Descenso Gradiente de Newton

Alan Reyes-Figueroa
Métodos Numéricos II (Aula 22) 28.septiembre.2023

Descenso Gradiente

Otra dirección de búsqueda importante es la **dirección de Newton**. Ésta se deriva de la aproximación de Taylor de segundo orden

$$\begin{aligned} f(\mathbf{x}_k + \mathbf{d}) &= f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{d} + \tfrac{1}{2}\mathbf{d}^T D^2 f(\mathbf{x}_k)\, \mathbf{d} + o(||\mathbf{d}||^2). \\ &\approx \underbrace{f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T \mathbf{d} + \tfrac{1}{2}\mathbf{d}^T D^2 f(\mathbf{x}_k)\, \mathbf{d}}_{m_k(\mathbf{d})}. \end{aligned} \qquad (1)$$

Observe que $m_k(\mathbf{d})$ es una función cuadrática en $\mathbb{R}^n$. Si $D^2 f(\mathbf{x}_k)$ es positiva definida, entonces m_k es convexa, y encontramos la dirección de Newton hallando el vector $\mathbf{d} \in \mathbb{R}^n$ como el mínimo global de esta función cuadrática. Esto es

$$\nabla m_k(\mathbf{d}) = \nabla f(\mathbf{x}_k) + D^2 f(\mathbf{x}_k)\, \mathbf{d} = \mathbf{0} \quad \Longrightarrow \quad \mathbf{d}_{Newton} = -\big(D^2 f(\mathbf{x}_k)\big)^{-1} \nabla f(\mathbf{x}_k).$$

- Podemos usar la dirección de Newton en un método de descenso gradiente siempre que $D^2 f \succ 0$.
- Usamos tamaño de paso $\alpha = 1$ con la dirección de Newton. Sin embargo, α puede ajustarse cuando los resultados no son satisfactorios.

Descenso Gradiente

Algoritmo: (*Descenso gradiente*, versión Newton)
Inputs: $f : \mathbb{R}^n \to \mathbb{R}$ función de clase C^2, con Hessiana $D^2 f$ positiva definida en cada punto; $\mathbf{x}_0 \in \mathbb{R}^n$, $\alpha_k > 0$ tamaño de paso (usualmente $\alpha_k = 1$).
Outputs: $\mathbf{x}$ punto crítico de f.

```
For k = 0,1,2,... hasta que se cumpla un criterio de paro:
    Define d_k = -(D^2 f(x_k))^{-1} ∇f(x_k),
    Set x_{k+1} = x_k + α_k d_k.
Return x_{k+1}.
```

For k = 0,1,2,... hasta que se cumpla un criterio de paro:
- Define $\mathbf{d}_k = -\big(D^2 f(\mathbf{x}_k)\big)^{-1} \nabla f(\mathbf{x}_k)$,
- Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$.

Return $\mathbf{x}_{k+1}$.

Obs:

- Cuando $D^2 f(\mathbf{x}_k)$ no es positiva definida en alguno de los puntos iterados $\mathbf{x}_k$, el método aún se pude utilizar. En este caso, se reemplaza el hessiano por su aproximación simétrica $A \in \mathbb{R}^{n\times n}$, más cercana, que sea positiva definida.
- Esto puede hacerse hallando la descomposición espectral $D^2 f(\mathbf{x}_k) = U \Lambda U^T$, y reemplazando todos los autovalores negativos de Λ por $\varepsilon > 0$; $A = U \Lambda_\varepsilon U^T$.

Descenso Gradiente

Algoritmo: (*isPSD*, is Positive Definite?)
Inputs: $A \in \mathbb{R}^{n \times n}$ matriz simétrica, $\varepsilon > 0$ un número muy cercano a 0 (*e.g.* $\varepsilon = 10^-6$).
Outputs: True or False, dependiento de si A es positiva definida.

```
Get all eigenvector λ_i of A.
If all λ_i > ε:  return True.
Else:  return False.
```

Algoritmo: (*nearPSD*, Aproximación Positiva Definida)
Inputs: $A \in \mathbb{R}^{n \times n}$ matriz simétrica, $\varepsilon > 0$ un número muy cercano a 0 (*e.g.* $\varepsilon = 10^-6$).
Outputs: A^+, la matriz positiva definida más cercana a A.

```
If isPSD(A) = True:  return A.
Get UΛU^T the spectral decomposition of A.
Λ+ = Λ.copy()
Λ+[Λ+ < ε] = ε.  (sustituir los valores λ_i negativos ó 0 por ε)
Reconstruct A+ = UΛ+U^T
Return A+.
```

Descenso Gradiente

Otra alternativa para aproximar la dirección de búsqueda, es hacer uso de una la siguiente aproximación de Taylor de primer orden, sobre el gradiente de f:

$$\begin{aligned} \nabla f(\mathbf{x}_k + \alpha \mathbf{d}_k) &= \nabla f(\mathbf{x}_k) + \alpha\, D^2 f(\mathbf{x}_k)^T\, \mathbf{d}_k + o(||\mathbf{d}_k||). \\ &\approx \nabla f(\mathbf{x}_k) + \alpha\, D^2 f(\mathbf{x}_k)^T\, \mathbf{d}_k. \end{aligned} \tag{2}$$

Queremos hallar el valor de α que minimiza el valor para $f(\mathbf{x}_k + \alpha \mathbf{d}_k)$. Para ello, hacemos $\nabla f(\mathbf{x}_k + \alpha \mathbf{d}_k) = \mathbf{0}$. (Observe que esto funciona si $D^f(\mathbf{x}_k)$ es positiva definida).

Luego, $\mathbf{0} = \nabla f(\mathbf{x}_k) + \alpha\, D^2 f(\mathbf{x}_k)^T\, \mathbf{d}_k$. Multiplicando esta ecuación por $\nabla f(\mathbf{x}_k)^T$ de ambos lados, obtenemos:

$$0 = \nabla f(\mathbf{x}_k)^T\, \nabla f(\mathbf{x}_k) + \alpha\, \nabla f(\mathbf{x}_k)^T\, D^2 f(\mathbf{x}_k)^T\, \mathbf{d}_k.$$

Sustituyendo $\mathbf{d}_k = -\nabla f(\mathbf{x})_k$ y despejando α de la ecuación resultante, obtenemos

$$\boxed{\alpha_k = \frac{\nabla f(\mathbf{x}_k)^T\, \nabla f(\mathbf{x}_k)}{\nabla f(\mathbf{x}_k)^T\, D^2 f(\mathbf{x}_k)\, \nabla f(\mathbf{x}_k)}.} \tag{3}$$

Descenso Gradiente

La ecuación (3) usa la información del Hessiano para elegir el tamaño de paso α_k que debemos movernos.

Algoritmo: (*Descenso gradiente*, versión Hessiano aproximado)
Inputs: $f : \mathbb{R}^n \to \mathbb{R}$ función de clase C^2, con Hessiana $D^2 f$ positiva definida en cada punto; $\mathbf{x}_0 \in \mathbb{R}^n$.
Outputs: $\mathbf{x}$ punto crítico de f.
`For k = 0,1,2,... hasta que se cumpla un criterio de paro:`
    `Define` $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$,
    `Compute` α_k using equation (3)

$$\alpha_k = \frac{\nabla f(\mathbf{x}_k)^T \, \nabla f(\mathbf{x}_k)}{\nabla f(\mathbf{x}_k)^T \, D^2 f(\mathbf{x}_k) \, \nabla f(\mathbf{x}_k)}.$$

    `Set` $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$.
`Return` $\mathbf{x}_{k+1}$.

Obs: Si $D^2 f(\mathbf{x}_k)$ no es positiva definida, sustituimos por la aproximación positiva definida más cercana.

Descenso Gradiente

- El cálculo de la hessiana $D^2f(\mathbf{x}_k)$ en cada iteración, consume mucho costo computacional (sobretodo en altas dimensiones).

  Existen otros métodos de tipo gradiente que, en lugar de calcular exactamente el hessiano $D^2f(\mathbf{x}_k)$, utilizan una aproximación B_k, que se actualiza en cada paso.

  De la aproximación de Taylor

$$\begin{aligned}\nabla f(\mathbf{x}_k + \mathbf{d}) &= \nabla f(\mathbf{x}_k) + \int_0^1 D^2f(\mathbf{x}_k + t\mathbf{d})\,\mathbf{d}\,dt \\ &= \nabla f(\mathbf{x}_k) + D^2f(\mathbf{x}_k)\,\mathbf{d} + \underbrace{\int_0^1 \big[D^2f(\mathbf{x}_k + t\mathbf{d}) - D^2f(\mathbf{x}_k)\big]\mathbf{d}\,dt}_{o(||\mathbf{d}||)}.\end{aligned}$$

  Haciendo $\mathbf{d} = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\Rightarrow \nabla f(\mathbf{x}_{k+1}) = \nabla f(\mathbf{x}_k) + D^2f(\mathbf{x}_k)\,(\mathbf{x}_{k+1} - \mathbf{x}_k) + o(||\mathbf{d}||)$. Cuando $\mathbf{x}_k, \mathbf{x}_{k+1}$ están en una región cercana al mínimo $\mathbf{x}^*$, donde $D^2f(\mathbf{x}_k) \succ 0$, resulta

$$D^2f(\mathbf{x}_k)\,\mathbf{d} \approx \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k). \tag{4}$$

Descenso Gradiente

Así, elegimos la aproximación de B_{k+1} de modo que imite la propiedad (4) anterior. Así, requerimos que B_{k+1} cumpla la **ecuación secante**:

$$B_{k+1}\mathbf{s}_k = \mathbf{y}_k, \tag{5}$$

donde $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, y $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$. Además, requerimos que B_{k+1} sea simétrica, y que la diferencia $B_{k+1} - B_k$ sea de bajo rango.

Estos son los métodos llamados **métodos quasi-Newton**. Dos de las fórmulas más populares para actualizar el hessiano son

- el método **simétrico de rango 1 (SR1)**:

$$B_{k+1} = B_k + \frac{(\mathbf{y}_k - B_k\mathbf{s}_k)(\mathbf{y}_k - B_k\mathbf{s}_k)^T}{(\mathbf{y}_k - B_k\mathbf{s}_k)^T\mathbf{s}_k}.$$

- el método **BFGS (Broyden-Fletcher-Goldfarb-Shanno)**:

$$B_{k+1} = B_k - \frac{B_k\mathbf{s}_k\mathbf{s}_k^T B_k^T}{\mathbf{s}_k^T B_k\mathbf{s}_k} + \frac{\mathbf{y}_k\mathbf{y}_k^T}{\mathbf{y}_k^T\mathbf{s}_k}.$$