_2(3)$	$\log_2(5)$	$\log_2(6)$	$\log_2(7)$	$\log_2(9)$
1.58	2.32	2.58	2.81	3.17

GOOD LUCK!

NAME [redacted]

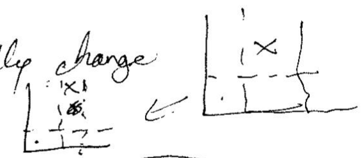
UID: [redacted]

Problem 1 (SHORT QUESTIONS, TRUE/FALSE (28 pts))

Choose either True or False for each of the following statements. For each response please give a very brief explanation of why you believe it is true/false. **Answers with no justification will not get credit.**

(a) Feature scaling is an important step of data preprocessing before training a decision tree using ID3 algorithm. **TRUE** / **FALSE** [4pts]

it's true for KNN, but for this it won't meaningfully change decision boundary

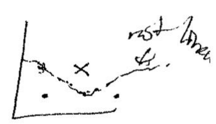


(b) We cannot apply nearest neighbor classification on categorical data. **TRUE** / **FALSE** [4pts]

one-hot encoding is counterexample

(c) Nearest neighbor classification always produces a linear decision boundary. **TRUE** / **FALSE** [4pts]

~~there~~ boundary based on lp-norm usually; very rarely does this yield a linear decision boundary



(d) Gradient descent for training a logistic classification model will not converge if the data is not linearly separable. **TRUE** / **FALSE** [4pts]

loss function is convex, so there is a global optimum & given an appropriate learning rate, it will converge.

(e) In a least-squares linear regression problem, adding an ℓ_2 regularization penalty cannot decrease the training error. **TRUE** / **FALSE**. [4pts]

LS w/out regularization already minimizes training error; the issue is that it may not generalize well to test data, so one might wish to regularize to constrain weights

(f) On the same training set, the gradient descent and stochastic gradient descent will converge to the same solution always. **TRUE** / **FALSE**. [4pts]

By LLN and assuming a convex loss function, both will eventually converge to the global optimum.

(g) The function $k(\mathbf{x}_n, \mathbf{x}_m) = 100(2 + \mathbf{x}_n^T\mathbf{x}_m) + 0.2e^{-\|x_n - x_m\|_2^2/2\sigma^2}$ is a valid kernel function. **TRUE** / **FALSE**. [4pts]

It's a sum of valid kernel functions so it should also be a valid kernel function.

Problem 2 (Multiple choice questions (18 pts))

Make sure to choose all choices that you think fit the question description. There could be more than one correct choice.

(a) If data is not linearly separable, choose all the following methods which can possibly reach a training (classification) error 0. *[6pts]*

a. Decision tree
b. KNN
c. ~~Perceptron~~
d. ~~Averaged perceptron~~
e. ~~Logistic regr~~ession
f. None of above

(b) Select all the following choices that can be used to reduce overfitting. *[6pts]*

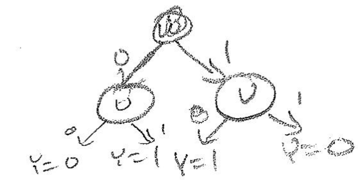
a. Increase the value of K in K-nearest neighbor.
b. Prune the decision tree by setting the MaxDepth.
c. Use stochastic gradient descent instead of (batch) gradient descent to compute the optimal solution in logistic regression.
d. Increase training set size.
e. Use kernel methods to map the original feature into higher dimensional feature space.
f. None of above.

(c) Select all the following choices that could potentially help with the following issue: Suppose the error-versus-sample size curves (learning curves) converge to similar (normalized) test and training error with sample size, but both of them are high. ← high bias situation *[6pts]*

a. Incr~~ease training set size~~
b. Use kernel methods to map the original feature into higher dimensional feature space.
c. ~~Simplify the hypothesis s~~pace
d. ~~Reduce the feature set~~.
e. ~~Add ℓ_2 regularization~~ term to the objective function of linear regression.
f. None of above

Problem 3 (DECISION TREE (28 pts))

You get the following data set:



#	Attribute V	Attribute W	Attribute X	Label Y
1	0	0	0	0
2	0	1	0	1
3	1	0	0	1
4	1	1	0	0
5	1	1	1	0

Each sample has attribute V, W and X. Your task is to build a decision tree for classifying label Y. For any log you used in this question, please use $\log_2$. You may find the $\log_2$ look up table on page 1 useful.

(a) Compute the information gains $Gain(Y|V)$, $Gain(Y|W)$ and $Gain(Y|X)$. Which attribute would ID3 select first? [8pts]
Hint: It is OK to leave the answer without computing the ultimate value. You may simplify expressions in terms of $\log_2(\cdot)$, and substitute the value of $\log_2(\cdot)$ from the look-up table only when it's necessary.

$$H(Y) = \frac{2}{5}\log_2\frac{5}{2} + \frac{3}{5}\log_2\frac{5}{3}$$

$$H(Y|V) = \frac{2}{5}[H(Y|V=0)] + \frac{3}{5}[H(Y|V=1)]$$

$$H(Y|V) = \frac{2}{5}(1) + \frac{3}{5}\left[\frac{1}{3}\log_2 3 + \frac{2}{3}\log_2\frac{3}{2}\right] = \frac{2}{5} + \frac{3}{5}\left(\frac{1}{3}\cdot 1.58 + \frac{2}{3}\cdot 1.58 -\right.$$

$$Gain(Y,V) = -\left(\frac{2}{5} + \frac{3}{5}\left(1.58 - \frac{2}{3}\right)\right) + H(Y)$$

$$H(Y|W) = \frac{2}{5}[H(Y|W=0)] + \frac{3}{5}[H(Y|W=1)]$$

$$H(Y|W) = \frac{2}{5}(1) + \frac{3}{5}\left[\frac{1}{3}\log_2 3 + \frac{2}{3}\log_2\frac{3}{2}\right]$$

$$H(Y|W) = \frac{2}{5} + \frac{3}{5}\left(1.58 - \frac{2}{3}\right)$$

$$Gain(Y,W) = H(Y) - \left(\frac{2}{5} + \frac{3}{5}\left(1.58 - \frac{2}{3}\right)\right)$$

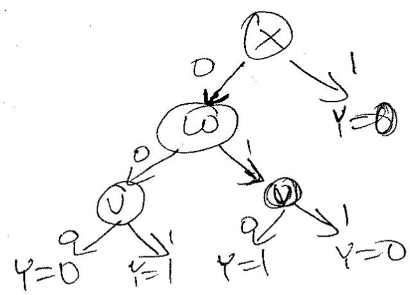
$$H(Y|X) = \frac{4}{5}[H(Y|X=0)] + \frac{1}{5}[H(Y|X=1)]$$

$$H(Y|X) = \frac{4}{5}(1) + \frac{1}{5}(0)$$

$$Gain(Y|X) = H(Y) - \frac{4}{5}$$

↑ largest information gain. ID3 selects X

(b) Write down the entire decision tree constructed by ID3. [8pts]



W & U could be interchanged, for what it's worth

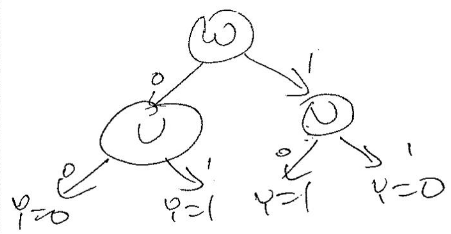
(c) Find the labels using the constructed tree for the following test set. Does the constructed tree give zero test error? [4pts]

$$(V, W, X) = (1, 0, 0), \text{ with label } Y = 1 \rightarrow \hat{Y} = 1$$
$$(V, W, X) = (0, 0, 1), \text{ with label } Y = 0 \rightarrow \hat{Y} = 0$$
$$(V, W, X) = (0, 1, 0), \text{ with label } Y = 0 \rightarrow \hat{Y} = 1$$

test error $= \frac{1}{3}$. $(V, W, X) = (0, 1, 0)$ is misclassified.

(d) Can you find a tree with smaller height than the tree returned by ID3 in b), which also have zero training error? What conclusion does that imply about the performance of the ID3 algorithm? [8pts]

Hint: Try to design the tree with the first splitting attribute as either V or W.

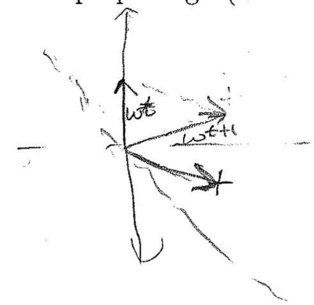


← height of this tree is 2, as opposed to tree in (b), which has height of 3.

This indicates that ID3 does not yield optimal solutions (in terms of minimizing tree depth & hence model complexity). ID3, a greedy algorithm, at best approximates the optimal tree based on the information gain heuristic,

Problem 4 (Perceptron and logistic regression (30 pts))

(a) Write down the perceptron learning rule by filling in the blank below with a proper sign (+ or −). Note that η is a small constant learning rate factor. [4pts]



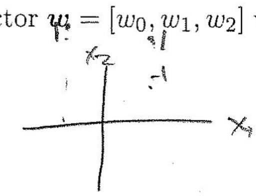
1. Input $\boldsymbol{x}$ is falsely classified as negative:

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t \underline{\quad + \quad} \eta \boldsymbol{x}$$

2. Input $\boldsymbol{x}$ is falsely classified as positive:

$$\boldsymbol{w}^{t+1} = \boldsymbol{w}^t \underline{\quad - \quad} \eta \boldsymbol{x}$$

(b) Consider a perceptron algorithm to learn a 3-dimensional weight vector $\boldsymbol{w} = [w_0, w_1, w_2]$ with w_0 the bias term. Suppose we have training set as following:



#	1	2	3
x	[1,1]	[1,2]	[-1,3]
y	-1	1	1

1. Show the weights at each step of the perceptron learning algorithm. Loop through the training set **once** (i.e. MaxIter = 1) with the same order presented in the above table. Start the algorithm with initial weight $\boldsymbol{w} = [w_0, w_1, w_2] = [1, 0, 0]$. And we assume the learning rate $\eta = 1$. [8pts]

$\tilde{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$

$w_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

$\hat{y}_1 = \text{sign}(w_0^T \tilde{x}_1) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = 1$

$\hat{y}_1 \neq y_1$, so update

$w_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix}$

$\hat{y}_2 = \text{sign}(w_1^T \tilde{x}_2) = \begin{bmatrix} 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = -1$

$\hat{y}_2 \neq y_2$, so update

$w_2 = \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$

$\hat{y}_3 = \text{sign}(w_2^T \tilde{x}_3) = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 3 \end{bmatrix} = 1$

$\hat{y}_3 = y_3$, so no update necessary.

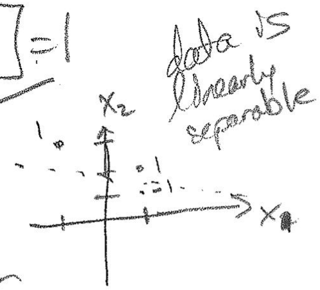
at the end of one complete loop through data,

$w = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$

2. Does the weight vector you learned after one iteration (i.e. the final weight vector you find in 1.) separate the dataset perfectly? If yes, briefly explain. If no, suppose now MaxIter = ∞, can the perceptron algorithm described in 1. finally find a weight vector that perfectly separates the training set? [4pts]

No, misclassifies point 1: $\text{sign}(\tilde{w}^T \tilde{x}_1) = [1\ 0\ 1]\begin{bmatrix}1\\1\\1\end{bmatrix} = 1$

$\hat{y} \neq y_1$, so not classified correctly



Eventually, it should find a linear separator because as shown in my diagram, the data is linearly separable.

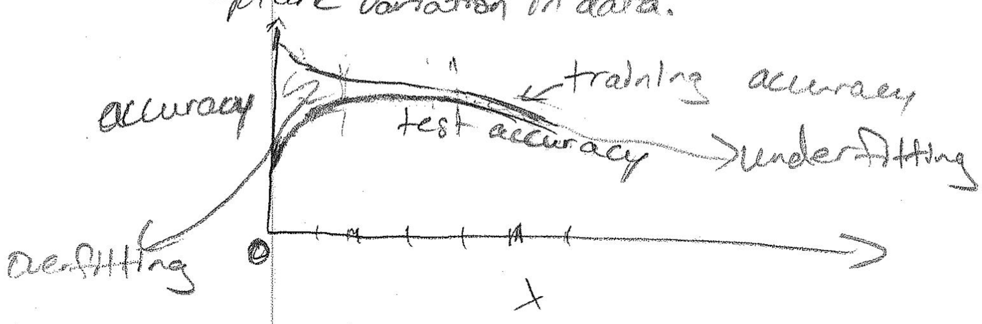
(c) Consider regularized training objective function for logistic regression:

$$\mathcal{L}(\boldsymbol{w}) = -\left[\sum_i \left[y_i \log \sigma(\boldsymbol{w}^T \boldsymbol{x}_i) + (1 - y_i) \log(1 - \sigma(\boldsymbol{w}^T \boldsymbol{x}_i))\right]\right] + \frac{1}{2}\lambda \boldsymbol{w}^T \boldsymbol{w}$$

1. Draw a graph with two curves that shows how the training accuracy and test accuracy are expected to vary with λ. [6pts]
2. On the same graph, point out the ranges of λ for which your model is more likely to underfit or overfit **respectively**. [4pts]
3. Briefly explain. [4pts]

Note: 1) A qualitative sketch is enough. 2) You will NOT get any points without clearly indicating which curve corresponds to which accuracy.

Overfits if λ is small, underfits is λ is large.
Small λ encourages large weights, and these weights are likely to generalize poorly.
Large λ encourages small weights and will not adequately capture variation in data.



Problem 5 (Linear Regression (16 pts))

In class you have seen linear regression where the objective is as follows

$$J(\boldsymbol{\theta}) = ||\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}||_2^2$$

where the rows of $\boldsymbol{X}$ are the data points. We had seen that the closed form for the global minimizer of $J(\boldsymbol{\theta})$ is

$$\boldsymbol{\theta}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Now consider the regularized objective function $\tilde{J}(\boldsymbol{\theta})$:

$$\tilde{J}(\boldsymbol{\theta}) = ||\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}||_2^2 + ||\boldsymbol{\theta}||_{\boldsymbol{\Lambda}}^2,$$

$$\begin{pmatrix} \theta_{1,1} & \theta_{2,1} \cdots \theta_{n,1} \\ \theta_{1,2} & \ddots \\ \theta_{1,n} & \cdots \theta_{n,n} \end{pmatrix} \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \lambda_2 & \vdots \\ 0 & \cdots & \lambda_n \end{pmatrix} \begin{pmatrix} \theta_{1,1} \cdots \theta_{1,n} \\ \vdots \ddots \\ \theta_{n,1} \cdots \theta_{n,n} \end{pmatrix}$$

where $||\boldsymbol{\theta}||_{\boldsymbol{\Lambda}}^2 = \boldsymbol{\theta}^T\boldsymbol{\Lambda}\boldsymbol{\theta}$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_{D+1})$ is a diagonal matrix with diagonal elements $\lambda_i > 0, i = 1, \ldots, D+1$.

(a) Find the gradient of $\tilde{J}(\boldsymbol{\theta})$. *[6pts]*

$$\tilde{J}(\underline{\theta}) = (X\theta - y)^T (X\theta - y) + \theta^T \Lambda \theta$$

$$\tilde{J}(\underline{\theta}) = (\theta^T X^T - y^T)(X\theta - y) + \theta^T \Lambda \theta$$

$$\tilde{J}(\underline{\theta}) = \theta^T X^T X\theta - y^T X\theta - y^T y - \theta^T X^T y + \theta^T \Lambda \theta$$

scalar; y doesn't depend on θ

equivalent

$$\nabla_\theta \tilde{J}(\underline{\theta}) = 2\underline{X}^T \underline{X}\underline{\theta} - 2\underline{X}^T \underline{y} + 2\underline{\Lambda}\underline{\theta}$$

(b) Assuming that $\widetilde{J}(\boldsymbol{\theta})$ is convex, find the optimal solution $\boldsymbol{\theta}^*$ to minimize $\widetilde{J}(\boldsymbol{\theta})$. [*6pts*]

$$\nabla_\theta \tilde{J}(\theta) = 2X^TX\theta - 2X^Ty + 2\Lambda\theta = 0$$

$$(X^TX + \Lambda)\theta = X^Ty$$

$$\theta^* = (X^TX + \Lambda)^{-1}X^Ty$$

(c) Recall that the ℓ_2-regularized objective done in class has the following form: [*4pts*]

$$\hat{J}(\boldsymbol{\theta}) = ||\boldsymbol{X}\boldsymbol{\theta} - \boldsymbol{y}||_2^2 + \lambda||\boldsymbol{\theta}||_2^2,$$

How does the objective in this question relate to the ℓ_2-regularized objective?

The set of objectives specified by the objective in this question contains the l_2-regularized objective.

for l_2-reg: $\Lambda = \begin{bmatrix} \lambda & & 0 \\ & \lambda & \\ 0 & & \lambda \end{bmatrix} \rightarrow \lambda_1 = \lambda_2 = \lambda_3 = \lambda$

this question: $\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_{D+1} \end{bmatrix}$