Tarea 0: Fundamentos

Inteligencia Artificial
Lic. en Ciencias de la Computación
Universidad de Sonora
Semestre 2023-1

Esta tarea consiste de una parte escrita y una parte de programación. Todas tus respuestas deben ser escritas en TEX (o algún derivado), deben aparecen en el mismo orden que las preguntas y estar correctamente etiquetadas. Tus programas deben estar escritos en el lenguaje de programación Python.

Debes cargar tu trabajo al grupo de Microsoft Teams del curso como un archivo comprimido llamado `fundamentos` que contenga tus respuestas en un archivo en formato PDF llamado `fundamentos` y tu código en un archivo de Python llamado `fundamentos`.

1. Optimización y probabilidad.

a. Sean $x_1, \ldots, x_n$ reales que representan posiciones en una recta. Sean $w_1, \ldots, w_n$ reales positivos que representan la importancia de cada una de estas posiciones. Considera la función cuadrática

$$f(\theta) = \sum_{i=1}^{n} w_i(\theta - x_i)^2.$$

donde θ es un escalar. ¿Qué valor de θ minimiza $f(\theta)$? Muestra que el óptimo que encuentres es un mínimo. ¿Qué problemas pueden surgir si algunas de las w_i son negativas?

> *Nota:* Puedes pensar sobre este problema como intentar encontrar el punto θ que no está demasiado lejos de las x_i.
>
> **Expectativa:** Una expresión para el valor de θ que minimiza $f(\theta)$ y cómo la obtuviste. Un breve cálculo o argumento que muestre que es un mínimo. Una o dos oraciones describiendo el problema que puede surgir si alguna de las w_i es negativa.

b. Considera las siguientes igualdades.

$$\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d,$$

$$f(\boldsymbol{x}) = \max_{s \in [-1,1]} \sum_{i=1}^{d} s x_i,$$

$$g(\boldsymbol{x}) = \sum_{i=1}^{d} \max_{s_i \in [-1,1]} s_i x_i.$$

¿Cuál de $f(\boldsymbol{x}) \leq g(\boldsymbol{x})$, $f(\boldsymbol{x}) = g(\boldsymbol{x})$ o $f(\boldsymbol{x}) \geq g(\boldsymbol{x})$ es verdadera para toda $\boldsymbol{x}$? Demuéstralo.

Nota: Puede ser de ayuda replantear las expresiones para que estén maximizando la misma cantidad sobre conjuntos de diferentes tamaños.

Expectativa: Una demostración breve de tres a cinco oraciones. Debes utilizar notación matemática en tu demostración, pero también puedes argumentar con palabras.

c. Supongamos que lanzas repetidamente un dado honesto de seis caras hasta obtener un 1 o un 2 (e inmediatamente después detenerte). Cada vez que obtienes un 3 pierdes a puntos, cada vez que obtienes un 6 ganas b puntos. No ganas ni pierdes puntos si obtienes un 4 o 5. ¿Cuál es la cantidad esperada de puntos (como función de a y b) que tienes al final?

Nota: Puede ser de ayuda definir una recurrencia. Si defines V como el número de puntos que obtienes de jugar al juego. ¿Qué pasa si lanzas un 3? Pierdes a puntos y juegas de nuevo. ¿Qué pasa en los otros casos? ¿Puedes escribir esto como una recurrencia?

Expectativa: Una recurrencia para representar el problema y la expresión resultante para resolver la recurrencia (no más de un par de líneas).

d. Supongamos que la probabilidad de que una moneda caiga en águila es $0 < p < 1$, y que lanzamos esta moneda seis veces obteniendo $\{S, A, A, A, S, A\}$. Sabemos que la probabilidad de obtener esta secuencia es

$$L(p) = (1-p)ppp(1-p)p = p^4(1-p)^2.$$

¿Cuál valor de p maximiza $L(p)$? Muestra que este valor de p maximiza $L(p)$. ¿Cuál es una interpretación intuitiva de este valor de p?

Nota: Considera tomar la derivada de $\log L(p)$. También puedes tomar directamente la derivada de $L(p)$. Puedes verificar que el valor de p que maximiza $\log L(p)$ también debe maximizar $L(p)$.

Expectativa: El valor de p que maximiza $L(p)$ y los cálculos que realizaste para resolverlo. Debes mostrar que es el máximo y escribir una oración donde planteas una interpretación intuitiva del valor de p.

e. Supongamos que A y B son dos eventos tales que $P(A|B) = P(B|A)$. Sabemos también que $P(A \cup B) = \frac{1}{2}$ y que $P(A \cap B) > 0$. Muestra que $P(A) > \frac{1}{4}$.

Nota: Observa que A y B no son mutuamente excluyentes necesariamente. Considera cómo podemos relacionar $P(A \cup B)$ y $P(A \cap B)$.

Expectativa: Una demostración o derivación breve de a lo más 5 renglones.

f. Consideremos $\boldsymbol{w} \in \mathbb{R}^d$ (representado como un vector columna), constantes $\boldsymbol{a}_i, \boldsymbol{b}_j \in \mathbb{R}^d$ (también representados como vectores columna), $\lambda \in \mathbb{R}$ y un entero positivo n. Define la función

$$f(\boldsymbol{w}) = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \left(\boldsymbol{a}_i^\top \boldsymbol{w} - \boldsymbol{b}_j^\top \boldsymbol{w} \right)^2 \right) + \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2,$$

donde el vector es $\boldsymbol{w} = (w_1, \ldots, w_d)^\top$ y $\|\boldsymbol{w}\|_2 = \sqrt{\sum_{k=1}^{d} w_k^2} = \sqrt{\boldsymbol{w}^\top \boldsymbol{w}}$ es conocida como la norma L_2. Calcula el gradiente $\nabla f(\boldsymbol{w})$.

Nota: Recuerda que el gradiente es un vector d-dimensional de las derivadas parciales con respecto a cada w_i:

$$\nabla f(\boldsymbol{w}) = \left(\frac{\partial f(\boldsymbol{w})}{\partial w_1}, \ldots, \frac{\partial f(\boldsymbol{w})}{\partial w_d} \right)^\top .$$

Si batallas con este problema, primero intenta trabajarlo usando escalares en lugar de vectores y derivadas en lugar de gradientes.

Expectativa: Una expresión para el gradiente y el desarrollo realizado para derivarlo. No necesitas expandir términos innecesariamente, intenta escribir la respuesta final de forma compacta.

2. Complejidad. Al diseñar algoritmos, es útil poder hacer cálculos rápidos y detallados para ver cuánto tiempo o espacio necesita.

a. Supongamos que tenemos una cuadrícula de $n \times n$, donde nos gustaría colocar cuatro rectángulos alineados a los ejes (es decir, los lados del rectángulo son paralelos a los ejes). No hay restricciones sobre la ubicación o tamaño de los rectángulos. Por ejemplo, es posible que todas las esquinas de un rectángulo sean el mismo punto (resultando en un rectángulo de tamaño cero) o que los cuatro rectángulos se traslapen entre sí. ¿Cuántas formas distintas hay de colocar los cuatro rectángulos en la cuadrícula? En general solo nos interesa la complejidad asintótica, entonces presenta tu respuesta de la forma $O(n^c)$ u $O(c^n)$ para algún entero c.

Nota: No es necesario considerar si el orden importa en este problema, ya que se pide la complejidad asintótica. La respuesta final no cambia.

Expectativa: Una cota superior asintótica (en notación O grande) de la cantidad de maneras de colocar cuatro rectángulos y una explicación o razonamiento simple para la respuesta.

b. Supongamos que tenemos una cuadrícula de $3 \times 3n$. Comenzamos en la esquina superior izquierda (posición $(1,1)$) y nos gustaría alcanzar la esquina inferior derecha (posición $(n,3n)$) tomando pasos individuales hacia abajo o hacia la derecha. Supongamos que se nos provee con una función $c(i,j) \in \mathbb{R}$ del costo de pasar por la posición (i,j) y que toma tiempo constante calcular cada posición. Presenta un algoritmo para calcular el costo del camino de mínimo costo desde $(1,1)$ hasta $(n,3n)$ de la manera más eficiente (con la menor complejidad en tiempo en notción O grande). ¿Cuál es el tiempo de ejecución?

> **Expectativa:** Una descripción del algoritmo para calcular el costo del camino de mínimo costo de la manera más eficiente posible. El tiempo de ejecución en notación O grande y una explicación breve de cómo se obtiene esta complejidad del algoritmo.

3. Programación. En este problema, debes implementar varias funciones pequeñas. El propósito principal del ejercicio es que te familiarices con Python, sin embargo, las funciones que implementes van a ser útiles en futuras tareas.

> **Expectativa:** El archivo `fundamentos.py` con las funciones implementadas de forma más clara y simple posible.

a. Implementa `find_alphabetically_first_word`.

b. Implementa `euclidean_distance`.

c. Implementa `mutate_sentences`.

d. Implementa `sparse_vector_dot_product`.

e. Implementa `increment_sparse_vector`.

f. Implementa `find_nonsingleton_words`.