Inteligencia Artificial - Fundamentos

Rodolfo Armando Jaramillo Ruiz
27 de Enero de 2023

Optimización y probabilidad

a. Empiezo por derivar $f(\theta)$:

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

Posteriormente se iguala a cero y procedo a despejar θ:

$$2\sum_{i=1}^{n} w_i(\theta - x_i) = 0$$

$$\sum_{i=1}^{n} w_i\theta - \sum_{i=1}^{n} w_i x_i = 0$$

$$\theta\sum_{i=1}^{n} w_i = \sum_{i=1}^{n} w_i x_i$$

$$\theta = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \tag{2}$$

Obtuve el valor de θ que optimiza la la función, pero ¿cómo sé que es un valor mínimo? Usando el criterio de la segunda derivada, por lo que procedo a derivar nuevamente la ecuación (1).

$$f''(\theta) = 2\sum_{i=1}^{n} w_i \tag{3}$$

Inmediatamente se puede ver que la segunda derivada es siempre positiva, dado que w_i son reales positivos. Por lo tanto, tomando el criterio de la segunda derivada, se concluye que el valor de θ obtenido en la ecuación (2) es un mínimo.
Este valor de θ obtenido tiene la forma de la expresión del centro de masa para una distribución discreta de masa, por tanto si hubiera alguna w_i negativa, alejaría el valor del análogo al centro de masa, como si fuera una "masa negativa".
b. Consideraciones:

$$\mathbf{x} = (x_1, ..., x_d) \in \mathbb{R}^d \tag{4}$$

$$f(\mathbf{x}) = \max_{s\in[-1,1]} \Sigma_{i=1}^{d} s x_i \tag{5}$$

$$g(\mathbf{x}) = \Sigma_{i=1}^{d} \max_{s_i\in[-1,1]} s_i x_i \tag{6}$$

Se ve que lo que hace la función f es tomar la suma de d reales y multiplicarla por un factor $s \in [-1, 1]$ que lo haga máximo, mientras que g hace lo mismo, pero no lo hace con la suma en sí, si no que multiplica a cada término por un s_i haciendo a cada termino $s_i x_i$ lo más grande para cada término. Por lo tanto, f es a lo más igual de grande que g (como en el caso de $d = 1$ donde $f = g$), es decir $f(\mathbf{x}) \leq g(\mathbf{x})$.
c. La función de recurrencia se contruye como sigue:

$$V(n) = 0 \quad \text{En el caso que n=1}$$

$$V(n) = 0 \quad \text{En el caso que n=2}$$

$$V(n) = V(n) - a \quad \text{En el caso que n=3}$$

$$V(n) = V(n) \quad \text{En el caso que n=4}$$

$$V(n) = V(n) \quad \text{En el caso que n=5}$$

$$V(n) = V(n) + b \quad \text{En el caso que n=6}$$

Por lo que la función completa sería:

$$V(n) = 4V(n) + b - a \tag{7}$$

Por lo que solo queda calcular la esperanza como sigue:

$$E[V(n)] = \frac{1}{6}(4V(n) + b - a)$$

d. Tengo que para la secuencia $\{S, A, A, A, S, A\}$ la probabilidad de obtenerla es la siguiente función:

$$L(p) = (1-p)ppp(1-p)p = p^4(1-p)^2 \tag{8}$$

El valor de p que maximiza la probabilidad de obtener esta secuencia se obtiene de la siguiente forma: Primero; se puede ver que $L(p)$ es un polinomio de grado 6, por lo que la primera derivada es un polinomio de grado 5, que al igualarla a cero para optimizar $L(p)$ se obtendrá que hay que despejar una ecuación de quinto grado, lo cual es, por decir lo menos, complicado. Por lo tanto, me iré por el camino de optimizar la función $ln[L(p)]$ que, debido a que la función logaritmo natural es una función creciente, conserva los máximos y mínimos de la función de la función original.
Procedo:

$$\frac{d}{dp}\{ln[L(p)]\} = \frac{d}{dp}\{ln[p^4(1-p)^2]\} \tag{9}$$

$$\frac{d}{dp}\{ln[L(p)]\} = \frac{d}{dp}\{ln[p^4] + ln[(1-p)^2]\}$$

$$\frac{d}{dp}\{ln[L(p)]\} = \frac{d}{dp}\{4ln[p] + 2ln[(1-p)]\}$$

$$\frac{d}{dp}\{ln[L(p)]\} = 4\frac{d}{dp}ln[p] + 2\frac{d}{dp}ln[(1-p)]$$

$$\frac{d}{dp}\{ln[L(p)]\} = \frac{4}{p} - \frac{2}{1-p} \tag{10}$$

Se iguala la derivada a cero y se despeja p:

$$\frac{4}{p} - \frac{2}{1-p} = 0 \tag{11}$$

$$\frac{4(1-p) - 2p}{p(1-p)} = 0$$

$$4(1-p) - 2p = 0$$

$$4 - 4p - 2p = 0 \rightarrow 4 - 6p = 0$$

$$p = \frac{2}{3} \tag{12}$$

Para saber si este valor es un máximo, tengo que verificar que la segunda derivada evaluada en este punto sea menor a cero:

$$\frac{d^2}{dp^2}\{ln[L(p)]\} = -\frac{4}{p^2} - \frac{2}{(1-p)^2} \tag{13}$$

$$\frac{d^2}{dp^2}\{ln[L(p)]\} = -\left(\frac{4}{p^2} + \frac{2}{(1-p)^2}\right) \tag{14}$$

Inmediatamente se ve que lo contenido entre paréntesis en la ecuación anterior es positivo, cada término es positivo sin importar el valor de p. Por lo tanto, $p = 2/3$ maximiza la probabilidad de obtener la secuencia $\{S, A, A, A, S, A\}$. Significa que la moneda debe estar cargada para caer águila con probabilidad $2/3$ para que sea lo más probable obtener esa secuencia.

e. Al ser que $P(A|B) = P(B|A)$ y que $P(X|Y) = P(X \cap Y)/P(X)$ se tiene que:

$$\frac{P(A \cap B)}{P(B)} = \frac{P(A \cap B)}{P(A)} \to \frac{1}{P(B)} = \frac{1}{P(A)} \to P(A) = P(B)$$

Ahora tenemos que la probabilidad de que ocurra A es igual a la probabilidad de que ocurra B. Considerando la probabilidad de la unión:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) = 1/2 \tag{15}$$

$$2P(A) - P(A \cap B) = 1/2 \to 2P(A) = 1/2 + P(A \cap B)$$

$$P(A) = 1/4 + \frac{1}{2}P(A \cap B) \tag{16}$$

Por lo tanto, sabiendo que $P(A \cap B) > 0$:

$$P(A) > 1/4 \tag{17}$$

f. La función

$$f(\mathbf{w}) = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{a}_i^\top \mathbf{w} - \mathbf{b}_i^\top \mathbf{w})^2 \right) + \frac{\lambda}{2} ||\mathbf{w}||_2^2 \tag{18}$$

Tiene un gradiente que se calcula como sigue:

$$\frac{\mathrm{d}}{\mathrm{d}w_k} f(w_k) = \frac{\mathrm{d}}{\mathrm{d}w_k} \left[\left(\sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{a}_i^\top w_k - \mathbf{b}_i^\top w_k)^2 \right) + \frac{\lambda}{2} ||w_k||_2^2 \right]$$

$$\frac{\mathrm{d}}{\mathrm{d}w_k} f(w_k) = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} 2(\mathbf{a}_i^\top - \mathbf{b}_i^\top)(\mathbf{a}_i^\top w_k - \mathbf{b}_i^\top w_k) \right) + \lambda ||w_k||_2$$

$$\frac{\mathrm{d}}{\mathrm{d}w_k} f(w_k) = \left(2 \sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{a}_i^\top - \mathbf{b}_i^\top)(\mathbf{a}_i^\top w_k - \mathbf{b}_i^\top w_k) \right) + \lambda ||w_k||_2$$

Este sería la forma de los elementos del gradiente:

$$\nabla_i f(\mathbf{w}) = \left(2 \sum_{i=1}^{n} \sum_{j=1}^{n} (\mathbf{a}_i^\top - \mathbf{b}_i^\top)(\mathbf{a}_i^\top w_i - \mathbf{b}_i^\top w_i) \right) + \lambda ||w_i||_2 \tag{19}$$

Complejidad

a. Para empezar este problema podemos considerar el problema de un rectángulo en una cuadricula de $n \times n$, donde se tiene que recorrer cada vértice que conforma la cuadricula (n^2), en donde se va a incrementar la longitud y la altura (n^2), de forma que para este caso sería una complejidad $O(n^4)$. Por lo que para cuatro cuadrados sería $O(n^16)$.
b. Bajo estas condiciones la función de recurrencia

$$V(i, j) = c(i, j) + \min\{V(i-1, j), V(i, j-1)\}$$

Para no calcular los mismos costos de utiliza una matriz que conserva los costos de cada nodo que ya fue calculado. El algoritmo queda como sigue:

function V:

if $i == 0$ **or** $j == 0$ **then**

return 0

end

if $grid[i][j] == cost$ **then**

return $grid[i][j] = c(i, j) + \text{mín}\{V(i-1, j), V(i, j-1)\}$

end

return $grid[i][j]$