OPTIMIZACIÓN

Primer Cuatrimestre 2019

Práctica N° 0: Repaso.

Repaso de Álgebra Lineal

Ejercicio 1 Sea $A \in \mathbb{R}^{n\times n}$. Probar que la aplicación $<,>: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ dada por $< v, w >= v^t Aw$ define un producto interno si y sólo si A es simétrica y definida positiva.

Ejercicio 2 Sea $A \in \mathbb{R}^{n\times n}$ simétrica. Probar que si $\lambda \in \mathbb{C}$ es autovalor de A entonces $\lambda \in \mathbb{R}$. Si además A es definida positiva, probar que $\lambda > 0$.

Ejercicio 3 Sea $A \in \mathbb{R}^{n\times n}$ simétrica y sean v_1 y v_2 autovectores de A correspondientes a los autovalores λ_1 y λ_2 respectivamente. Probar que si $\lambda_1 \neq \lambda_2$ entonces v_1 y v_2 son ortogonales con el producto interno usual.

Ejercicio 4 Sea $A \in \mathbb{R}^{n\times n}$. Probar que A es simétrica si y sólo si $O^t AO$ es simétrica para toda $O \in \mathbb{R}^{n\times n}$ ortogonal (es decir, $O^{-1} = O^t$).

Ejercicio 5 Sea $B = \{q_1, ..., q_n\}$ una base ortonormal. Probar que la matriz Q que tiene por columnas a los vectores q_i es ortogonal.

Ejercicio 6 Probar que si $A \in \mathbb{R}^{n\times n}$ es simétrica, entonces admite una base ortonormal de autovectores.

Ejercicio 7 Sea $A \in \mathbb{R}^{n\times n}$ simétrica y definida positiva, y sea $A_k \in \mathbb{R}^{k\times k}$ la submatriz con coeficientes (a_{ij}) para $1 \leq i, j \leq k$. Probar que A_k es simétrica definida positiva $(k = 1, 2, \dots, n)$. Concluir que $\det(A_k) > 0$ para todo k. Los valores $\det(A_k)$ se llaman *menores principales* de A.

Ejercicio 8 Dado un subespacio $\mathbb{S}$, un $x \in \mathbb{R}^n$ y un producto interno $<,>$, probar que la proyección ortogonal de x sobre $\mathbb{S}$ es $p = \sum_{i=1}^{n} < x, v_i > v_i$, donde los v_i forman una base ortonormal de $\mathbb{S}$.

Repaso de Análisis

Ejercicio 9 Calcular la fórmula de Taylor de segundo orden para las funciones dadas en el punto indicado. Escribir la forma de Lagrange del residuo.

a) $f(x, y) = (x + y)^2$ en $(0, 0)$ b)$f(x, y) = e^{x+y}$ en $(0, 0)$

Ejercicio 10 Sea $f(x, y) = xe^y$. Calcular el polinomio de Taylor de orden 1 de f en el punto $P = (1, 0)$. Usar este polinomio para aproximar el valor $f(0{,}98, 0{,}02)$. Estimar el error cometido.

Ejercicio 11 Calcular los puntos críticos de $f(x, y) = xe^{-y^2} + x^2$, $g(x, y) = x^2 + y^4$ y $h(x, y) = x^4 - y^4$ y sus hessianos en dichos puntos. ¿Son máximos locales, mínimos locales o puntos silla? ¿Son globales?

Ejercicio 12 Sea $f \in C^2(\mathbb{R}^n)$ tal que tiene un extremo estricto en $a \in \mathbb{R}^n$. ¿Es necesariamente $Hf(a)$ definida positiva o negativa?

Ejercicio 13 Determinar los extremos absolutos de $f|_A$ en los siguientes casos:

a. $f(x,y) = xy(x-y)^2 \qquad A = \{(x,y) \in \mathbb{R}^2 : x > 0, y > 0\}$.

b. $f(x,y) = 2x + 4y - x^2 - y^2 - 3 \qquad A = \mathbb{R}^2$.

c. $f(x,y) = 2x + 4y - x^2 - y^2 - 3 \qquad A = \{(x,y) \in R^2 : |x| \leq 1, |y| \leq 1\}$.

Ejercicio 14 La temperatura de una placa en un punto cualquiera (x,y) viene dada por la función $T(x,y) = 25 + 4x^2 - 4xy + y^2$. Una alarma térmica, situada sobre los puntos de la circunferencia $x^2 + y^2 = 25$, se dispara a temperaturas superiores a 180 grados o inferiores a 20 grados. ¿Se disparará la alarma?

Repaso Cálculo Numérico

Ejercicio 15 Decidir si son verdaderas o falsas las siguientes afirmaciones y justificar:

a. n es $O(n)$

b. $3n^2 + 7n + 4$ es $O(n^2)$

c. n^i es $O(n^j)$ si $i < j$

Ejercicio 16 Mostrar que $n!$ no es $O(r^n)$, cualquiera sea el entero positivo r. Concluir que una complejidad de $O(n!)$ es "peor" que una complejidad exponencial de cualquier base.

Ejercicio 17 Considerar el problema de cuadrados mínimos, es decir el problema de buscar, dadas por $A \in \mathbb{R}^{n\times m}$ $(n > m)$ y $b \in \mathbb{R}^n$, un punto $x^* \in \mathbb{R}^m$ tal que:

$$\|Ax^* - b\| = \min_{x\in\mathbb{R}^m} \|Ax - b\|.$$

Probar que la descomposición QR de A permite resolver este problema. En particular, que si $A = QR$, entonces x^* satisface que $Rx^* = Q^t b$.

Ejercicio 18 Dada $A \in \mathbb{R}^{n\times m}$, con descomposición en valores singulares $A = U\Sigma V^*$, se define $A^\dagger = V\Sigma^\dagger U^*$ la *pseudoinversa* de Moore-Penrose de A, donde $\Sigma^\dagger$ es la matriz resultante de invertir los elementos no nulos de Σ y trasponerla:

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & 0 & \dots & & & 0 \\ 0 & \sigma_2 & 0 & \dots & & & \\ 0 & 0 & \ddots & \dots & & & \vdots \\ \vdots & & & \sigma_r & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \\ \vdots & & & & & & \vdots \\ 0 & \dots & & & & & 0 \end{pmatrix} \longrightarrow \Sigma^\dagger = \begin{pmatrix} \frac{1}{\sigma_1} & 0 & 0 & \dots & & & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_2} & 0 & \dots & & & \vdots & & \\ 0 & 0 & \ddots & \dots & & & \vdots & & \\ \vdots & & & \frac{1}{\sigma_r} & & & & & \\ & & & & 0 & & & & \\ & & & & & \ddots & & & \vdots \\ & & & & & & 0 & \dots & 0 \end{pmatrix}$$

a) Probar que $A^\dagger$ satisface las condiciones de Penrose (En muchas ocasiones $A^\dagger$ se define como la única matriz que cumple estas condiciones):

   i. $A^\dagger A A^\dagger = A^\dagger$

   ii. $AA^\dagger A = A$

   iii. $(AA^\dagger)^* = AA^\dagger$

   iv. $(A^\dagger A)^* = A^\dagger A$

b) Concluir que la transformación dada por $AA^\dagger$ es la proyección ortogonal sobre $\mathrm{Im}(A)$.

Ejercicio 19 Dada $A \in \mathbb{R}^{n\times m}$ con $n \geq m$ y $b \in \mathbb{R}^n$. Sea $A = U\Sigma V^t$ la descomposición en valores singulares de A. Se quiere encontrar un $\hat{x} \in \mathbb{R}^m$ tal que:

$$\|A\hat{x} - b\|_2 = \min_{x\in\mathbb{R}^n} \|Ax - b\|_2 \tag{1}$$

a) Probar que si $\mathrm{rg}(A) = m$, entonces $\hat{x}$ es única y $\hat{x} = A^\dagger b$. (Sugerencia: recordar que $\mathrm{Nu}(A^t) = \mathrm{Im}(A)^\perp$).

b) Si $\mathrm{rg}(A) = r < m$, $\hat{x}$ no es única. En efecto, si consideramos $\tilde{x} \in \mathrm{Nu}(A)$, $A(\hat{x} + \tilde{x}) = A\hat{x}$. Probar que $\hat{x} = A^\dagger b$ es la única solución a (1) en $\mathrm{Nu}(A)^\perp$.

Algoritmos e implementación

Ejercicio 20 Implementar una función en Python que intercambie circularmente los valores de tres variables: la segunda tomará el valor de la primera, la tercera el valor de la segunda y la primera el de la tercera variable.

Ejercicio 21 Implementar una función en Python que, dado un entero positivo n, devuelva la matriz identidad de dimensión n.

Ejercicio 22 Programar una función en Python que calcule el determinante de una matriz cuadrada recursivamente.

Ejercicio 23 Escribir algoritmos para realizar suma, resta y multiplicación de matrices. Determinar la complejidad y expresarla en función de la dimensión de las matrices.

Ejercicio 24 Descomposición de Cholesky: Es sabido que si $A \in \mathbb{R}^{n\times n}$ es simétrica y definida positiva, entonces puede ser expresada como $A = LL^t$ donde L es triangular inferior.

a) Implementar una función que reciba como input una matriz A simétrica definida positiva y que devuelva L matriz triangular inferior tal que $A = LL^t$.

b) Implementar la sustitución *forward*: sean $L \in \mathbb{R}^{n\times n}$ triangular inferior y $b \in \mathbb{R}^n$, la función debe tomar como input a L y a b y devolver la solución de $Lx = b$.

c) Implementar la sustitución *backward*: sean $L \in \mathbb{R}^{n\times n}$ triangular superior y $b \in \mathbb{R}^n$, la función debe tomar como input a L y a b y devolver la solución de $Lx = b$.

d) Implementar una función que resuelva sistemas de n ecuaciones lineales con n incógnitas donde la matriz asociada es simétrica definida positiva.

e) Dados puntos distintos en el plano $\{(x_1, y_1), ...(x_n, y_n)\}$, el problema de Cuadrados Mínimos consiste en hallar el polinomio $p \in \mathcal{P}_k$ que mejor aproxime estos datos, en el sentido de minimizar la expresión: $\sum_{i=1}^{n}(y_i - p(x_i))^2$.

   Implementar una función que tome como input a k la cantidad de coeficientes del polinomio (i.e. $k = gr(P) + 1$), y a los datos $\{(x_1, y_1), ...(x_n, y_n)\}$ y que devuelva los coeficientes del polinomio p. Utilizar la descomposición de Cholesky.

f) Utilizando la librería de Python `matplotlib` y el archivo `datos.txt` (disponible en la página de la materia), graficar los datos y el polinomio obtenido aplicando la función del item anterior para distintos valores de k. ¿Qué ocurre a medida que aumentamos el valor de k?

Ejercicio 25 Descomposición QR: es sabido que cualquier $A \in \mathbb{R}^{n\times n}$ puede ser expresada como $A = QR$ donde Q es una matriz ortonormal y R es triangular superior.

a) Implementar una función que calcule la descomposición QR de una matriz aplicando de manera directa el método de ortonormalización de Gram-Schmidt.

b) Importar desde `numpy.random` el comando `rand` (con `from numpy.random import rand`) para generar matrices de manera aleatoria. Por ejemplo, si queremos generar una matriz de 4×4, lo hacemos de la siguiente manera:

$$\texttt{A} = \texttt{rand}((4,4))$$

   Importar desde `numpy.linalg` el comando `qr` (con `from numpy.linalg import qr`). Para distintas matrices, comparar los resultados de aplicar la función implementada en *a)* con el resultado de aplicar la función `qr` de Python.

c) Implementar una función que, dada una matriz $A \in \mathbb{R}^{n\times n}$ y un vector $b \in \mathbb{R}^n$, devuelva la solución de $Ax = b$.

d) Implementar una función que resuelva el problema de Cuadrados Mínimos utilizando la descomposición QR.

e) Utilizando la librería de Python `matplotlib` y el archivo `datos.txt` (disponible en la página de la materia), graficar los datos y el polinomio obtenido aplicando la función del item anterior para distintos valores de k.

Ejercicio 26 Descomposición en valores singulares (SVD): Implementar una función que reciba como input una matriz A y un entero positivo r y:

- Calcule la descomposición en valores singulares $A = U\Sigma V'$, utilizando el comando `svd` de `numpy.linalg` (se importa agregando la siguiente línea `from numpy.linalg import svd`).
- Corrija la matriz Σ poniendo $\sigma_i = 0$, $\forall i > r$.
- Devuelva $B = U\widetilde{\Sigma}V'$, siendo $\widetilde{\Sigma}$ la matriz que resulta de corregir Σ según el ítem anterior.

Este programa calcula la matriz B de rango r tal que la norma $\|A - B\|_2$ es mínima. Aplicar este programa a distintas matrices, con distintos valores de r.

Observación: el comando `svd` de Python ya devuelve a V traspuesta pero no devuelve Σ, sino que devuelve su diagonal. Para obtener Σ basta hacer lo siguiente:

```
U, s, Vt = svd(A)
sigma = np.diag(s)
```

Ejercicio 27 El programa del ejercicio anterior puede utilizarse para comprimir imágenes. Primero, importamos las librería `image` y `pyplot` de `matplotlib` de la siguiente manera:

```
import matplotlib.image as mimg
import matplotlib.pyplot as plt
```

Dada una imagen blanco y negro, el comando `mimg.imread` convierte la imagen en una matriz en la que cada casillero representa un píxel y su valor corresponde al color del píxel en escala de grises. El comando `plt.imshow` aplicado a esta matriz seguido del comando `plt.show()`, muestra la imagen, mientras que el comando `mimg.imsave` permite guardar la matriz como un archivo de imagen.

Implementar un algoritmo que reciba como input un archivo de imagen y un entero positivo r, realice la compresión como en el ejercicio anterior y guarde el resultado en otro archivo. Experimentar con el valor de r, observando cuán chico puede ser r en relación con el tamaño de la matriz si se desea conservar calidad en la imagen.

La misma experiencia puede repetirse con imágenes en color. En este caso, debe tenerse en cuenta que el comando `imread` devuelve un arreglo A de tamaño $m \times n \times 3$: la imagen es de $m \times n$ píxeles y $A[:,:,0]$, $A[:,:,1]$, $A[:,:,2]$ son las matrices correspondientes a su descomposición RGB (red-green-blue).