Math 151BH - HW2

Will Firmin

May 1, 2023

Problem 1:

(a)

$$\begin{aligned}
\frac{\|f(x+h)-f(x)-D_f(x)h\|_Y}{\|h\|_X} &\geq \left|\frac{\|f(x+h)-f(x)\|_Y-\|D_f(x)h\|_Y}{\|h\|_X}\right| \\
\Rightarrow \lim_{\|h\|_X\to 0}\left|\frac{\|f(x+h)-f(x)\|_Y-\|D_f(x)h\|_Y}{\|h\|_X}\right| &\leq 0 \\
\Rightarrow \lim_{\|h\|_X\to 0}\left|\frac{\|f(x+h)-f(x)\|_Y-\|D_f(x)h\|_Y}{\|h\|_X}\right| &= 0 \\
\Rightarrow \lim_{\|h\|_X\to 0}\frac{\|f(x+h)-f(x)\|_Y-\|D_f(x)h\|_Y}{\|h\|_X} &= 0 \\
\Rightarrow \lim_{\|h\|_X\to 0}\frac{\|f(x+h)-f(x)\|_Y}{\|h\|_X} &= \lim_{\|h\|_X\to 0}\frac{\|D_f(x)\|_Y}{\|h\|_X} \\
&\leq \lim_{\|h\|_X\to 0}\frac{\|D_f(x)\|\|h\|_X}{\|h\|_X} \\
&= \|D_f(x)\| \\
\text{Let } \lim_{\|h\|_X\to 0}\frac{\|f(x+h)-f(x)\|_Y}{\|h\|_X} &= L \\
\text{Then } \forall\epsilon>0,\ \exists\delta>0\ s.t.\ \|h\|_X\leq\delta\Rightarrow\frac{\|f(x+h)-f(x)\|_Y}{\|h\|_X} &< L+\epsilon\leq\|D_f(x)\|+\epsilon
\end{aligned}$$

(b)

$$\lim_{\|h\|_X \to 0} \frac{\|f(x)+h) - f(x)\|_Y}{\|h\|_X} = \lim_{\|h\|_X \to 0} \frac{\|D_f(x)h\|_Y}{\|h\|_X}$$
$$= \lim_{k \to 0} \frac{k\|D_f(x)u\|_Y}{k\|u\|_X} \text{ for } ku = h,\ \|u\|_X = 1$$
$$= \|D_f(x)u\|_Y$$
$$\geq \|D_f(x)\| - \epsilon \text{ for some choice of } h, u$$
$$\text{Let } \tilde{u} \text{ be this choice of } u \text{ with } \tilde{h}_\delta \text{ s.t. } \|\tilde{h}_\delta\|_X = \delta$$
$$\Rightarrow \lim_{\delta \to 0} \frac{\|f(x)+\tilde{h}_\delta) - f(x)\|_Y}{\|\tilde{h}_\delta\|_X} = L > \|D_f(x)\| - \epsilon$$
$$\text{Then, same as part a, } \forall \epsilon > 0\ \exists \delta > 0\ s.t.\ \frac{\|f(x)+\tilde{h}_\delta) - f(x)\|_Y}{\|\tilde{h}_\delta\|_X} > L - \epsilon > \|D_f(x)\| - 2\epsilon$$
$$\text{Finally, we have that } \forall \epsilon > 0\ \exists \delta > 0\ s.t.$$
$$\|D_f(x)\| - 2\epsilon < \sup_{\|h\|_X \leq \delta} \frac{\|f(x)+h) - f(x)\|_Y}{\|h\|_X} < \|D_f(x)\| + \epsilon$$
$$\Rightarrow \forall \epsilon > 0\ \exists \delta > 0\ s.t.\ \left| \sup_{\|h\|_X \leq \delta} \frac{\|f(x)+h) - f(x)\|_Y}{\|h\|_X} - \|D_f(x)\| \right| < 2\epsilon$$
$$\Rightarrow \lim_{\delta \to 0} \sup_{\|h\|_X \leq \delta} \frac{\|f(x)+h) - f(x)\|_Y}{\|h\|_X} = \|D_f(x)\|$$

Problem 2:

(a) We clearly have symmetry since $a_{ij} = a_{ji}$ by symmetry of A.
Let X be the last k columns of $I \in \mathbb{R}^{n \times n}$. Then $AX =$ the right k columns of A, and $X^\top AX =$ the bottom k rows of AX and thus the $k \times k$ lower right principal submatrix. Therefore the principal submatrix is also positive definite.

(b)

$$B_1 = \begin{bmatrix} 0 & 0 \\ 0 & K \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & -\frac{ww^\top}{a_{11}} \end{bmatrix}$$

The first matrix in the above decomposition is a lower right principal submatrix, so it is also symmetric. The second matrix is clearly symmetric at the 1 and 0 entries. $(ww^\top)^\top = ww^\top$, so the lower right corner is also symmetric. Since both above matrices are symmetric, their sum B_1 is also symmetric.

Let $v \neq 0$. Then $v^\top B_1 v = v^\top L^{-1} L B_1 L_1^\top (L_1^\top)^{-1} v = (v^\top L^{-1}) A (v^\top L^{-1})^\top > 0$ by positive definiteness of A. So B_1 is symmetric positive definite.

Problem 3:

Let L_i denote the ith row of L. Then $\alpha_i = L_i(L_i)^\top = l_i^2 + v_i^2$ for $i > 1$ with $l_1^2 = \alpha_1$. Also, $\gamma_i = L_i(L_{i-1})^\top = l_{i-1}v_i$ for $i > 1$. This gives $v_i = \frac{\gamma_i}{l_{i-1}}$. Combining both equations gives the recursive formula $l_i = \sqrt{\alpha_i - \frac{\gamma_i^2}{l_{i-1}^2}}$ for $i > 1$ starting with $l_1 = \sqrt{\alpha_1}$. Rearranging the same equations gives $v_i = \gamma_i(\alpha_{i-1} - v_{i-1}^2)^{-\frac{1}{2}}$ for $i > 2$ with $v_2 = \frac{\gamma_2}{\sqrt{\alpha_1}}$.

Problem 4:

```
import numpy as np
import numpy.random

np.random.seed(151)
def Cholesky(n):
    b = np.random.normal(size=n)
    gamma = np.random.uniform(size=n)
    alpha = gamma+1
    gamma[0] = 0

    l = np.zeros(n)
    l[0] = alpha[0]
    for i in range(1,n):
        l[i] = alpha[i] - gamma[i]**2 / l[i-1]
    l = np.sqrt(l)
    v = gamma[1:] / l[:-1]
    v = np.insert(v,0,0)

    # Solve Ly=b:
    y = [b[0]/l[0]]
    for i in range(1,n):
        y.append((b[i]-y[-1]*v[i])/l[i])

    # Solve L^Tx=y:
    x = np.zeros(n)
    x[n-1] = y[n-1]/l[n-1]
    for i in range(n-2,-1,-1):
        x[i] = (y[i]-x[i+1]*v[i+1])/l[i]

    # Calculate Ax:
    b2 = np.zeros(n)
    for i in range(1,n-1):
        b2[i] = gamma[i]*x[i-1] + alpha[i]*x[i] + gamma[i+1]*x[i+1]
    b2[0] = alpha[0]*x[0]+gamma[1]*x[1]
    b2[-1] = gamma[-1]*x[-2]+alpha[-1]*x[-1]

    return np.max(np.abs(b2-b))

for N in [10,100,1000,10000]:
    print("Infinity norm of error for N = {}: {}".format(N,Cholesky(N)))
```

Output:

```
Infinity norm of error for N = 10: 1.1102230246251565e-15
Infinity norm of error for N = 100: 2.6645352591003757e-15
Infinity norm of error for N = 1000: 1.1102230246251565e-15
Infinity norm of error for N = 10000: 2.9976021664879227e-15
```

Problem 5:

$$\begin{aligned}
\exists v \neq 0 \; s.t. \; Av = \lambda v \\
\Rightarrow \sum_j v_j a_{ij} = \lambda v_i \\
\Rightarrow \lambda = \frac{\sum_j v_j a_{ij}}{v_i} \text{ for } v_i \neq 0 \\
\Rightarrow \lambda = \sum_{j \neq i} \frac{v_j}{v_i} a_{ij} + a_{ii} \\
\Rightarrow |\lambda - a_{ii}| = |\sum_{j \neq i} \frac{v_j}{v_i} a_{ij}| \\
\Rightarrow |\lambda - a_{ii}| \leq \sum_{j \neq i} |\frac{v_j}{v_i} a_{ij}| \\
\text{Let } i = \arg\max_k |v_k| \Rightarrow |\frac{v_j}{v_i}| \leq 1 \\
\Rightarrow |\lambda - a_{ii}| \leq \sum_{j \neq i} |a_{ij}| = r_i \\
\Rightarrow \lambda \in B_{a_i i}(r_i)
\end{aligned}$$

Problem 6:

(a) Suppose A is singular. Then $\dim(\ker(A)) > 0 \Rightarrow \exists v \neq 0$ s.t. $Av = 0$: $\sum_j a_{ij} v_j = 0 \; \forall i$. Fix $i = \arg\max_k |v_k|$. Then

$$\begin{aligned}
a_{ii} &= -\sum_{j \neq i} a_{ij} \\
\Rightarrow |a_{ii}| &= |\sum_{j \neq i} a_{ij} \frac{v_j}{v_i}| \\
&\leq \sum_{j \neq i} |a_{ij} \frac{v_j}{v_i}| \\
&\leq \sum_{j \neq i} |a_{ij}| \Rightarrow\Leftarrow
\end{aligned}$$

(b)

$$\begin{aligned}
v^\top A v &= \sum_i \sum_j a_{ij} v_i v_j \\
&= \sum_i a_{ii} v_i^2 + \sum_i \sum_{j \neq i} a_{ij} v_i v_j \\
&> \sum_i \sum_{j \neq i} |a_{ij}| v_i^2 + \sum_i \sum_{j \neq i} a_{ij} v_i v_j \\
&\geq \sum_i \sum_{j \neq i} |a_{ij}| (v_i^2 - |v_i v_j|)
\end{aligned}$$

Problem 7:

Suppose $\|P\|_2 \in (0,1)$. Then $\|P\|_2 = \|P^2\|_2 \leq \|P\|_2^2 < \|P\|_2 \Rightarrow\Leftarrow$ So $\|P\|_2 \geq 1$.

Suppose P is an orthogonal projection: then $v = Pv + r \Rightarrow r \perp Pv$. By the Pythagorean Theorem, $\|v\|^2 = \|P\|^2 + \|r\|^2 = 1 \; \forall \|v\| = 1$. Then $\|Pv\|^2 \leq 1 \Rightarrow \|Pv\| \leq 1 \Rightarrow \|P\| = 1$.

Suppose P is not an orthogonal projection: then $\mathrm{Im}(I - P) \not\perp \mathrm{Im}(P)$. For $r = v - Pv$, $\|v\| = 1$, pick $v \notin \mathrm{Im}(P)$ s.t. $v \perp \mathrm{Im}(I - P)$. Then by the Pythagorean Theorem $\|Pv\|^2 = \|v\|^2 + \|r\|^2 = 1 + \|r\|^2 > 1$. Therefore $\|P\| > 1$.

Together, this gives that $\|P\| = 1 \iff$ P is an orthogonal projector.

Problem 8:

$$
\begin{aligned}
x^* &= (A^*A)^{-1}A^*b \\
&= ((QR)^*QR)^{-1}(QR)^*b \\
&= (R^*Q^*QR)^{-1}(QR)^*b \\
&= (R^*R)^{-1}R^*Q^*b \\
&= R^{-1}(R^*)^{-1}R^*Q^*b \\
&= R^{-1}Q^*b
\end{aligned}
$$

Problem 9: The solution is incorrect but I'm too tired to keep debugging

```
A1 = np.array([[-0.897,-1.004,-0.284,0.822],
               [1.066,-0.484,-1.327,-0.679],
               [1.33,1.37,-0.878,0.678],
               [1.368,0.669,-0.392,0.221],
               [-0.964,0.84,1.497,-0.427]])
b1 = np.array([-0.847,-0.829,-0.579,0.831,-1.088])


def SolveLSQR(A,b):
    m,n=A.shape
    v = np.zeros((m,n))
    A = np.vstack([A,np.zeros(n)])

    for j in range(n):
        x = A[j:m,j]
        A[j+1:,j] = np.sign(x[0])*np.linalg.norm(x,2)*np.eye(1,m-j,0)+x
        A[j+1:,j] /= np.linalg.norm(A[j+1:,j],2)
        A[j:m,j:]=A[j:m,j:]-2*np.outer(A[j+1:,j],np.matmul(A[j+1:,j],A[j:m,j:]))
        print(A)

    # y = QTb:
    y = np.zeros(n)
    for j in range(n):
        y[j] = b[:m-j].dot(A[j+1:,j])

    # Rx=y: this part is confirmed to work
    x = np.zeros(n)
    x[n-1] = y[n-1]/A[n-1,n-1]
    for j in range(n-2,-1,-1):
        x[j] = (y[j]-x[j+1:].dot(A[j,j+1:]))/A[j,j]
    print(x)

SolveLSQR(A1,b1)
```

Output:

```
[[ 0.15112678  0.90617251 -1.68794625  0.06056942]
 [-1.14604309 -1.07445783 -0.89302294 -0.44363244]
 [-0.15011922  0.63331247 -0.33654644  0.97165747]
 [-0.09873771 -0.08873575  0.16492366  0.52304768]
 [ 0.6190842   1.37395999  1.10454795 -0.63984646]
 [-0.22979289  0.          0.          0.        ]]
```

```
[[ 0.15112678  0.90617251 -1.68794625  0.06056942]
 [-1.14604309  0.28953842  1.21079962  0.08962018]
 [-0.15011922 -1.18296457 -0.79094235  0.85648245]
 [-0.09873771  0.23315138  0.22859075  0.53918528]
 [ 0.6190842  -0.66602193  0.1187442  -0.8897166 ]
 [-0.22979289  0.41626575  0.          0.        ]]
[[ 0.15112678  0.90617251 -1.68794625  0.06056942]
 [-1.14604309  0.28953842  1.21079962  0.08962018]
 [-0.15011922 -1.18296457  0.50051677 -0.79321848]
 [-0.09873771  0.23315138 -1.169555    0.77156908]
 [ 0.6190842  -0.66602193  0.04462166 -0.76900205]
 [-0.22979289  0.41626575  0.07226878  0.        ]]
[[ 0.15112678  0.90617251 -1.68794625  0.06056942]
 [-1.14604309  0.28953842  1.21079962  0.08962018]
 [-0.15011922 -1.18296457  0.50051677 -0.79321848]
 [-0.09873771  0.23315138 -1.169555    0.1059268 ]
 [ 0.6190842  -0.66602193  0.04462166  1.1992666 ]
 [-0.22979289  0.41626575  0.07226878 -0.38191358]]
[-341.92231689   43.48875778   -8.63883168   -6.6005247 ]
```
