Otimização via Gradiente Descendente

Prof. Dr. Guilherme de Alencar Barreto

Outubro/2015

Departmento de Engenharia de Teleinformática
Programa de Pós-Graduação em Eng. de Teleinformática (PPGETI)
Universidade Federal do Ceará (UFC), Fortaleza-CE

gbarreto@ufc.br

1 Definições Preliminares

Considere uma função escalar (contínua) de p variáveis, $f : \mathbb{R}^p \rightarrow \mathbb{R}$; ou seja, $f(\cdot)$ é uma função que produz uma saída escalar $y \in \mathbb{R}$ e que possui p ($p \geq 1$) variáveis de entrada. Formalmente, podemos escrever

$$y = f(\mathbf{x}) = f(x_1, x_2, \ldots, x_p), \tag{1}$$

em que $\mathbf{x}$ é um vetor cujas componentes são as variáveis x_i, $i = 1, \ldots, p$. Como exemplos, para $p = 1$, temos que a parábola é uma função dada por

$$y = f(x) = (x - a)^2 + b, \tag{2}$$

cujo gráfico para $a = b = 5$ no plano cartesiano está mostrada na Figura 1a. A função equivalente em três dimensões é chamada de parabolóide, sendo escrita matematicamente como

$$z = f(x, y) = (x - a)^2 + (y - b)^2 + c, \tag{3}$$

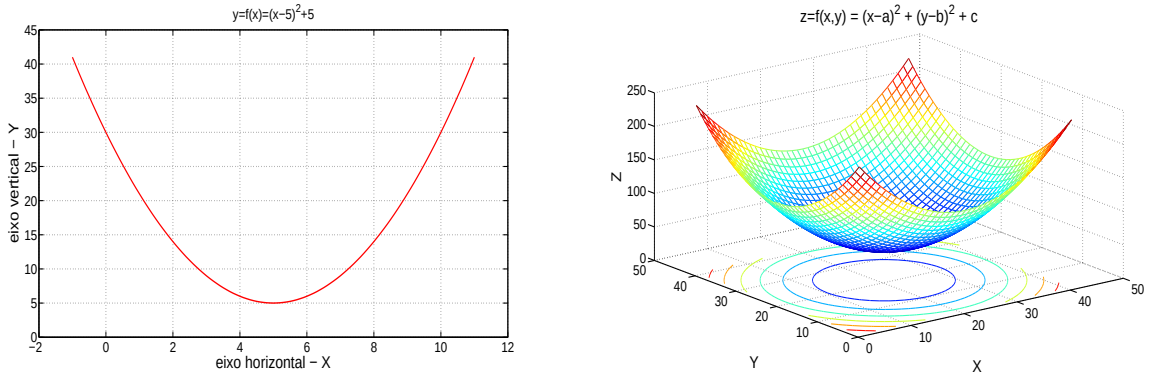
em que a, b e c são constantes reais. O gráfico desta função é uma superfície em 3 dimensões, mostrada na Figura 1b.

Note que as funções acima são contínuas e de variação suaves (i.e. não possuem interrupções ao longo do seu domínio, nem mudanças bruscas em seus gráficos). Além disso, são também funções convexas, ou seja, possuem apenas um ponto extremo, chamado de *mínimo global* neste caso.

Se a função mostrada na Eq. (2) for usada como função-custo ou função-objetivo em algum problema de minimização, o valor da variável x para o qual $y = f(x)$ produz seu maior valor é chamado de valor ótimo de x, simbolizado como x_{opt}. Se o problema envolver duas variáveis, como a função mostrada na Eq. (3), busca-se um vetor ótimo $\mathbf{x}_{opt} = [x_{opt} \quad y_{opt}]^T$, em que T simboliza o vetor-transposto.

De um modo geral, o problema de minimização de funções (sem restrições) pode ser formalizado matematicamente da seguinte maneira. O vetor $\mathbf{x}_{opt} \in \mathbb{R}^p$ é o vetor-ótimo se

$$f(\mathbf{x}_{opt}) < f(\mathbf{x}), \quad \forall \mathbf{x} \neq \mathbf{x}_{opt}. \tag{4}$$



(a) (b)

Figura 1: Representação gráfica de funções no plano cartesiano e em 3 dimensões. (a) $y = f(x) = (x-5)^2 + 5$, (b) $z = (x-20)^2 + (y-20)^2 + 50$.

ou, de maneira alternativa, como

$$\mathbf{x}_{opt} = \arg\min_{\forall \mathbf{x}} f(\mathbf{x}), \tag{5}$$

Nas próximas seções são apresentadas duas técnicas para obter o vetor-ótimo para uma função convexa, ambas utilizando o vetor-gradiente da função de interesse.

1.1 Método Não-Iterativo

Este é o método mais simples e direto quando a função é convexa, contínua e diferenciável de ordem 1 (i.e. a primeira derivada existe). Para este fim, basta determinar o vetor-gradiente da função de interesse e igualá-lo ao vetor-nulo de dimensão p. Formalmente, esta técnica pode ser escrita como

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \mathbf{0}, \tag{6}$$

que, em termos das componentes do vetor-gradiente, pode também ser escrita como

$$\begin{bmatrix} \frac{\partial f(x_1,\ldots,x_p)}{\partial x_1} \\ \frac{\partial f(x_1,\ldots,x_p)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x_1,\ldots,x_p)}{\partial x_p} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{7}$$

1.1.1 Exemplo 1

Tomando como exemplo a função mostrada na Eq. (3), temos que o vetor-gradiente para esta função é dado por

$$\begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix} = \begin{bmatrix} 2(x-a) \\ 2(y-b) \end{bmatrix}, \tag{8}$$

de modo, que ao igualarmos cada componente a zero, teremos

$$\begin{bmatrix} 2(x-a) \\ 2(y-b) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{9}$$

de onde obtemos o vetor-ótimo como $\mathbf{x}_{opt} = [a \;\; b]^T$.

1.2 Método Iterativo

Em muitas ocasiões não é possível obter uma solução-ótima imediata e fechada para o vetor-ótimo da função de interesse através da técnica mostrada na Eq. (6). Nestes casos, uma solução iterativa, normalmente utilizando uma equação recursiva, surge como a opção mais viável.

Um procedimento muito comum de se chegar iterativamente ao vetor-ótimo $\mathbf{x}_{opt}$ ou a uma vizinhança deste, é conhecido como método do gradiente descendente (ou gradiente estocástico). Para este fim, utiliza-se a seguinte equação recursiva:

$$\mathbf{x}(n+1) = \mathbf{x}(n) - \alpha \frac{\partial f(\mathbf{x}(n))}{\partial \mathbf{x}(n)}, \tag{10}$$

tal que a constante $0 < \alpha \ll 1$ é chamada de *passo de adaptação*. Note que $\mathbf{x}(n)$ denota o valor do vetor-solução $\mathbf{x}$ na iteração n, logo o termo $\frac{\partial f(\mathbf{x}(n))}{\partial \mathbf{x}(n)}$ denota o valor instantâneo do gradiente de $f(\mathbf{x})$ para $\mathbf{x} = \mathbf{x}(n)$. A equação (10) é interpretada da seguinte forma:

> O vetor-solução atual, $\mathbf{x}(n)$, é modificado na direção do vetor gradiente de $f(\cdot)$, em um sentido que percorre uma trajetória de descida - daí, a razão por trás do uso do sinal (-) na Eq. (10) e do termo *descendente* no nome do algoritmo.

Em outras palavras, o vetor-solução, $\mathbf{x}(n)$, vai sendo modificado incrementalmente, na direção de máxima variação de $f(\cdot)$, com sinal negativo porque estamos tratando de um problema de minimização; logo, buscamos por um ponto de mínimo (local ou global). Se o problema for o de encontrar o ponto em que ocorre o máximo valor (pico ou moda) de uma função ou de maximização de uma função-custo, o sinal de menos (-) deve ser trocado por um de mais (+).

À medida que a equação recursiva do método do gradiente é executada, espera-se que ao longo de vários ciclos, a diferença entre $\mathbf{x}(n+1)$ e $\mathbf{x}(n)$ vai diminuindo. Em termos mais formais, à medida que $n \to \infty$, tem-se que $\|\mathbf{x}(n+1) - \mathbf{x}(n)\| \to 0$, em que $\|\mathbf{v}\|$ denota a norma euclidiana do vetor $\mathbf{v}$.

Se um ponto de mínimo (local ou global) for alcançado, teremos $\mathbf{x}(n+1) \approx \mathbf{x}(n)$, de onde resulta que

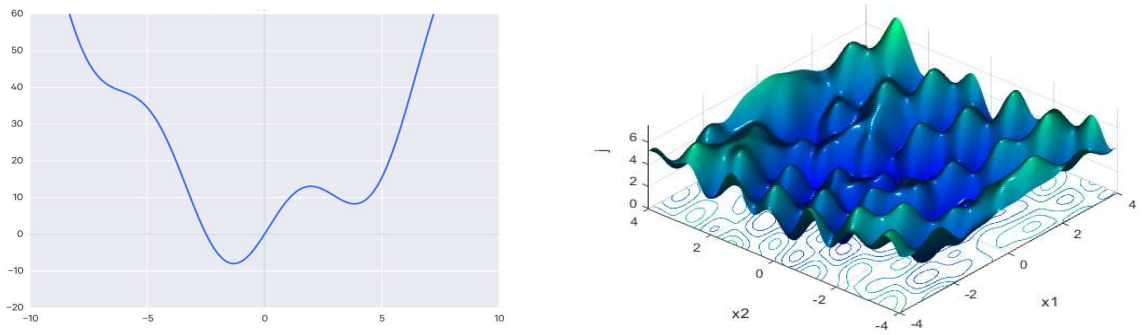
$$\frac{\partial f(\mathbf{x}(n))}{\partial \mathbf{x}(n)} = \mathbf{0}, \tag{11}$$

que é a condição a ser satisfeita por um extremo da função de interesse, conforme explicitado na Eq. (6).

Observação: Como a Eq. (10) é recursiva, faz-se necessário definir um valor inicial $\mathbf{x}(0)$ para o vetor-solução em $n = 0$. Tem-se então duas situações que podem acontecer quando o método do gradiente é utilizado em otimização. São elas:

Função Convexa - Neste caso, a função tem apenas um ponto mínimo, chamado de global. Assim, qualquer que seja o valor atribuído a $\mathbf{x}(0)$, o algoritmo sempre convergirá para a solução ótima após algumas iterações. As funções mostradas na Fig. (1) são exemplos de funções convexas.

Função Não-Convexa - Neste caso, a função tem vários pontos mínimos, chamados de mínimos locais. Logo, o algoritmo convergirá para a solução (ponto de mínimo) mais próximo do vetor inicial $\mathbf{x}(0)$ e não sairá mais deste, uma vez que para qualquer ponto ótimo (seja local ou global) teremos $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_{opt}} = \mathbf{0}$.



(a) (b)

Figura 2: Exemplos de funções não-convexas no plano cartesiano e em 3 dimensões..

1.2.1 Exemplo 2

Tomando como exemplo a função mostrada na Eq. (3), temos que o vetor-gradiente na iteração n para esta função é dado por

$$\begin{bmatrix} \frac{\partial f(x(n),y(n))}{\partial x(n)} \\ \\ \frac{\partial f(x(n),y(n))}{\partial y} \end{bmatrix} = \begin{bmatrix} 2(x(n)-a) \\ \\ 2(y(n)-b) \end{bmatrix}, \tag{12}$$

de modo, que a Eq. (10) para este exemplo passa a ser escrita como

$$\begin{bmatrix} x(n+1) \\ \\ y(n+1) \end{bmatrix} = \begin{bmatrix} x(n) \\ \\ y(n) \end{bmatrix} - 2\alpha \begin{bmatrix} x(n)-a \\ \\ y(n)-b \end{bmatrix}, \tag{13}$$

em que α é o passo de adaptação. Na Fig. 3 estão mostradas as trajetórias $\{x(n)\}_{n=0}^{50}$ e $\{y(n)\}_{n=0}^{50}$ para $a = b = 20$, $c = 50$, α=0,1, $x(0) = 2$ e $y(0) = 25$. Note que as trajetórias convergem para os valores ótimos $x_{opt} = y_{opt} = 20$ após algumas iterações. A convergência pode ser acelerada se escolhermos um passo de adaptação maior (e.g. α=0,25). Código Matlab/Octave para esta simulação está disponível na Fig. (4).

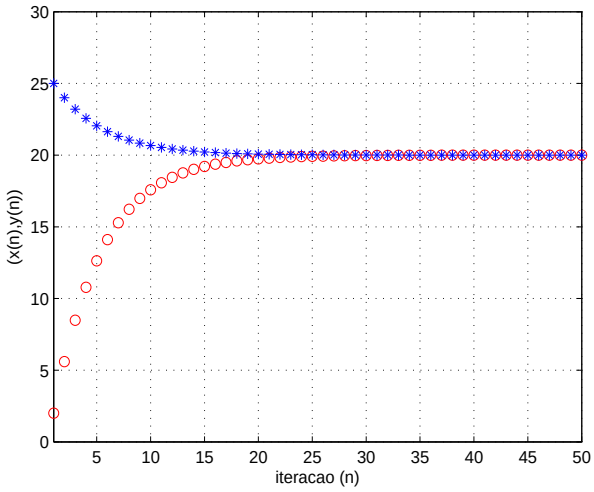


Figura 3: Trajetórias das coordenadas $x(n)$ (círculos vermelhos) e $y(n)$ (asteriscos azuis) para $a = b = 20$, $c = 50$, α=0,1, $x(0) = 2$ e $y(0) = 25$.

```
% Metodo do gradiente descendente (funcao de duas variaveis)
% Funcao: z=f(x,y)=(x-a)^2+(y-b)^2+c
% Vetor-gradiente: dz(v)/dv = [dz(x,y)/dx dz(x,y)/dy], onde v=[x y]^T
% Regra recursiva: v(n+1)=v(n)-eta*dz/dv
% Autor: Guilherme Barreto
% Data:  23/11/2015

clc; clear;

v(:,1)=[2;25];  % vetor de condicoes iniciais

eta=0.1; % Passo de aprendizagem

a=20; b=a; c=50;  % constantes da funcao
for n=1:50,
    gradvec(1,n)=2*(v(1,n)-a);
    gradvec(2,n)=2*(v(2,n)-b);
    v(:,n+1)=v(:,n)-eta*gradvec(:,n);
end

figure;
plot(1:51,v(1,:),'ro'); hold on;
plot(1:51,v(2,:),'b*'); grid;
axis([1 50 0 1.5*a]);
xlabel('iteracao (n)');
ylabel('(x(n),y(n))'); hold off
```

Figura 4: Código Matlab/Octave para Exemplo 2.