IMT2220, Cálculo para ciencia de datos, 2023-2
Tarea 2

Fecha entrega: 22 de Septiembre de 2023

Instrucciones

Pueden discutir los problemas con sus compañeros. Sin embargo, sus entregas, tanto la parte escrita como los códigos, deben ser individuales. **No está permitido** copiar las respuestas de alguien más ni dejar que otros copien sus respuestas. El hacerlo se reflejará con la nota mínima en la evaluación (1.0).
Su entrega debe estar compuesta por un único archivo .pdf **escrito en** LaTeX que incluya todas las respuestas y todos los gráficos que se están pidiendo, junto con un archivo .zip que incluya todos los códigos que produjeron y utilizaron durante esta tarea. Estos dos archivos deben ser subidos en Canvas. Si hay más de una línea en un gráfico, utilice distintos estilos de línea o distintos colores para diferenciarlas e incluya leyendas. No se olvide de nombrar todos sus ejes y líneas en las leyendas. **Cada gráfico debe ser comentado** (un gráfico sin un análisis del mismo no es una respuesta apropiada).

1. (12 puntos) Determine si los siguientes límites existen. Si existen, diga cuanto vale, en caso contrario de un contraejemplo de por qué no existe.

   (a) $\lim\limits_{(x,y)\to(0,0)} \dfrac{x^2-y^2}{x^2+y^2}$.

   (b) $\lim\limits_{(x,y)\to(0,0)} \dfrac{xy}{x^2+y^2+2}$.

   (c) $\lim\limits_{(x,y)\to(0,0)} \dfrac{\sin(2x)-2x+y}{x^3+y}$.

   (d) $\lim\limits_{(x,y,z)\to(0,0,0)} \dfrac{2x^2y\cos(z)}{x^2+y^2+z^2}$.

   Hint: Puede se conveniente usar coordenadas esféricas, que son como el análogo de coordenadas polares para 3 dimensiones:

   $$\begin{aligned} x &= r\cos(\theta)\sin(\varphi) \\ y &= r\sin(\theta)\sin(\varphi) \\ z &= r\cos(\varphi) \end{aligned}$$

   que son tales que $x^2+y^2+z^2=r^2$, $r\geq 0$, $\theta\in[0,2\pi)$, $\varphi\in[0,\pi)$.

2. (12 puntos) Calcule el gradiente de las siguientes funciones $f:\mathbb{R}^n\mapsto\mathbb{R}$.

(a) $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$

(b) $f(\boldsymbol{x}) = \boldsymbol{x}^\top \mathbf{A}\boldsymbol{x}$, para $\mathbf{A} \in \mathbb{R}^{n\times n}$ matriz dada.

(c) $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_2^2 \ln(1 + \|\boldsymbol{x}\|_2^2)$

(d) $f(\boldsymbol{x}) = \dfrac{\prod_{i=1}^n x_i}{1 + \|\boldsymbol{x}\|_2^2}$

Hint: Basta con calcular la derivada parcial $\partial f/\partial x_j$ para $j \in \{1, \ldots, n\}$ arbitrario y luego señalar que:

$$(\nabla f)_j = \frac{\partial f}{\partial x_j}$$

Además, puede serle de utilidad saber que:

$$\frac{\partial x_i}{\partial x_j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

3. (10 puntos) Considere $\boldsymbol{f} : D \subset \mathbb{R}^n \mapsto \mathbb{R}^m$ una función tal que:

$$\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{y})\| \leq L\,\|\boldsymbol{x} - \boldsymbol{y}\|^\alpha$$

para $L, \alpha > 0$ constantes. Demuestre que $\boldsymbol{f}$ es continua. **Observación:** Si $\boldsymbol{f}$ satisface lo anterior, se dice que $\boldsymbol{f}$ es Hölder continua, y para el caso $\alpha = 1$ se dice que $\boldsymbol{f}$ es Lipschitz continua.

4. En esta pregunta estudiaremos el algoritmo de descenso de gradiente para optimizar una función objetivo $f : \mathbb{R}^n \to \mathbb{R}$. En particular, considere el problema de optimización

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$$

donde f es una función diferenciable. El algoritmo de descenso de gradiente para resolver este problema es como sigue:

Algorithm 1: Descenso de gradiente

Data: Initial guess $\boldsymbol{x}_0$ y N número de iteraciones
Result: $\boldsymbol{x}_{N+1} \in \mathbb{R}^n$ estimación de un mínimo local de f.
for $k = 0, \ldots, N$ **do**
  $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k - \eta_k \nabla f(\boldsymbol{x}_k)$
end

donde η_k son tamaños de pasos pequeños ($\eta_k \geq 0$). Note que se puede escribir este ciclo `for` como un `while`, y saliendo de él cuando se alcance un umbral de error tal como muestra el Algoritmo 2.

Algorithm 2: Descenso de gradiente (versión ciclo `while`)

Data: Initial guess $\boldsymbol{x}_0$ y $\varepsilon > 0$ tolerancia.
Result: $\boldsymbol{x} \in \mathbb{R}^n$ estimación de un mínimo local de f.
$\delta x = 1$
$k = 0$
while $\delta x > \varepsilon$ **do**
  $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k - \eta_k \nabla f(\boldsymbol{x}_k)$
  $\delta x = \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2$
  $k = k + 1$
end

A partir de ahora, considere que $n = 2$.

(a) (5 puntos) Considere la función $f(x, y) = x \sin(\pi y) + x^2 + y^2$. Realice sus gráficos 3D y de contorno para $(x, y) \in [-1, 1] \times [-1, 1]$. Para este último muestre al menos 20 niveles. A primera vista, ¿cerca de qué puntos podemos encontrar mínimos locales?

(b) (7 puntos) Programe el método de descenso de gradiente (Algoritmo 2) para la función f presentada anteriormente, con criterio de parada $\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2 \leq 10^{-12}$. Corra este algoritmo con puntos iniciales $\boldsymbol{x}_0 = (1/2, -3/4)^\top$ y $\boldsymbol{x}_0 = (3/4, 0)^\top$ para tamaños de pasos $\eta_k = 1/k$ para $k \geq 1$. Reporte el número de pasos requeridos para cada punto inicial en alcanzar un mínimo local y grafique los puntos resultantes en el gráfico de contorno.

(c) (4 puntos) Considere la siguiente modificación al algoritmo presentado, para $\alpha \in [0, 1]$:

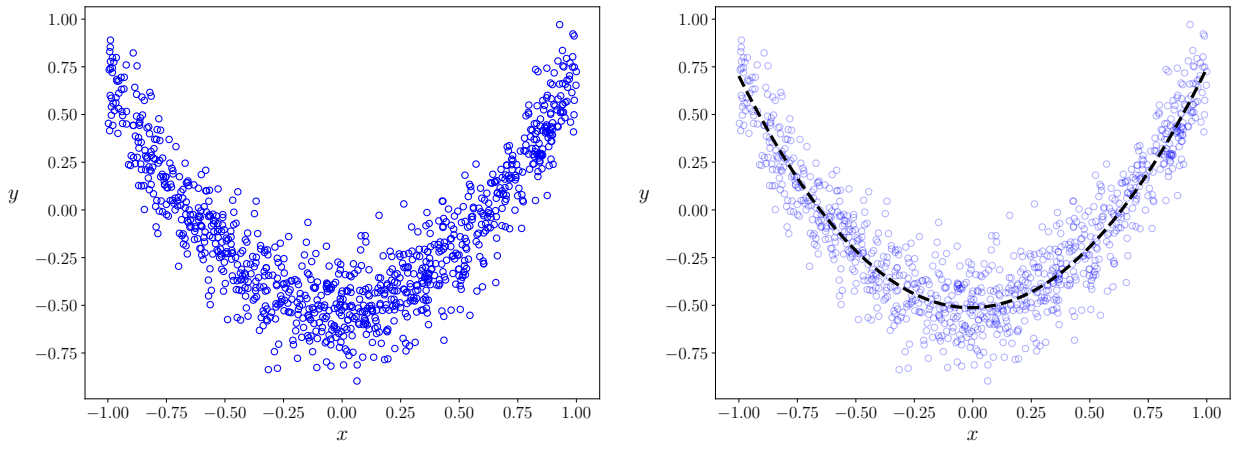
$$\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k - \eta_k \nabla f(\boldsymbol{x}_k) + \alpha \Delta \boldsymbol{x}_k, \quad \Delta \boldsymbol{x}_k \coloneqq \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$$

desde la segunda iteración en adelante. Programe el algoritmo con esta modificación y reporte nuevamente todo lo de la pregunta anterior para un valor $\alpha = 0.5$ y el mismo η_k utilizado anteriormente. ¿Cómo se comparan estos resultados con los obtenidos para el algoritmo clásico?

(d) (10 puntos) Considere ahora el problema donde queremos aprender que fenómeno vemos a través de un set de datos $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1,\dots,N} \subset \mathbb{R}^2$. En este caso, queremos aprender el comportamiento de los datos mediante una función de pérdida $\boldsymbol{L}$ que representa cuan veraz es nuestro modelo. Mientras más pequeña la función de pérdida, mejor es el modelo propuesto. Es decir, queremos resolver:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^3} \boldsymbol{L}(\boldsymbol{\theta}; \boldsymbol{X}) = \min_{a,b,c} \sum_{i=1}^{N} L(\boldsymbol{\theta}; \boldsymbol{x}_i)$$

Es decir, dato el i-ésimo dato $\boldsymbol{x}_i = (x_i, y_i) \subset \mathbb{R}^2$, tenemos asociada una función de pérdida $L(\boldsymbol{\theta}, \boldsymbol{x}_i)$. En particular, queremos que esta función de pérdida haga que busquemos la mejor parábola que se ajuste a los datos. La Figura 1 muestra el set de datos y un posible ajuste de ellos.



(a) Plot de `data_HW2.csv`.

(b) Un posible ajuste cuadrático a los datos anteriores.

Figura 1: Datos y ajuste cuadrático de datos `data_HW2.csv`.

Para obtener resultados de este estilo, usamos la función

$$L(\boldsymbol{\theta}; \boldsymbol{x}) = L(a, b, c; x, y) = \left(ax^2 + bx + c - y\right)^2$$

Y para encontrar dichas constantes (a, b, c), debemos aprender de los datos. Considere el Algoritmo 3. Este se usa para buscar un vector de parámetros $\boldsymbol{\theta} = (a, b, c)$ que minimice la función de pérdida $L(\boldsymbol{\theta}; \boldsymbol{x})$, donde $\boldsymbol{x} = (x, y)$ es un dato. Por ejemplo, para el dato $\boldsymbol{x}_1$ tenemos que la función de pérdida corresponde a:

$$L(\boldsymbol{\theta}; \boldsymbol{x}_1) = (ax_1^2 + bx_1 + c - y_1)^2$$

El Algoritmo 3 se ocupa para aprender de dichos datos y, así, minimizar esta instancia particular de $\boldsymbol{L}$.

Algorithm 3: Descenso de gradiente estocástico

Data: Initial guess $\boldsymbol{\theta}_0$ y datos $\{\boldsymbol{x}_i\}_{i=1,\dots,N}$
Result: Vector de parámetros estimados $\boldsymbol{\theta}_{N+1} = (a_{N+1}, b_{N+1}, c_{N+1})$
$\boldsymbol{\theta}_1 \leftarrow \boldsymbol{\theta}_0$
for $k = 1, \dots, N$ **do**
  $\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k - \eta_k \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_k; \boldsymbol{x}_k)$
end

Para el set de datos `data_HW2.csv`:

i. Programe este algoritmo para un paso no constante (por ejemplo, $\eta_k = 1/k, k \geq 1$) y ejecutelo para este set de datos en 100 ordenes distintos. Para cada ejecución, almacene la historia de la función $\boldsymbol{L}(\boldsymbol{\theta})$ definida por:

$$\boldsymbol{L}(\boldsymbol{\theta}) = \sum_{i=1}^{N} L(\boldsymbol{\theta}; \boldsymbol{x}_i)$$

y retornela junto con el vector de parámetros $\boldsymbol{\theta} = (a, b, c)$.

ii. Encuentre los 5 vectores de parámetros con menor valor de $\boldsymbol{L}$ final. Para estos, grafique la historia de $\boldsymbol{L}$ en un gráfico log-log (`matplotlib.pyplot.loglog` en python) y comente sobre si todas son iguales y por qué no lo son. Grafique, como en la Figura 1b, las curvas obtenidas sobre los datos para los parámetros $\boldsymbol{\theta}_{N+1}$ correspondientes. ¿Son todos estos ajustes iguales? Finalmente, reporte los promedios de estos estimadores, es decir

$$\boldsymbol{\theta}_{\text{avg}} = \frac{1}{5} \sum_{i=1}^{5} \widehat{\boldsymbol{\theta}}_i$$

donde $\hat{\boldsymbol{\theta}}_i$ son los mejores 5 estimadores obtenidos del Algoritmo 3.