Comenzado el	sábado, 8 de agosto de 2020, 10:21
Estado	Finalizado
Finalizado en	sábado, 8 de agosto de 2020, 15:50
Tiempo empleado	5 horas 28 minutos
Calificación	**93,33** de 120,00 (**78**%)

Pregunta 1

Correcta

Puntúa 40,00 sobre 40,00

Considere el siguiente conjunto de datos $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ que se quieren ajustar con la siguiente función $f(x, \boldsymbol{\beta}) = \beta_1 + \sin(\beta_2\, x) + \cos(\beta_3\, x)$, donde $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)$. Para esto requerimos minimizar la función $S(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - f(x_i, \boldsymbol{\beta}))^4$. De similar forma a lo realizado en el certamen anterior, debemos encontrar un mínimo, para lo cual debemos obtener las derivadas parciales con respecto a los parámetros, esto es:

$$\begin{aligned}
\frac{\partial S}{\partial \beta_1} &= \sum_{i=1}^{n} -4(y_i - \beta_1 - \sin(\beta_2\, x_i) - \cos(\beta_3\, x_i))^3 = 0 \\
\frac{\partial S}{\partial \beta_2} &= \sum_{i=1}^{n} -4x_i \cos(\beta_2\, x_i)(y_i - \beta_1 - \sin(\beta_2\, x_i) - \cos(\beta_3\, x_i))^3 = 0 \\
\frac{\partial S}{\partial \beta_3} &= \sum_{i=1}^{n} 4x_i \sin(\beta_3\, x_i)(y_i - \beta_1 - \sin(\beta_2\, x_i) - \cos(\beta_3\, x_i))^3 = 0
\end{aligned}$$

Podemos llevar la expresión anterior al problema $\mathbf{F}(\boldsymbol{\beta}) = \mathbf{0}$, donde $\mathbf{F}(\boldsymbol{\beta})$ se define como:

$$\mathbf{F}(\boldsymbol{\beta}) = \begin{bmatrix} \displaystyle\sum_{i=1}^{n} (y_i - \beta_1 - \sin(\beta_2\, x_i) - \cos(\beta_3\, x_i))^3 \\ \displaystyle\sum_{i=1}^{n} x_i \cos(\beta_2\, x_i)\,(y_i - \beta_1 - \sin(\beta_2\, x_i) - \cos(\beta_3\, x_i))^3 \\ \displaystyle\sum_{i=1}^{n} x_i \sin(\beta_3\, x_i)\,(y_i - \beta_1 - \sin(\beta_2\, x_i) - \cos(\beta_3\, x_i))^3 \end{bmatrix}. \qquad (1)$$

Para obtener el $\boldsymbol{\beta}$ que minimiza S, utilizaremos una variante del método de Newton en alta dimensión, el cual es conocido como el método de **Levenberg-Marquardt.** En resumen, este método es muy similar al método de Newton en $\mathbb{R}^n$ pero el sistema de ecuaciones que se resuelve en cada iteración es el siguiente:

$$\begin{aligned}
\left(J(\boldsymbol{\beta}_i)^T J(\boldsymbol{\beta}_i) + \lambda\, I\right) \Delta\boldsymbol{\beta}_i &= -J(\boldsymbol{\beta}_i)^T \mathbf{F}(\boldsymbol{\beta}_i), \\
\boldsymbol{\beta}_{i+1} &= \boldsymbol{\beta}_i + \Delta\boldsymbol{\beta}_i
\end{aligned} \qquad (2)$$

donde $J(\boldsymbol{\beta}_i)$ es la matriz Jacobiana de $\mathbf{F}$ evaluada en $\boldsymbol{\beta}_i$. El objetivo del parámetro λ es reducir el número de condición asociado a la matriz $J(\boldsymbol{\beta}_i)^T\, J(\boldsymbol{\beta}_i)$, además de mejorar la convergencia del método. Sin embargo, al utilizar un valor de λ muy grande, la solución obtenida es una perturbación de la solución límite, es decir, cuando λ tiende a 0. Para evitar calcular la matriz Jacobiana $J(\boldsymbol{\beta})$ utilizaremos una aproximación. Considere la siguiente linealización de la función $\mathbf{F}$:

$$\mathbf{F}(\boldsymbol{\beta} + \varepsilon\, \mathbf{v}) = \mathbf{F}(\boldsymbol{\beta}) + J(\boldsymbol{\beta})(\varepsilon\, \mathbf{v}) + \text{Términos de orden } \varepsilon^2 + \ldots.$$

donde $\mathbf{v}$ es un vector que será definido más adelante y $0 < \varepsilon \ll 1$ un escalar. Basado en la expansión anterior, podemos despejar el producto de la matriz Jacobiana por el vector $\mathbf{v}$ de la siguiente forma:

$$J(\boldsymbol{\beta})\, \mathbf{v} \approx \frac{\mathbf{F}(\boldsymbol{\beta} + \varepsilon\, \mathbf{v}) - \mathbf{F}(\boldsymbol{\beta})}{\varepsilon} \qquad (3)$$

Con esta aproximación, podemos notar que al definir $\mathbf{v} = \mathbf{e}_1 = [1, 0, 0]^T$, se obtiene una aproximación de la primera columna de $J(\boldsymbol{\beta})$, con $\mathbf{v} = \mathbf{e}_2 = [0, 1, 0]^T$ la segunda columna y $\mathbf{v} = \mathbf{e}_3 = [0, 0, 1]^T$ la tercera columna. Dicho de otra manera, tenemos:

$$J(\boldsymbol{\beta}) \approx \left[\begin{array}{c|c|c} \frac{\mathbf{F}(\boldsymbol{\beta} + \varepsilon\, \mathbf{e}_1) - \mathbf{F}(\boldsymbol{\beta})}{\varepsilon} & \frac{\mathbf{F}(\boldsymbol{\beta} + \varepsilon\, \mathbf{e}_2) - \mathbf{F}(\boldsymbol{\beta})}{\varepsilon} & \frac{\mathbf{F}(\boldsymbol{\beta} + \varepsilon\, \mathbf{e}_3) - \mathbf{F}(\boldsymbol{\beta})}{\varepsilon} \end{array}\right]$$

La gran ventaja de la aproximación anterior es que uno puede obtener numéricamente una estimación razonable de la matriz Jacobiana omitiendo el manejo algebraico de obtener el gradiente de cada termino en la ecuación (1).

Considerando el cálculo de la matriz Jacobiana aproximada por la ecuación (3) con $\varepsilon = 10^{-10}$ y la iteración de Levenberg-Marquardt definida en (2), determine el valor del parámetro β_1 luego de 3 iteraciones del método de Levenberg-Marquardt utilizando $\lambda = 1$ y para resolver el sistema de ecuaciones en (2), utilice al algoritmo de SOR(1.1) con 3 iteraciones, *initial guess* $\boldsymbol{\beta}_0 = [1, 1, 1]^T$ y 10^{-10} como tolerancia en SOR(1.1).

Los datos para ajustar el modelo se encuentran en https://github.com/sct-utfsm/INF-285/tree/master/cop-4/data/lm/1.npy. En la primera columna encontrará los valores de x_i y la segunda columna los valores de y_i.

Debe entregar sus resultados con 5 decimales sin redondear. Por ejemplo si su resultado es 2365.1345871237 debe completar con 2365.13458.

Preguntas:

1. **[6 puntos]** ¿Cuál es el valor de la segunda componente de $\mathbf{F}(\boldsymbol{\beta}_0)$? Respuesta: -1322.
2. **[10 puntos]** ¿Cuál es la norma de Frobenius de la aproximación del Jacobiano en $\boldsymbol{\beta}_0$, es decir, $\|J(\boldsymbol{\beta}_0)\|_F$? Respuesta: 16783.
3. **[24 puntos]** ¿Cuál es el valor del parámetro β_1 luego de ajustar el modelo? Respuesta: 1.0739

✓

Pregunta 2

Correcta

Puntúa 26,67 sobre 40,00

Consideremos el problema de interpolar una colección de puntos (x_i, y_i) con $i \in 1:N$ utilizando una spline cúbica. Para ello, se define la spline de la siguiente forma:

$$S(x) = \begin{cases} S_1(x), & x \in [x_1, x_2], \\ S_2(x), & x \in [x_2, x_3], \\ \vdots & \\ S_{N-1}(x), & x \in [x_{i-1}, x_i], \end{cases}$$

donde cada $S_i(x),\ i = 1:N-1$, tiene la siguiente estructura:

$$S_i(x) = y_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3.$$

Sabemos que para encontrar una única solución para los $3N-3$ coeficientes de $S(x)$ debemos resolver un sistema de ecuaciones lineales que posea $3N-3$ ecuaciones. Estas ecuaciones se obtienen a partir de las siguientes propiedades:

- Propiedad 1 - Continuidad:

$$S_i(x_i) = y_i \text{ y } S_i(x_{i+1}) = y_{i+1} \ \text{ donde } i = 1:N-1.$$

- Propiedad 2 - Diferenciabilidad:

$$S'_{i-1}(x_i) = S'_i(x_i) \ \text{ donde } i = 2:N-1.$$

- Propiedad 3 - Continuidad en la segunda derivada:

$$S''_{i-1}(x_i) = S''_i(x_i) \ \text{ donde } i = 2:N-1.$$

Estas 3 propiedades nos entregan $3N-5$ ecuaciones por lo que al considerar condiciones de borde se obtienen las dos ecuaciones lineales adicionales faltantes. Sin embargo, si además queremos considerar continuidad en la 3ra derivada, es decir $S'''_{i-1}(x_i) = S'''_i(x_i)$ para $i = 2:N-1$ obtendríamos un sistema de ecuaciones lineales sobre-determinado, es decir, más ecuaciones que incógnitas. En este caso, considerando que es muy importante la Propiedad 1, continuidad, podemos definir el siguiente problema de minimización con restricciones:

$$\min \sum_{i=1}^{n-1} \left(S'_{i-1}(x_i) - S'_i(x_i)\right)^2 + \left(S''_{i-1}(x_i) - S''_i(x_i)\right)^2 + \left(S'''_{i-1}(x_i) - S'''_i(x_i)\right)^2,$$

sujeto a $S_i(x_{i+1}) = y_{i+1}$ para $i = 2:N-1$.

Una alternativa conveniente pero no exacta a resolver un problema de minimización con restricciones, es resolver un problema de mínimos cuadrados dándole un peso distinto a las distintas ecuaciones.
Esto significa resolver la siguiente minimización:

$$\min \sum_{i=1}^{n-1} w_1^2 \left(S_i(x_{i+1}) - y_{i+1}\right)^2 \sum_{i=2}^{n-1} w_2^2 \left(S'_{i-1}(x_i) - S'_i(x_i)\right)^2 + w_3^2 \left(S''_{i-1}(x_i) - S''_i(x_i)\right)^2 + w_4^2 \left(S'''_{i-1}(x_i) - S'''_i(x_i)\right)^2,$$

el cual se traduce en el siguiente sistema de ecuaciones lineales sobre-determinado:

$$W\,A\,\mathbf{c} = W\,\mathbf{b},$$

donde $W = \text{diag}(w_1, w_1, \ldots, w_1, w_2\ \ldots, w_4) \in \mathbb{R}^{(4\,N-7\,\times\,4\,N-7)}$ es la matriz diagonal y cuadrada con los pesos asociados a cada ecuación, $A \in \mathbb{R}^{(4\,N-7\,\times\,3\,N-3)}$ es la matriz tradicional de splines cúbicas pero agregandole en las últimas filas las ecuaciones asociadas a la continuidad en la tercera derivada, $\mathbf{c} \in \mathbb{R}^{3\,N-3}$ y $b \in \mathbb{R}^{4\,N-7}$. Recuerde que la matriz A es rectangular porque aumenta la cantidad de filas y la cantidad de columnas permanece invariante.
Con este procedimiento hemos adaptado el problema anterior de minimización con restricciones a una de mínimos cuadrados sin restricciones y podemos utilizar lo que ya conocemos!, lo único que necesitamos hacer es definir un mayor peso a las ecuaciones que nos interesa ajustar de mejor manera. Notar que en este caso nuestra matriz a resolver es el producto $W\,A$ y nuestro lado derecho es $W\,\mathbf{b}$, es decir, debemos considerarlos de forma conjunta en ambos casos.

El problema que usted deberá resolver será el de encontrar una casi-spline cubica (ya que no será exactamente una spline cúbica, solo será una función definida por intervalos) que minimice el error cuadrático ponderado al pasar por los puntos almacenado en Dataset1.npy (link) que provienen de la función $f(x) = \sin(x)$ y considere adicionalmente la continuidad de la tercera derivada.

Para encontrar los coeficientes de $S(x)$ se deberá utilizar mínimos cuadrados utilizando Ecuaciones normales.
Para esto se deberá construir la matriz A con las $4\,N - 7$ ecuaciones que provienen de las propiedades descritas anteriormente.
Los pesos a utilizar se definen de la siguiente forma:

- $w_1 = 1,$
- $w_2 = 0.1,$
- $w_3 = 1,$
- $w_4 = 1.$

Preguntas

Encuentre la norma de Frobenius de A, el valor del coeficiente b_1 y la norma infinita del error de interpolación ($\max_i |f(x_i) - S(x_i)|$) utilizando 1000 puntos equiespaciados en el intervalo $[0, 2\pi]$ de evaluación. (**Para cada una de sus respuestas entregue solo los 5 decimales después de la coma, es decir si su resultado es 123.567864 su respuesta debe ser 56786)**

Consideración

El archivo Dataset1.npy contiene dos arreglos. La primera columna contiene los puntos x y la segunda columna contiene los puntos y.

[40/3 puntos] Norma de Frobenius = 92462

[40/3 puntos] Coeficiente b_1 = 33800

[40/3 puntos] Error de interpolación = 05052

✓

Pregunta 3

Correcta

Puntúa 26,67 sobre 40,00

Considere el siguiente conjunto de pares de matrices en $\mathbb{R}^{n\times n}$: $\{(X_1, Y_1), (X_2, Y_2)\}$, y el siguiente algoritmo del tipo Interpolación de Lagrange, ahora en $\mathbb{R}^{n\times n}$, es decir $P(X) : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$:

$$P(X) = Y_1 \left(X - X_2\right) \left(X_1 - X_2\right)^{-1} + Y_2 \left(X - X_1\right) \left(X_2 - X_1\right)^{-1}$$

Advertencia: ***No está permitido el uso de la matriz inversa, es decir, cuando aparezca la inversa de una matriz usted debe buscar la forma de modificar el problema y convertirlo a un sistema de ecuaciones lineales***.

La data $(X_1, Y_1, X_2, Y_2, \widetilde{Y})$ que le corresponde utilizar está en el repositorio https://github.com/sct-utfsm/INF-285/tree/master/cop-4/data/IML/ con los archivos X_1 =X1-0.npy, Y_1 =Y1-0.npy, X_2 =X2-0.npy, Y_2 =Y2-0.npy y $\widetilde{Y}$ =Yt-0.npy, respectivamente.

Preguntas:

1. **[5 puntos]** Obtenga el valor de la norma matricial de Frobenious de la matriz W_1 de la siguiente ecuación matricial: $X_1\, W_1 = Y_1$. Entregue en la casilla los primeros 5 decimales, si el número es 45.173926 usted debe ingresar 17392.

15594

2. **[5 puntos]** Obtenga el valor de la norma matricial de Frobenious de la matriz W_2 de la siguiente ecuación matricial: $W_2\, X_2 = Y_2$. Entregue en la casilla los primeros 5 decimales. Hint 1: Just recall that when applying the transpose operator to $A\, B = C$ we obtain $B^T\, A^T = C^T$.

67795

3. **[15 puntos]** Implemente la evaluación del polinomio matricial y obtenga el valor de la norma matricial de Frobenious $P(\underline{0})$, donde $\underline{0}$ es la matriz nula (i.e. la matriz de 0 de dimensión $n \times n$). Usted debe entregar el valor de la norma matricial de Frobenious $||P(\underline{0})||_F$. Entregue en la casilla los primeros 5 decimales.

16699

4. **[15 puntos]** Obtenga el valor de la norma matricial de Frobenious de la matriz X de la siguiente ecuación matricial $\widetilde{Y} = P(X)$. Entregue en la casilla los primeros 5 decimales.

85427

✓

Avenida España 1680, Valparaíso · +56 32 2654000 · dgc@usm.cl

Sitio web administrado por la Dirección General de Comunicaciones.