José Pablo Sánchez
174139

Examen Final

May 17, 2021

Dentro de su carpeta personal `Alumnos/<nombre de usuario>`, generar una nueva carpeta con el nombre de `ExamenFinal`.

- Dentro de esta carpeta introducir su código para la parte práctica y un documento que tenga las respuestas a la parte teórica.
- La duración del examen es de 2 horas y 45 minutos.
- Y el método de entrega es un Pull Request al repositorio del curso.

1 Teoría (40pts)

1.1 Gradiente Conjugado

- Demuestre que si los vectores no nulos $p_1, p_2, ..., p_l$ satisfacen que :

  $$p_i^T A p_j = 0, \forall i \neq j,$$

  y A es simétrica y positiva definida, entonces los vectores son linealmente independientes.
- Dado este resultado, ¿Por qué el gradiente conjugado converge en a lo más n iteraciones?.

1.2 BFGS

- Verifique que B_{k+1} y H_{k+1} son inversas una de la otra.

2 Code (60 puntos)

Los algoritmos que se piden correr a continuación tienen que ser propios (con excepción de los que de hecho hicimos en clase). Lo hecho en clase y modificaciones a los mismos es lo que se espera. No se espera que usen las funciones de optimización de scipy.

2.1 DFP (20 puntos)

Corre el Algoritmo BFGS, pero con la actualización DFP con la función cuadrados en dimensión 10 con punto inicial:

```
x0 = [(-1)**i*10 for i in range(10) ]
```

2.2 Gradiente Conjugado (40 puntos)

Con **TU** clave única como semilla, genera una matriz diagonal con 10^6 números aleatorios en la diagonal, después de esto, genera un vector del mismo tamaño. I.e.

```
import random
random.seed(108683) #  Cambien a su propia clave
Diag_A = [random.randint(1,1000) for x in range(1000000)]
b = [random.randint(1,1000) for x in range(1000000)]
```

Ahora resuelve el sistema lineal $Ax = b$ con el algoritmo de Gradiente Conjugado.

- **NOTAS**:
  - No intentes generar la matriz completa , ya que se requieren 7.28 TiB de memoria para almacenarlo.
  - La matriz tiene 10^{12} entradas, pero sólo 10^6 de ellas son distintas de cero :).
  - Sólo se puede usar una implementación propia de matriz rala.

1.1 Gradiente Conjugado

- Demuestre que si los vectores no nulos $p_1, p_2, ..., p_l$ satisfacen que :

$$p_i^T A p_j = 0, \forall i \neq j,$$

y A es simétrica y positiva definida, entonces los vectores son linealmente independientes.

P.d. $\{p_1, \dots, p_l\}$ son linealmente independientes donde $p_i^T A p_j = 0 \ \forall i \neq j$ y A def. positiva

i.e. si $\alpha_i \in \mathbb{R} \ \forall i \mid 1 \leq i \leq l$

$\Rightarrow \alpha_1 p_1 + \dots + \alpha_l p_l = 0 \iff \alpha_1 = \dots = \alpha_l = 0$

dem.

Supongamos que $\{p_1, \dots, p_l\}$ no es l.i.,

$\Rightarrow \alpha_1 p_1 + \dots + \alpha_l p_l = 0$ con al menos un $\alpha_k \neq 0$, $k \in (1, l$

luego,

$$A(\alpha_1 p_1 + \dots + \alpha_l p_l) = 0$$

$$\Rightarrow p_k^T A(\alpha_1 p_1 + \dots + \alpha_l p_l) = 0$$

$$\Rightarrow p_k^T A \cdot \alpha_1 p_1 + \dots + p_k^T A \alpha_l p_l = 0$$

$$\Rightarrow \alpha_1 (p_k^T A p_1) + \dots + \alpha_l (p_k^T A p_l) = 0$$

pero, por hipótesis, $p_i^T A p_j = 0 \ \forall i \neq j$

$$\Rightarrow 0 + \dots + \alpha_k (p_k^T A p_k) + \dots + 0 = 0$$

$$\Rightarrow \alpha_k (p_k^T A p_k) = 0$$

pero $\alpha_k \neq 0$

$$\Rightarrow (p_k^T A p_k) = 0$$

pero A es definida positiva, luego

$$p_k^T A p_k > 0 \quad !$$

$\therefore \{p_1, \dots, p_l\}$ son linealmente independientes

- Dado este resultado, ¿Por qué el gradiente conjugado converge en a lo más n iteraciones?.

Recordemos que la iteración entre soluciones está dada por

$$x_{k+1} = x_k + \alpha_k p_k \qquad \text{con } \alpha_k = \frac{-r_k^T \cdot p_k}{p_k^T \cdot A \cdot p_k}$$

Una vez que probamos que $\{p_1, \ldots, p_n\}$ son l. i.,

span. $\{p_1, \ldots, p_n\} = \mathbb{R}^n$

y $x_* - x_0 \in \mathbb{R}^n$ donde x_* es solución del problema.

$\Rightarrow x_* - x_0 \in$ span. $\{p_1, \ldots, p_n\}$

$\Rightarrow \exists \gamma_1, \ldots, \gamma_n \in \mathbb{R}$ tal que $x_* - x_0 = \gamma_1 p_1 + \ldots + \gamma_n p_n$

$$\Rightarrow \gamma_i = \frac{p_i^T A (x_* - x_0)}{p_i^T A p_i} \qquad \forall i = 1, \ldots, n$$

por otro lado,

$$x_k = x_0 + \alpha_0 p_0 + \ldots + \alpha_{k-1} p_{k-1}$$

$$\Leftrightarrow x_k - x_0 = \alpha_0 p_0 + \ldots + \alpha_{k-1} p_{k-1}$$

$$\Leftrightarrow p_k^T \cdot A (x_k - x_0) = 0$$

entonces

$$p_k^T \cdot A (x_* - x_0) = p_k^T A (x_* - x_k) = p_k^T \cdot r_k$$

$\therefore$ $\alpha_k = \gamma_k$, lo cual implica que el algoritmo termina en a lo más n iteraciones pues $x_n = x_*$ //

1.2 BFGS

- Verifique que B_{k+1} y H_{k+1} son inversas una de la otra.

Veamos que

$$H_{K+1} = (I - p_K s_K y_K^T)\cdot H_K \cdot (I - p_K y_K s_K^T) + p_K s_K s_K^T$$

$$B_{K+1} = B_K - \frac{B_K s_K s_K^T B_K}{s_K^T \cdot B_K \cdot s_K} + \frac{y_K \cdot y_K^T}{y_K^T \cdot s_K}$$

P.d. $B_{K+1}^{-1} = H_{K+1}$, i.e. $B_{K+1}\cdot H_{K+1} = I$

dem.

Eliminemos el subíndice K de p, s, y, y veamos que

$$B_{K+1}\cdot H_{K+1} = B_{K+1}\left((I - psy^T)\cdot H \cdot (I - pys^T) + pss^T\right)$$

$$= \left(B_{K+1} - pyy^T\right)\cdot H_K \cdot (I - pys^T) + pys^T)$$

$$= \left(B_K - \frac{B_K s s^T B_K}{s^T \cdot B_K \cdot s}\right)\cdot H_K \cdot (I - pys^T) + pys^T$$

$$= \left(I - \frac{B_K ss^T}{s^T B_K s}\right)(I - pys^T) + pys^T$$

$$= (I - pys^T) - \left(\frac{B_K ss^T}{s^T B s}\right)(I - pys^T) + pys^T$$

$$= I - \left(\frac{B_K \cdot s \cdot s^T}{s^T B_K s}\right)(I - pys^T)$$

pero sabemos que

$$\frac{B_K ss^T}{s^T B_K s}\cdot pys^T = \frac{1}{s^T y}\cdot \frac{B_K s(s^T y)\cdot s^T}{s^T B_K s} = \frac{B_K s \cdot s^T}{s^T B_K s}$$

$$\Rightarrow \left(\frac{B_K \cdot s \cdot s^T}{s^T B_K s}\right)(I - pys^T) = \frac{B_K \cdot s \cdot s^T}{s^T B_K s} - \frac{B_K ss^T}{s^T B_K s}\cdot pys^T$$

$$= \frac{B_K \cdot s \cdot s^T}{s^T B_K s} - \frac{B_K s \cdot s^T}{s^T B_K s}$$

$$= 0$$

$$\therefore B_{K+1}\cdot H_{K+1} = I$$