Como contornar a Hessiana

Método da sub-amostragem e método da secante

(link para o github)

Daniel Roizman

05/11/202

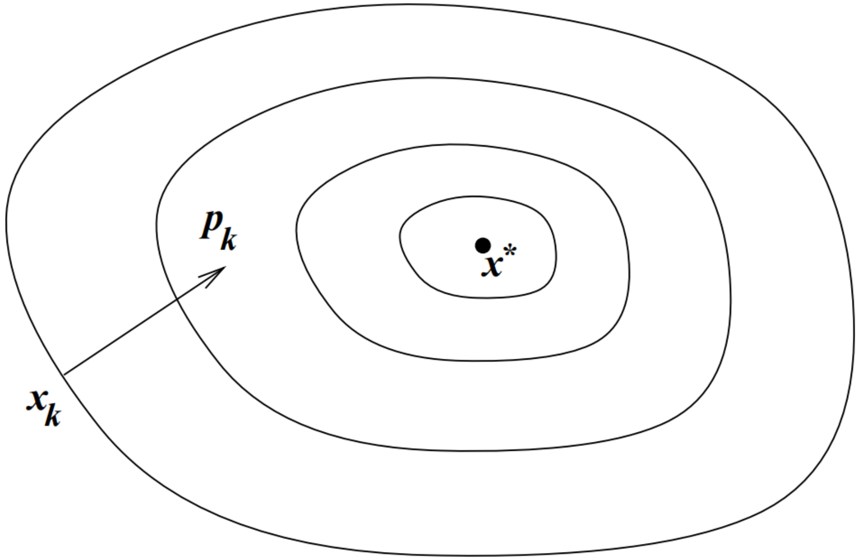
PARTE 1

Problema e motivação

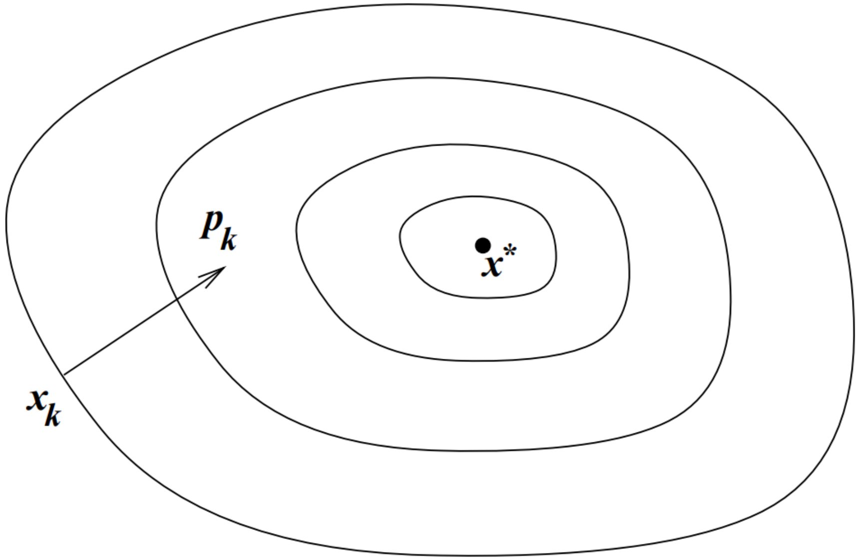
1.1. O problema

1.1. O problema

a. Queremos $x^* = \text{argmin } g(x)$.



1.1. O problema



a. Queremos $x^* = \text{argmin } g(x)$.

b. Métodos de 2ª ordem:

- $x_k - x_{k-1} = p_k \approx -(\nabla^2 g_{k-1})^{-1}(\nabla g_{k-1})$, se $\nabla^2 g$ for positiva definida.

1.2. Considerações computacionais

1.2. Considerações computacionais

a. Instabilidade numérica.

1.2. Considerações computacionais

a. Instabilidade numérica.

b. Hessiana custa $\boldsymbol{O(n^2)}$, e inverter matrizes custa de $\boldsymbol{O(n)}$ (diagonal) a $\boldsymbol{O(n^3)}$ (pior cenário, via Gauss-Jordan)

1.3. Alternativas

1.3. Alternativas

a. Aproximamos a Hessiana, usando métodos *quasi-Newton*:

1.3. Alternativas

a. Aproximamos a Hessiana, usando métodos *quasi-Newton*:

1. Sub-amostragem;

2. Aproximação linear (secante);

1.3. Alternativas

a. Aproximamos a Hessiana, usando métodos *quasi-Newton*:

   1. Sub-amostragem;
   2. Aproximação linear (secante);

b. Taxa de convergência ainda assim pode ser boa.

PARTE 2

Método da sub-amostragem da hessiana

2.1. O modelo original

2.1. O modelo original

A aproximação (quadrática) de Taylor de g em $x_{k+1} = x_k + p_k \in \mathbb{R}^n$ é

$$g_{k+1} \approx g_k + \nabla g_k p_k + \frac{1}{2}\langle p_k, \nabla^2 g_k p_k \rangle$$

2.1. O modelo original

A aproximação (quadrática) de Taylor de g em $x_{k+1} = x_k + p_k \in \mathbb{R}^n$ é

$$g_{k+1} \approx g_k + \nabla g_k p_k + \frac{1}{2}\langle p_k, \nabla^2 g_k p_k \rangle$$

A condição de primeira ordem nos dá:

$$\nabla g_k + \nabla^2 g_k p_k = 0$$

$$\Rightarrow p_k = -(\nabla^2 g_k)^{-1} \nabla g_k,$$

2.1. O modelo original

A aproximação (quadrática) de Taylor de g em $x_{k+1} = x_k + p_k \in \mathbb{R}^n$ é

$$g_{k+1} \approx g_k + \nabla g_k p_k + \frac{1}{2}\langle p_k, \nabla^2 g_k p_k \rangle$$

A condição de primeira ordem nos dá:

$$\nabla g_k + \nabla^2 g_k p_k = 0$$

$$\Rightarrow p_k = -(\nabla^2 g_k)^{-1} \nabla g_k,$$

De modo que o passo de Newton é

$$x_{k+1} = x_k + p_k = x_k - (\nabla^2 g_k)^{-1} \nabla g_k$$

2.2. Hessiana sub-amostrada

Isto é

$$\begin{bmatrix} x_1^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{bmatrix} = \begin{bmatrix} x_1^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix} + \begin{bmatrix} g_{k\,x_1x_1} & \cdots & g_{k\,x_nx_1} \\ \vdots & \ddots & \vdots \\ g_{k\,x_1x_n} & \cdots & g_{k\,x_nx_n} \end{bmatrix}^{-1} \begin{bmatrix} \nabla g_{x_1} \\ \vdots \\ \nabla g_{x_n} \end{bmatrix}$$

2.2. Hessiana sub-amostrada

Isto é

$$\begin{bmatrix} x_1^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{bmatrix} = \begin{bmatrix} x_1^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix} + \begin{bmatrix} g_{k\,x_1x_1} & \cdots & g_{k\,x_nx_1} \\ \vdots & \ddots & \vdots \\ g_{k\,x_1x_n} & \cdots & g_{k\,x_nx_n} \end{bmatrix}^{-1} \begin{bmatrix} \nabla g_{x_1} \\ \vdots \\ \nabla g_{x_n} \end{bmatrix}$$

$$\approx \begin{bmatrix} x_1^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix} + \begin{bmatrix} {g_k^{-1}}_{x_1x_1} & 0 & \cdots & 0 \\ 0 & {g_k^{-1}}_{x_2x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & {g_k^{-1}}_{x_nx_n} \end{bmatrix} \begin{bmatrix} \nabla g_{x_1} \\ \vdots \\ \nabla g_{x_n} \end{bmatrix}$$

2.3. Pseudo-código: newton original

Dê um chute w_0, uma tolerância $\varepsilon > 0$, um tamanho de passo $\alpha > 0$, e um limite de iterações M.

while $\|w_k - w_{k-1}\| > \varepsilon$ **and** $k < M$

$\nabla g_k = Grad(g_k)$

$H_k = Hessian(g_k)$

$p_\mathrm{k} = -(H_k)^{-1} \nabla g_k$

$w_{k+1} = w_k + \alpha p_k$

$k = k + 1$

end **while**

2.3. Pseudo-código: newton modificado

Dê um chute w_0, uma tolerância $\varepsilon > 0$, um tamanho de passo $\alpha > 0$, e um limite de iterações M.

while $\|w_k - w_{k-1}\| > \varepsilon$ **and** $k < M$

$$\nabla g_k = Grad(g_k)$$

$$H_k = Hessian(g_k)$$

$$p_\mathrm{k} = -(diag H_k)^{-1} \nabla g_k$$

$$w_{k+1} = w_k + \alpha p_k$$

$$k = k + 1$$

end **while**

2.4. Comparação computacional

2.4. Comparação computacional

Exemplo - Dados aleatórios (quase-separáveis), $N = 3$, $P = 500$ usando três métodos:

2.4. Comparação computacional

Exemplo - Dados aleatórios (quase-separáveis), N = 3, P = 500 usando três métodos:

1. Newton sub-amostrado
2. Newton original
3. Gradiente descendente

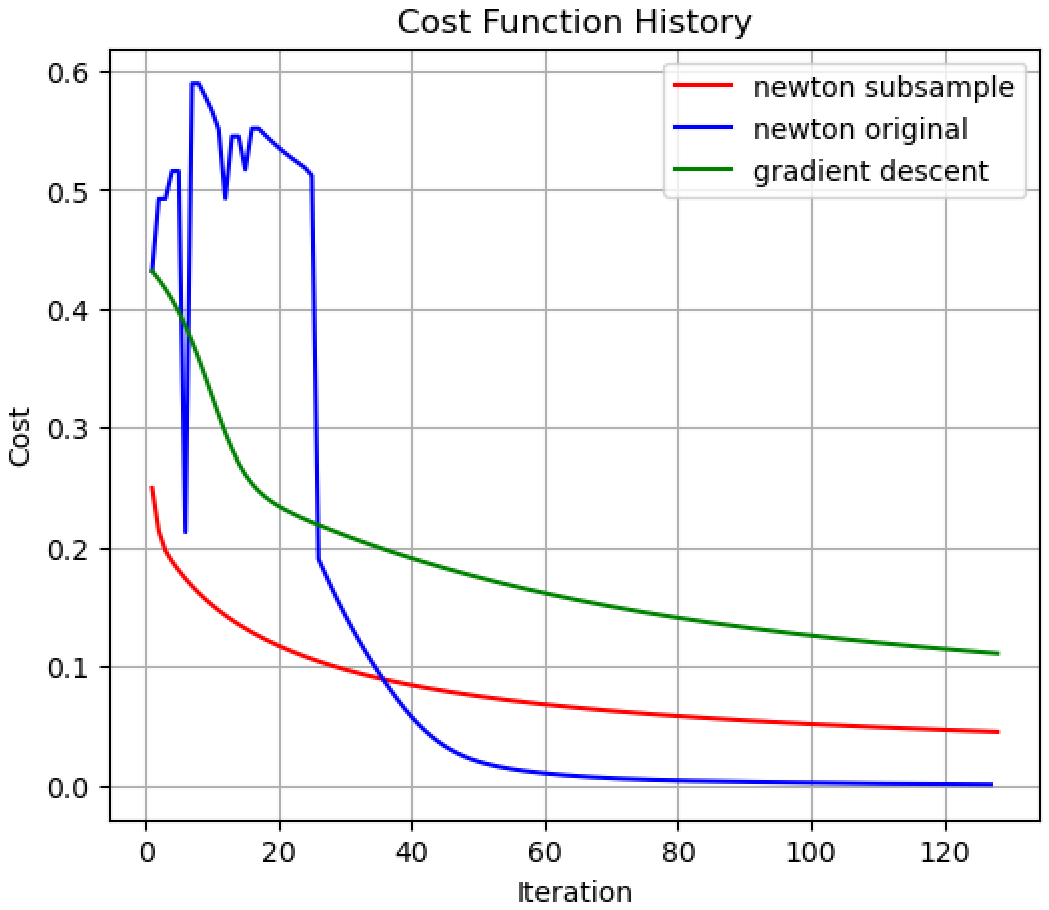
2.4. Comparação computacional

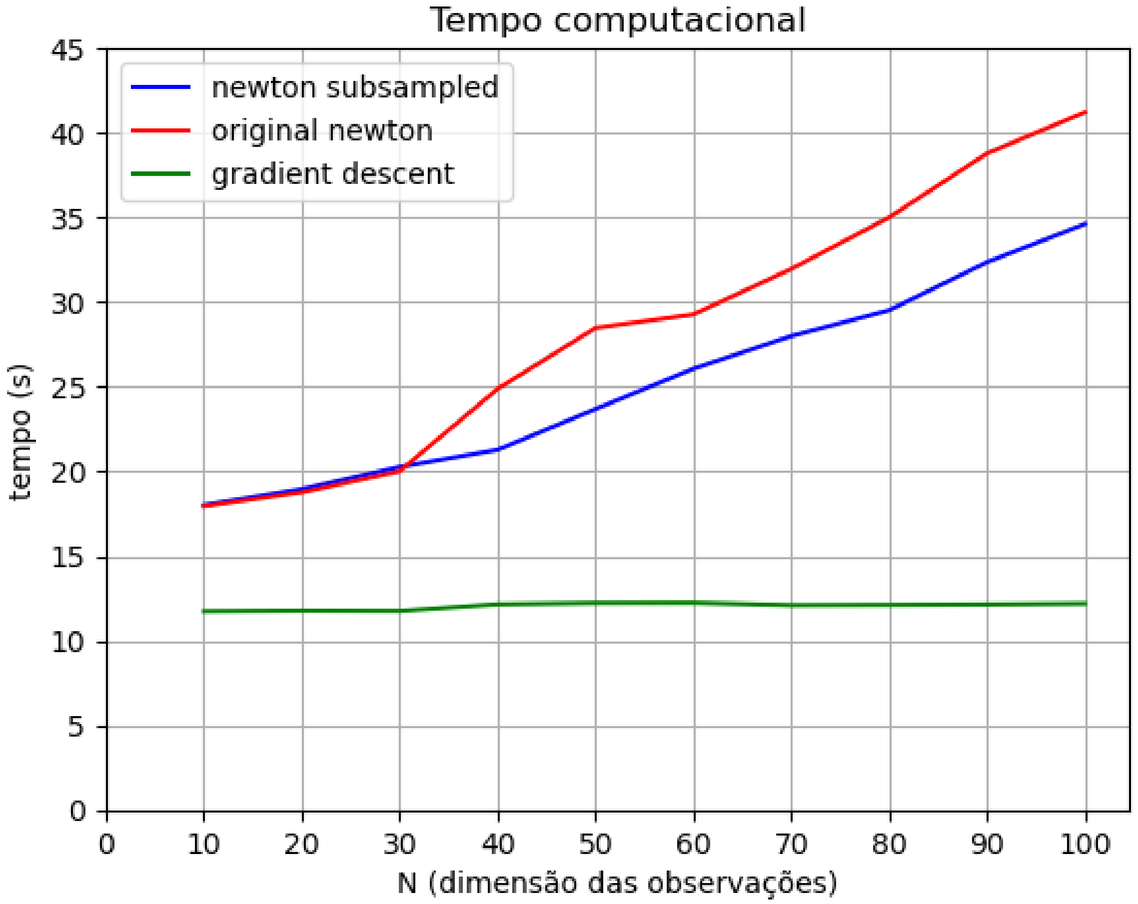
Exemplo - Dados aleatórios (quase-separáveis), N = 3, P = 500 usando três métodos:

1. Newton sub-amostrado
2. Newton original
3. Gradiente descendente

```
Acurácias:
newton subsample:   99.6 %
newton original:   100.0 %
gradient descent:   99.4 %
```

2.4. Comparação computacional



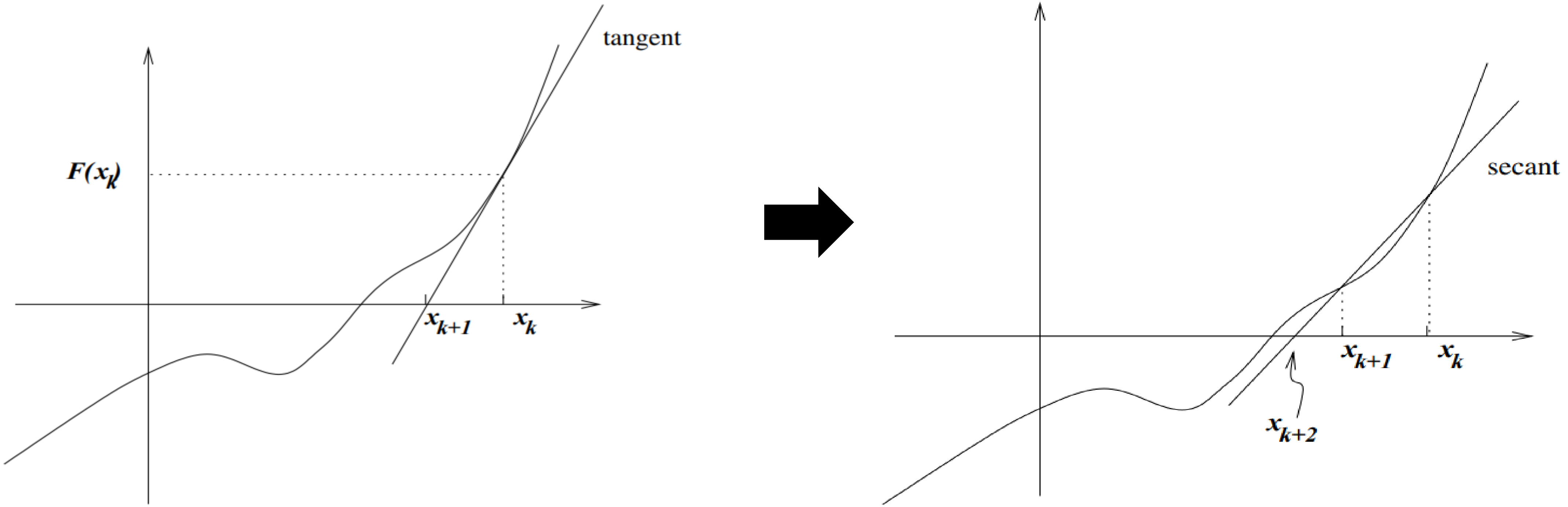


PARTE 3

Método da secante

3.1. A ideia

3.1. A ideia



3.2. O método BFGS

3.2. O método BFGS

A aproximação quadrática de $g(x_k) = g_k$ em $x_k \in \mathbb{R}^n$ é

$$g(x_k + p) \approx m_k(p) = g_k + \nabla g_k^T p + \frac{1}{2} p^T B_k p$$

onde $p \in \mathbb{R}^n$, e $B_k \in \mathbb{R}^{n \times n}$ é a aproximação à Hessiana.

3.2. O método BFGS

A aproximação quadrática de $g(x_k) = g_k$ em $x_k \in \mathbb{R}^n$ é

$$g(x_k + p) \approx m_k(p) = g_k + \nabla g_k^T p + \frac{1}{2} p^T B_k p$$

onde $p \in \mathbb{R}^n$, e $B_k \in \mathbb{R}^{n\times n}$ é a aproximação à Hessiana. Veja que a direção ótima $p_k = \operatorname{argmin} m_k$ ocorre em

$$p_k = -B_k^{-1} \nabla g_k,$$

3.2. O método BFGS

A aproximação quadrática de $g(x_k) = g_k$ em $x_k \in \mathbb{R}^n$ é

$$g(x_k + p) \approx m_k(p) = g_k + \nabla g_k^T p + \frac{1}{2} p^T B_k p$$

onde $p \in \mathbb{R}^n$, e $B_k \in \mathbb{R}^{n \times n}$ é a aproximação à Hessiana. Veja que a direção ótima $p_k = \operatorname{argmin} m_k$ ocorre em

$$p_k = -B_k^{-1} \nabla g_k,$$

e dá o novo passo

$$x_{k+1} = x_k + \alpha p_k.$$

3.2. O método BFGS

Condição de curvatura: Faz sentido esperar que

$$\nabla g(x_k + p) \approx \nabla m_k(p), \qquad \nabla g(x_{k+1} + p) \approx \nabla m_{k+1}(p)$$

3.2. O método BFGS

Condição de curvatura: Faz sentido esperar que

$$\nabla g(x_k + p) \approx \nabla m_k(p), \qquad \nabla g(x_{k+1} + p) \approx \nabla m_{k+1}(p)$$

Ou seja:

i. $\nabla m_{k+1}(p = 0) = \nabla g_{k+1}$.

ii. $\nabla m_{k+1}(p = -\alpha p_k) = \nabla g_k$

3.2. O método BFGS

Verificando as condições, temos que:

i. $\nabla m_{k+1}(0)$ é exatamente ∇g_{k+1}, pra qualquer B_{k+1}.

ii. $\nabla m_{k+1}(-\alpha_k p_k) = \nabla g_{k+1} - \alpha_k B_{k+1} p_k = \nabla g_k$

$$\Rightarrow B_{k+1} \alpha_k p_k = \nabla g_{k+1} - \nabla g_k$$

$$\Rightarrow B_{k+1} s_k = y_k,$$

(Equação secante)

onde $s_k := x_{k+1} - x_k$, e $y_k := \nabla g_{k+1} - \nabla g_k$

3.2. O método BFGS

Infinitas soluções B para a equação secante

3.2. O método BFGS

Infinitas soluções B para a equação secante

Queremos $B_{k+1} = \text{argmin}\|B_k - B_{k+1}\|$, sujeito a

$$B_{k+1} = B_{k+1}^T,$$

$$B_{k+1} s_k = y_k$$

3.2. O método BFGS

Em vez inverter B pra todo k, vamos trabalhar com $H_k = B_k^{-1}$

O novo problema é $H_{k+1} = \text{argmin}\|H_k - H_{k+1}\|$,

sujeito a $H_{k+1} = H_{k+1}^T,\ s_k = H_{k+1} y_k$

3.2. O método BFGS

Em vez inverter B pra todo k, vamos trabalhar com $H_k = B_k^{-1}$

O novo problema é $H_{k+1} = \text{argmin}\|H_k - H_{k+1}\|$,

sujeito a $H_{k+1} = H_{k+1}^T,\ s_k = H_{k+1}y_k$

BFGS sugere a norma de Frobenius ponderada $\|A\|_W = \left\|W^{\frac{1}{2}}AW^{\frac{1}{2}}\right\|_F$, onde

$$W = \int_0^1 \nabla^2 g(x_k + t\alpha p_k)dt$$

3.2. O método BFGS

Não calculamos $\nabla^2 g$, pois essa escolha dá uma solução simples:

$$H_{k+1} = \left(I - \rho_k s_k y_k^T\right) H_k \left(I - \rho_k s_k y_k^T\right) + \rho_k s_k s_k^T,$$

onde $\rho_k = \frac{1}{y_k^T s_k}$

3.3. pseudocódigo do BFGS

Dê um chute inicial x_0, uma tolerância $\varepsilon > 0$, uma aproximação H_0 à inversa da Hessiana, e um tamanho de passo $\alpha > 0$.

while $\|\nabla g_k\| > \varepsilon$:

$$p_{\mathrm{k}} = -H_k \nabla g_k$$

$$x_{k+1} = x_k + \alpha p_k$$

$$s_k = x_k - x_{k-1}$$

$$y_k = \nabla g_{k+1} - \nabla g_k$$

$$H_{k+1} = \mathrm{BFGS}(H_k)$$

$$k = k + 1$$

end **while**

3.4. Comparação computacional

3.4. Comparação computacional

Complexidade esperada é de $O(n^2)$

3.4. Comparação computacional

Complexidade esperada é de $O(n^2) \ll O(n^3)$

4. Referências

- BORHANI, R., KATSAGGELOS, A. K., WATT, J.: **Machine Learning Refined,** 2ª edição.
- NOCEDAL, J., WRIGHT, S. J.: **Numerical Optimization**

Fim