HW0 - CS 189

1.

Who else did you work with on this homework? In case of course events, just describe the group. How did you work on this homework? Any comments about the homework?

Mostly independently. However, I met up once and discussed HW0 with my study group for this class: Ehimare Okoyomon, Prashanth Ganesh, Daniel Mockaitis

I certify that all solutions are entirely in my words and that I have not looked at another student's solutions. I have credited all external sources in this write up}

Nicholas Lorio, 26089160

2. Linear Algebra Review Questions

CS189 - HW0 work paper

Nick Lario
26089160
1/18/18

3. lin Alg review

$u = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ $v = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ $M = uv^T$

a. $M = \begin{bmatrix} 1 \\ 2 \end{bmatrix} [2 \ 3] = \begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix}$

$\det(A - \lambda I) = 0 \Rightarrow \left| \begin{bmatrix} 2-\lambda & 3 \\ 4 & 6-\lambda \end{bmatrix} \right| = 0$ $(2-\lambda)(6-\lambda) - 12 = 0$

$12 - 8\lambda + \lambda^2 - 12 = 0$

$Ax = \lambda x$

$(A - \lambda I)x = 0$

$\lambda^2 - 8\lambda = 0$ $\left.\begin{matrix} \lambda = 0 \\ \lambda = 8 \end{matrix}\right]$ eigen values

$\lambda = 0$ $\begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \emptyset$ $\left[\begin{array}{cc|c} 2 & 3 & 0 \\ 4 & 6 & 0 \end{array}\right] \xrightarrow[-2R_1 + R_2 \to R_2]{} \left[\begin{array}{cc|c} 2 & 3 & 0 \\ 0 & 0 & 0 \end{array}\right]$

$x = \begin{bmatrix} -3/2 \\ 1 \end{bmatrix}$ eigenvector for $\lambda = 0$

$2x_1 + 3x_2 = 0$ $x_1 = -3/2 x_2$

$x_1 = -3/2$

$x_2 = 1$

Confirm $\begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix} \begin{bmatrix} -3/2 \\ 1 \end{bmatrix} = \begin{bmatrix} -3+3 \\ -6+6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ ✓

$\lambda = 8$ $\left[\begin{array}{cc|c} -6 & 3 & 0 \\ 4 & -2 & 0 \end{array}\right] \xrightarrow[R1\cdot(2/3) + R2 \to R_2]{} \left[\begin{array}{cc|c} -6 & 3 & 0 \\ 0 & 0 & 0 \end{array}\right]$ $-6x_1 + 3x_2 = 0$

$x_1 = 1/2 x_2$

$x_2 = 2$ $x_1 = 1$

$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ eigenvector for $\lambda = 8$

Confirm $\begin{bmatrix} -6 & 3 \\ 4 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -6+6 \\ 4-4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ ✓

b. $\mathrm{Det}(M) = \prod \lambda_i(M) = 0 \cdot 8 = 0 = 2\cdot 6 - 3 \cdot 4 = 0$

$\mathrm{rank}(M) = 2 - \mathrm{nullity}(M) = 1$ by rank nullity theorem

$\lambda = 0$
mult: 1
$\therefore$ nullity(M) = 1

c. $N = pq^T$ w/ $p \in \mathbb{R}^d$ & $q \in \mathbb{R}^d$ is a Square $d \times d$ Matrix.

A Matrix that is the product of two vectors has a rank of 1. Thus $\text{rank}(N) = 1$. This due to the fact that

$$\left\{ \begin{array}{l} N = pq^T \Rightarrow \text{rank}(N) = 1 \\ \\ \text{every column of } N \text{ is a multiple of } p. \end{array} \right.$$

N is not full rank, as it's rows are not linearly independent, therefore $\lambda = 0$ is an eigenvalue of the Matrix.

$\underline{\lambda = 0}$ has Multiplicity due to the rank nullity theorem

$\text{rank}(N) + \text{nullity}(N) = d \qquad \text{nullity}(N) = d - \text{rank}(N)$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \underbrace{d - 1}$

$\lambda = 0$ w/ $\underline{\text{Multiplicity } \boxed{\text{at least}}\ d-1}$ bring N to nullspace $\{v : Nv = 0\}$ $\lambda = 0$

the remaining eigenvalue has Multiplicity of 1.

$\underline{Np = \lambda p}$ general eigenspace, $Np = (pq^T)p = p(q^Tp) = (q^Tp)p = \lambda p$

"by rules/propertys of vector Multiplication"

$\therefore \underbrace{\lambda = q^Tp}$ w/ Multiplicity of 1, $\neq 0$

together that gives @ least d eigenvalues (Counting Multiplicity)

- $\det(N) = \prod_{i=1}^{d} \lambda_i(N) = 0$ as $\underset{\text{Mult: } d-1}{\lambda = 0}$, $\underset{\text{Mult: } 1}{\lambda = q^T p}$

- eigenvectors for respective eigenvalues

$\lambda = 0$: $\{v : Nv = 0\}$ vectors in the null space of N

$\lambda = q^T p$

$Np = \lambda p$

p is the eigenvector for $\lambda = q^T p$

$$\begin{bmatrix} p_1 q_1 & \cdots & p_1 q_d \\ & \ddots & \\ p_d q_1 & & p_d q_d \end{bmatrix} \begin{bmatrix} -q_i/q_1 \\ \vdots \\ 0 \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

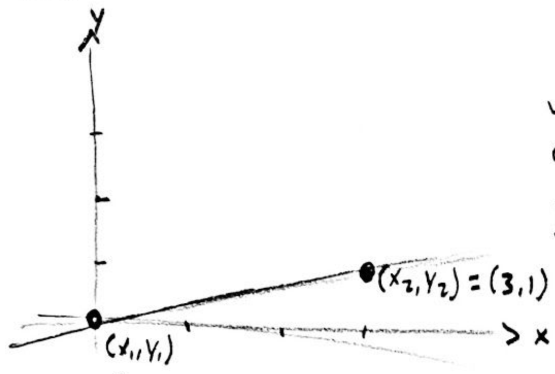
$V_i = \begin{bmatrix} -q_i/q_1 \\ 1 @ i^{th} \text{ entry in } d\times 1 \text{ vector column} \\ \end{bmatrix}$ $i = 2, \dots, d$, $V_i \in \mathbb{R}^{d\times 1}$

$$V_2 = \begin{bmatrix} -q_2/q_1 \\ 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \quad V_3 = \begin{bmatrix} -q_3/q_1 \\ 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix} \quad \cdots \quad V_d = \begin{bmatrix} -q_d/q_1 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$V_2 \dots V_d$ are eigenvectors corresponding to the $\lambda = 0$ eigenvalues of multiplicity $d-1$.

4. Linear Regression \& Adversarial Noise

4.a.



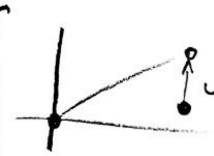
let us consider the case in which $n=2$. The adversary's capacity to alter only one of the data points through the introduction of noise means the adversary cannot make the model OLS Best Fit line have the adversary's desired w_1^* (slope) & w_2^* (y-intercept)

The following counterexample shows the above statement to be true & that the statement "adversary can always alter one point to make $\hat{w}_1 = w_1^*$ &/or $\hat{w}_2 = w_2^*$" to be false.

Given $(x_1, y_1) = (0,0)$ & $(x_2, y_2) = (3,1)$ => $w_1 = 3$ $w_2 = 0$

let us say our adversary wants $w_1^* = 1$ & $w_2^* = 3$

$\tilde{y}_i = y_i + \varepsilon_i$ we can adjust one pt.



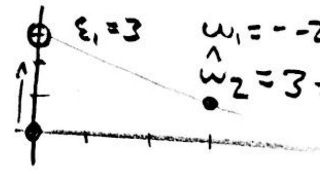
$\hat{w}_1 = 1 = w_1^*$

up two, $\varepsilon_2 = 2$

But $w_2 \neq w_2^* \neq 3$

$w_2 = 0$

or



$\varepsilon_1 = 3$

$w_1 = -2/3 \neq w_1^*$

$\hat{w}_2 = 3 = w_2^*$

Both cannot be satisfied by only adjusting one. ∴ it is not always true

4.b. 1. $(x_i, y_i)\ i = 1, \dots, n$

yes
From $n=2$ case we have reason to believe it's true for adversary to manipulate.

2. Optimization

3. $\min_{\vec{w}} \sum_{i=1}^{n} \left(w_1 x_i + w_2 - y_i - \vec{\varepsilon} \right)^2$. We know $\vec{w} = (A^T A)^{-1} A^T (\vec{y} - \vec{\varepsilon})$ as it follows OLS format.

For the model desired by adversary, the following two perturbations are introduced to our normal OLS model.

let $A = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}$, $A^T = \begin{bmatrix} x_1 & \dots & x_n \\ 1 & \dots & 1 \end{bmatrix}$ $\vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$ $\vec{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

$$\vec{w} = \begin{bmatrix} w_1^* \\ w_2^* \end{bmatrix} = \left(\underbrace{\begin{bmatrix} x_1 & \dots & x_n \\ 1 & \dots & 1 \end{bmatrix}}_{2 \times n} \underbrace{\begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}}_{n \times 2} \right)^{-1} \underbrace{\begin{bmatrix} x_1 & \dots & x_n \\ 1 & \dots & 1 \end{bmatrix}}_{2 \times n} \underbrace{\begin{bmatrix} y_1 - \varepsilon_1 \\ y_2 - \varepsilon_2 \\ y_3 \\ y_4 \\ \vdots \\ y_n \end{bmatrix}}_{n \times 2}$$

• Solve for ε_1 & ε_2 in terms of $(x_i, y_i)\ i = 1, \dots, n$

$$A^T A \begin{bmatrix} w_1^* \\ w_2^* \end{bmatrix} = \begin{bmatrix} x_1(y_1 - \varepsilon_1) + x_2(y_2 - \varepsilon_2) + x_3 y_3 + \dots + x_n y_n \\ y_1 - \varepsilon_1 + y_2 - \varepsilon_2 + y_3 + \dots + y_n \end{bmatrix}$$

$$\begin{bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{bmatrix} \begin{bmatrix} w_1^* \\ w_2^* \end{bmatrix} = \begin{bmatrix} x_1(y_1 - \varepsilon_1) + x_2(y_2 - \varepsilon_2) + x_3 y_3 + \dots + x_n y_n \\ \left(\sum_{i=1}^{n} y_i \right) - \varepsilon_1 - \varepsilon_2 \end{bmatrix}$$

$$\begin{bmatrix} w_1^* \sum x_i^2 + w_2^* \sum x_i \\ w_1^* \sum x_i + w_2^* n \end{bmatrix} = \quad ''$$

$$w_1^*\left(\sum x_i^2\right) + w_2^* \sum x_i = \left(\sum_{i=1}^{\wedge} x_i y_i\right) - \varepsilon_1 x_1 - \varepsilon_2 x_2 \quad ①$$

$$w_1^*\left(\sum_{i=1}^{\wedge} x_i\right) + w_2^* \wedge = \left(\sum_{i=1}^{\wedge} y_i\right) - \varepsilon_1 - \varepsilon_2 \quad ②$$

$$\varepsilon_1 = -w_1^* \sum^{\wedge} x_i - w_2^* \wedge - \varepsilon_2 + \sum_{i=1}^{\wedge} y_i$$

plug into ① & solve for ε_2

$$\varepsilon_2 x_2 = \sum^{\wedge} x_i y_i - \left(-w_1^* \sum x_i - w_2^* \wedge - \varepsilon_2 + \sum y_i\right) x_1 - w_1^* \sum x_i^2 - w_2^* \sum x_i$$

$$\varepsilon_2 x_2 - \varepsilon_2 x_1 = \sum x_i y_i + w_1^* x_1 \sum x_i + w_2^* \wedge x_1 - x_1 \sum y_i - w_1^* \sum x_i^2 - w_2^* \sum x_i$$

$\varepsilon_2(x_2 - x_1)$

$$\boxed{\varepsilon_2 = \frac{\left(\sum x_i y_i + w_1^* x_1 \sum x_i + w_2^* \wedge x_1 - x_1 \sum y_i - w_1^* \sum x_i^2 - w_2^* \sum x_i\right)}{(x_2 - x_1)}}$$

$$\boxed{\varepsilon_1 = \sum_{i=1}^{\wedge} x_i\left(-w_1^* - \left(\frac{w_1^* x_1}{x_2 - x_1}\right) + \left(\frac{w_2^*}{x_2 - x_1}\right)\right) + \wedge\left(-w_2^* - \left(\frac{w_2^* x_1}{(x_2 - x_1)}\right)\right) + \sum^{\wedge} y_i\left(1 + \left(\frac{x_1}{x_2 - x_1}\right)\right) - \frac{\sum^{\wedge} x_i y_i}{(x_2 - x_1)} + \frac{w_1^* \sum^{\wedge} x_i^2}{(x_2 - x_1)}}$$

thus the above mathematical mechanism shows that it is possible to manipulate to get w_1^* & w_2^* under the scenario described in the problem.

4.c. lessons taken away:

- given knowledge of Model used to fit & optimize path it is possible to directly alter points s.t the entire model output is no longer representative of the Data as a whole. ~~It doesn't~~

- the OLS ML model can be generalized to work w/ "noise" on it's data. ~~this~~

- Compartamentalizing & realizing the levels of abstraction used to set up our ML algorithm are very useful for defining goals & correct approach. In particular the 4 steps noted in lecture.

- It doesn't take a lot of manipulation of Data only a few points to alter results, increasing the amount of noise manipulation could easily change the output & twofold disguise the manipulation. Something to be wary of. Also highlights the potential large effects outlier points can have on some ML models used.

5. Background Review

Linear Algebra Math 54

Optimization IEOR 160, IEOR 165

Probability and Stochastic Processes Prob 140, Data 8 + Stat 88 Connector Course, IEOR 173, Math 55

Vector Calculus Math 53

Programming Experience CS61A (python), CS61B (java), Data 8 (statistical programming in python), Prob 140 (statistical programming in python)

6. My Question

a. Given the same set-up as Q4 but with normally distributed, non adversarial noisy y variables, break down how the lower precession of estimates from ordinary least squares regression results in less accurate predictions. How do standard errors of the prediction change in accordance with perturbations of the data?
b. What type of statistical common sense can we use to determine which variables might be outwardly noisy/potentially adversarially altered?

7. Resources Utilized, Cited

Lecture 10, Professor Jonathan Shewchuk CS189
https://www.youtube.com/watch?v=I5Xiu6vJ5lM
https://people.eecs.berkeley.edu/~jrs/189/lec/10.pdf

Math.StackExchange Post by Jack [August, 2011]
https://math.stackexchange.com/questions/55165/eigenvalues-of-the-rank-one-matrix-uvt

Ordinary Least Squares - Lecture 1/Lecture 2 Notes (Used in conjunction with personal notes taken during lecture)
http://www.eecs189.org/static/notes/n1.pdf
https://d1b10bmlvqabco.cloudfront.net/attach/jc8np1307m34ha/id1uo3l5whs1t2/jcj0f4j35xrz/CS189Notes.pdf

Linear Algebra Prereq Review Note 1
https://drive.google.com/file/d/1Mz4bpug1UkorbpSoQcL3zkauLrF9ZZ2I/view

NorthWestern, Introduction to Noisy Variables
https://www.kellogg.northwestern.edu/faculty/dranove/htm/dranove/coursepages/Mgmt%20469/noisy-variables.pdf