Michael Leibert

Math 504

Homework 1

2(a) Let u and v be two column vectors with dimension n. Show $u \cdot v = u^T v$.

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}, \quad v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \qquad \text{Definition: } \mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \ldots a_n b_n$$

$$\begin{aligned} u^T v &= \begin{pmatrix} u_1 & u_2 = \cdots & u_n \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix} \\ &= u_1 v_1 + u_2 v_2 + \cdots + u_n v_n \\ &= \sum_{i=1}^{n} u_i v_i \\ &= \mathbf{u} \cdot \mathbf{v} \end{aligned}$$

2(b) Let u be n dimensional, v by n' dimensional. Let M be a $n \times n'$ dimensional matrix. Show that $u \cdot Mv = \sum_{i=1}^{n} \sum_{j=1}^{n'} M_{ij} u_i v_j$. (Hint: Start by noticing that the kth coordinate of Mv, $(Mv)_k$, is given by $(Mv)_k = \sum_{j=1}^{n} M_{kj} v_j$)

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}, \quad v = \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n'} \end{pmatrix}, \quad M = \begin{pmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,n'} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n'} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n'} \end{pmatrix}$$

$$u \cdot Mv = u \cdot \begin{pmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,n'} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n'} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n'} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n'} \end{pmatrix}$$

$$= u \cdot \begin{pmatrix} \sum_{j=1}^{n'} m_{1j} v_j \\ \sum_{j=1}^{n'} m_{2j} v_j \\ \vdots \\ \sum_{j=1}^{n'} m_{nj} v_j \end{pmatrix} \qquad \text{Let } \mu = \begin{pmatrix} \sum_{j=1}^{n'} m_{1j} v_j \\ \sum_{j=1}^{n'} m_{2j} v_j \\ \vdots \\ \sum_{j=1}^{n'} m_{nj} v_j \end{pmatrix}$$

$$\begin{aligned} &= \mathbf{u} \cdot \boldsymbol{\mu} \\ &= \sum_{i=1}^{n} u_i \mu_i = \sum_{i=1}^{n} u_i \left(\sum_{j=1}^{n'} m_{ij} v_j \right) = \sum_{i=1}^{n} \sum_{j=1}^{n'} m_{ij} u_i v_j \end{aligned}$$

2(c) Let b and x be vectors in $\mathbb{R}^n$, Show that $\nabla(b^T x) = b$. (We did this in class.)

$$\begin{aligned}
\nabla\left(b^T x\right) &= \nabla\left(\sum_{i=1}^{n} b_i x_i\right) \\
&= \nabla\left(b_1 x_1 + b_2 x_2 + \cdots + b_n x_n\right) \\
&= \begin{pmatrix} \dfrac{\partial}{\partial x_1}\, b_1 x_1 + b_2 x_2 + \cdots + b_n x_n \\ \dfrac{\partial}{\partial x_2}\, b_1 x_1 + b_2 x_2 + \cdots + b_n x_n \\ \vdots \\ \dfrac{\partial}{\partial x_n}\, b_1 x_1 + b_2 x_2 + \cdots + b_n x_n \end{pmatrix} \\
&= \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \\
&= \mathbf{b}
\end{aligned}$$

2(d) Let A be an $n \times n$ matrix and $x \in \mathbb{R}^n$. Show that $\nabla(x^T A x) = (A + A^T)x$. Hint: Use (b).

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, \quad A^T = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{pmatrix}$$

$$x^T A x = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

$$\begin{aligned}
&= \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix} \begin{pmatrix} \sum_{j=1}^{n} a_{1j} x_j \\ \sum_{j=1}^{n} a_{2j} x_j \\ \vdots \\ \sum_{j=1}^{n} a_{1j} x_j \end{pmatrix} \\
&= x_1 \sum_{j=1}^{n} a_{1j} x_j + x_2 \sum_{j=1}^{n} a_{2j} x_j + ... + x_n \sum_{j=1}^{n} a_{nj} x_j
\end{aligned}$$

$$\nabla\left(x^T A x\right) = \begin{pmatrix} \dfrac{\partial}{\partial x_1}\left[x_1 \sum\limits_{j=1}^{n} a_{1j}x_j + x_2 \sum\limits_{j=1}^{n} a_{2j}x_j + ... + x_n \sum\limits_{j=1}^{n} a_{nj}x_j\right] \\ \dfrac{\partial}{\partial x_2}\left[x_1 \sum\limits_{j=1}^{n} a_{1j}x_j + x_2 \sum\limits_{j=1}^{n} a_{2j}x_j + ... + x_n \sum\limits_{j=1}^{n} a_{nj}x_j\right] \\ \vdots \\ \dfrac{\partial}{\partial x_n}\left[x_1 \sum\limits_{j=1}^{n} a_{1j}x_j + x_2 \sum\limits_{j=1}^{n} a_{2j}x_j + ... + x_n \sum\limits_{j=1}^{n} a_{nj}x_j\right] \end{pmatrix}$$

$$= \begin{pmatrix} 2a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n + a_{21}x_2 + ... + a_{n1}x_n \\ a_{12}x_1 + a_{21}x_1 + 2a_{22}x_2 + ... + a_{2n}x_n + ... + a_{n2}x_n \\ \vdots \\ a_{1n}x_1 + a_{2n}x_2 + ... + a_{n1}x_1 + a_{n2}x_2 + ... + 2a_{nn}x_n \end{pmatrix}$$

$$= \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n + a_{11}x_1 + a_{21}x_2 + ... + a_{n1}x_n \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n + a_{12}x_1 + a_{22}x_2 + ... + a_{n2}x_n \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n + a_{1n}x_1 + a_{2n}x_2 + ... + a_{nn}x_n \end{pmatrix}$$

$$= \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n \end{pmatrix} + \begin{pmatrix} a_{11}x_1 + a_{21}x_2 + ... + a_{n1}x_n \\ a_{12}x_1 + a_{22}x_2 + ... + a_{n2}x_n \\ \vdots \\ a_{1n}x_1 + a_{2n}x_2 + ... + a_{nn}x_n \end{pmatrix}$$

$$= \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} + \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

$$= \left(A + A^T\right) x$$

2(e) Explain why a general quadratic in $\mathbb{R}^n$ can be written as $f(x) = x^T Ax + b^T x + c$. No proof here, just explain the intuition.

The matrix-vector notation allows us to take an expression with n terms and multiple sums and compacted it into the form $x^T Ax + b^T + c$. We replaced summations with matrix multiplication and the use of dot products.

2(f) Let $A = \begin{pmatrix} 1 & 4 \\ 2 & 1 \end{pmatrix}$ and let $f(x) = x^T Ax$.

Show that $f(x)$ can be rewritten as $f(x) = x^T Bx$ with B symmetric. (Hint: write out $f(x)$ as a sum of terms rather than a matrix expression.) Then explain why for the general quadratic in (e), we can always asssume that A is symmetric.

$$A = \begin{pmatrix} 1 & 4 \\ 2 & 1 \end{pmatrix} \qquad B = \frac{A + A^T}{2} = \begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix}$$

$$\begin{aligned} x^T Ax &= \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 1 & 4 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ &= \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} x_1 + 4x_2 \\ 2x_1 + x_2 \end{pmatrix} \\ &= x_1(x_1 + 4x_2) + x_2(2x_1 + x_2) \\ &= x_1^2 + 4x_1x_2 + 2x_1x_2 + x_2^2 \\ &= x_1^2 + 6x_1x_2 + x_2^2 \end{aligned}$$

$$\begin{aligned} x^T Bx &= \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} 1 & 3 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ &= \begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} x_1 + 3x_2 \\ 3x_1 + x_2 \end{pmatrix} \\ &= x_1(x_1 + 3x_2) + x_2(3x_1 + x_2) \\ &= x_1^2 + 3x_1x_2 + 3x_1x_2 + x_2^2 \\ &= x_1^2 + 6x_1x_2 + x_2^2 \end{aligned}$$

We can assume A is symmetric because A is a square matrix and can be written as a sum, $A = A_S + A_A$. Where, $A_S = \frac{1}{2}\left(A + A^T\right)$, is a symmetric matrix known as the symmetric part of A; and $A_A = \frac{1}{2}\left(A - A^T\right)$, is an antisymmetric matrix known as the antisymmeric part of A.

The antisymmertic matrix is skew symmetric since, $A_A^T = \left(\frac{A - A^T}{2}\right)^T = \frac{A^T - A}{2} = -\frac{A - A^T}{2} = -A_A$. Also because $x^T A_A x$ is a scalar, $x^T A_A x = \left(x^T A_A x\right)^T = x^T A_A^T x$.

$$\begin{aligned} x^T \; A_A \; x &= x^T A_A^T \; x \\ x^T \; A_A \; x &= -x^T A_A \; x \\ 2x^T \; A_A \; x &= 0 \\ x^T A_A \; x &= 0 \end{aligned}$$

Now we can write,

$$\begin{aligned} x^T Ax &= x^T \left(\frac{A + A^T}{2} + \frac{A - A^T}{2}\right) x \\ &= x^T \left(\frac{A + A^T}{2}\right) x + x^T \left(\frac{A - A^T}{2}\right) x \\ &= x^T A_S x + x^T A_A x \\ &= x^T A_S x + 0 \\ &= x^T A_S x \end{aligned}$$

And we know A_S is symmetric. Because $x^T A x$ ignores the antisymmertic part of A we can assume A is symmetric.

2(g) Let $x \in \mathbb{R}^n$ and $f(x) = x^T A x + b^T x + c$ where A is an $n \times n$ matrix, b is a n dimensional vector and c is a scalar. Show that $\nabla f(x) = (A + A^T)x + b$. Solve for the critical point of $f(x)$. (We essentially did this in class.)

$$\begin{aligned}
\nabla f(x) &= \nabla \left(x^T A x + b^T x + c\right) \\
&= \nabla \left(x^T A x\right) + \nabla \left(b^T x\right) + \nabla c \\
&= \left(A + A^T\right) x + \mathbf{b}
\end{aligned}$$

$$\begin{aligned}
\mathbf{0} &= \left(A + A^T\right) x + \mathbf{b} \\
\left(A + A^T\right) x &= -\mathbf{b} \\
x &= -\left(A + A^T\right)^{-1} \mathbf{b} \\
x &= -\frac{1}{2} A^{-1} \mathbf{b}
\end{aligned}$$

3 Download the datafile `economic_data.txt` using the `read.table` function, don't forget to set `header=T`. See the datafile for details regarding the column values and meaning. Assume the linear model $B \sim \alpha_0 + \sum_{i=1}^{6} \alpha_i A_i$.

3(a) Fit this model using the normal equations to determine α. To do this, first derive the normal equations. (I did this in class, I just want you to go through the argument yourself. You can use your results from problem 2.) You will find that R gives you an error, saying that the matrix involved is computational singular. (The problem here has to do with calculating the inverse in the normal equations, we will explore this issue in the coming weeks.)

Derivation of the normal equations.

Objective: $\min\limits_{\alpha \in \mathbb{R}^{n+1}} L(\alpha) \Longrightarrow \nabla L(\alpha) = 0 \Longrightarrow \alpha = -\dfrac{1}{2} A^{-1} b$

$$L(\alpha) = \sum_{i=1}^{N} \left(\underbrace{y_i - \alpha_0 + \alpha_1 x_1^{(i)} + \alpha_2 x_2^{(i)} + \ldots \alpha_n x_n^{(i)}}_{r_i} \right)^2$$

$$r_i = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{pmatrix} = \begin{pmatrix} y_1 - \alpha_0 + \alpha_1 x_1^{(1)} + \alpha_2 x_2^{(1)} + \ldots + \alpha_n x_n^{(1)} \\ y_2 - \alpha_0 + \alpha_1 x_1^{(2)} + \alpha_2 x_2^{(2)} + \ldots + \alpha_n x_n^{(2)} \\ \vdots \\ y_N - \alpha_0 + \alpha_1 x_1^{(N)} + \alpha_2 x_2^{(N)} + \ldots + \alpha_n x_n^{(N)} \end{pmatrix}$$

$$= \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} - \begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \ldots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \ldots & x_n^{(2)} \\ \vdots & & & & \\ 1 & x_1^{(N)} & x_2^{(N)} & \ldots & x_n^{(N)} \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix}$$

$$\mathbf{r} = Y - B\alpha$$

$$
\begin{aligned}
L(\alpha) &= \sum_{i=1}^{N}\left(y_i - \alpha_0 + \alpha_1 x_1^{(i)} + \alpha_2 x_2^{(i)} + ...\alpha_n x_n^{(i)}\right)^2 \\
&= \sum_{i=1}^{N} r_i^2 \\
&= \mathbf{r}\cdot\mathbf{r} \\
&= (y - B\alpha)\cdot(y - B\alpha) \\
&= (y - B\alpha)^T (y - B\alpha) \\
&= \left(y^T - (B\alpha)^T\right)(y - B\alpha) \\
&= y^T y - (B\alpha)^T y - y^T B\alpha + (B\alpha)^T B\alpha \\
&= y^T y - 2(B^T y)^T\alpha + \alpha^T B^T B\alpha \\
\\
&\text{Let } A = B^T B,\ b = -2B^T y \text{ and } c = y^T y \\
&= \alpha^T A\alpha + b^T\alpha + c
\end{aligned}
$$

Normal equations

$$
\begin{aligned}
\alpha &= -\frac{1}{2}\left(B^T B\right)^{-1}\left(-2B^T y\right) \\
&= \left(B^T B\right)^{-1} B^T y
\end{aligned}
$$

```r
library(RCurl)
```

```
## Warning: package 'RCurl' was built under R version 3.3.3
## Loading required package: bitops
```

```r
dat <- getURL("https://raw.githubusercontent.com/mleibert/504/master/economic_data.txt")
dat <- read.table(text = dat ,header=T)
options(scipen = 999)

y<-dat[,ncol(dat)]
B<-as.matrix(dat[,2:(ncol(dat)-1)])
B<-cbind(1,B)

#solve(t(B) %*% B ) %*% t(B) %*% y
```

3(b) Now repeat (a), but throw out covariates until R does not issue an error. Can you determine the covariates responsible for the error?

```r
cor(B[,-1])
```

```
##           A1        A2         A3         A4        A5        A6
## A1 1.0000000 0.9915892  0.6205874  0.4647442 0.9791634 0.9911492
## A2 0.9915892 1.0000000  0.6042089  0.4464368 0.9910901 0.9952735
## A3 0.6205874 0.6042089  1.0000000 -0.1775504 0.6865068 0.6682035
## A4 0.4647442 0.4464368 -0.1775504  1.0000000 0.3644163 0.4172451
## A5 0.9791634 0.9910901  0.6865068  0.3644163 1.0000000 0.9939528
## A6 0.9911492 0.9952735  0.6682035  0.4172451 0.9939528 1.0000000
```

The variables A2, A5, and A6 have multicollinearity issues which is causing the the matrix to be singluar.

```
BB<-B[,-c(6,7)]
cor(BB[,-1])
```

```
##           A1        A2         A3         A4
## A1 1.0000000 0.9915892  0.6205874  0.4647442
## A2 0.9915892 1.0000000  0.6042089  0.4464368
## A3 0.6205874 0.6042089  1.0000000 -0.1775504
## A4 0.4647442 0.4464368 -0.1775504  1.0000000
```

```
solve(t(BB) %*% BB ) %*% t(BB) %*% y
```

```
##              [,1]
##    50135.42355558
## A1    55.34521216
## A2     0.03537306
## A3    -0.85377061
## A4    -0.54975517
```

3(c) Repeat (a) and (b), but fit the model using the **lm** function in R. Compare your results to (a),(b).

```
dat<-dat[,-1]
( lm(B~.,data=dat) )
```

```
##
## Call:
## lm(formula = B ~ ., data = dat)
##
## Coefficients:
##    (Intercept)               A1               A2               A3
## -3475440.82413         14.78948         -0.03575         -2.02020
##             A4               A5               A6
##       -1.03277         -0.04912       1825.54365
```

```
BB<-BB[,-1]
BB<-as.data.frame(BB)
BB$B<-dat$B
( lm(B~.,data=BB) )
```

```
##
## Call:
## lm(formula = B ~ ., data = BB)
##
## Coefficients:
## (Intercept)           A1           A2           A3           A4
## 50135.42356     55.34521      0.03537     -0.85377     -0.54976
```

The results do not look at all similar between the lm function and the normal equations with A5 and A6 removed. With the adjusted dataframe with A5 and A6 removed, the α's are exactly the same.

4 See how fast your computer is. Let $f(x) = x^2 + 3x + 2$. Write a R function **exhaustive(gridSize)** that minimizes $f(x)$ by testing a grid of points starting at -10, ending at 10, and with step size given by **gridSize**. How small can you make **gridSize** before the computation takes more than roughly 1 seconds? Decide how many multiplications your computer+R can do in 1 second.

```
exhustive<-function(gridSize){

    fx<-function(x){x^2+3*x+2}

    minimize<-data.frame( seq(-10,10,by=1/(gridSize)),
        fx( seq(-10,10,by=1/(gridSize)) )   )
    minimize<-minimize[which(minimize[,2] ==
        min(minimize[,2])), ]
        names(minimize)<-c("x","MIN")
    times<-system.time( min( fx( seq(-10,10,by=1/(gridSize)) ))  )[3]
    print(times);print(minimize)
}


######


 times<-rep(NA,10)

for ( i in 1:10) {

  times[i]<-system.time( for ( j in 1:(10^i) ) { 1*1 } )[1]
  if ( times[i] > 1 ) { break}}
times
```

```
##  [1] 0.00 0.00 0.00 0.00 0.02 0.28 2.53   NA   NA   NA
```

```
system.time( for( i in 1: (1/times[7] * 10^7) ) { 1*1} )
```

```
##    user  system elapsed
##    1.04    0.00    1.04
```

```
1/times[7] * 10^7
```

```
## [1] 3952569
```

```
######

exhustive(2000000)
```

```
## elapsed
##    1.22
##             x   MIN
## 17000001 -1.5 -0.25
```

About four million multiplications per second.