FGV EMAp

João Pedro Jerônimo

Otimização para Ciência de Dados

Revisão para A2

Rio de Janeiro

2025

Conteúdo

Introdução

Antes de iniciarmos com o conteúdo de verdade, vou relembrar alguns conceitos importantes da A1 que vão ajudar a entender os métodos presentes nesse PDF.

Teorema 1.1 (Aproximação de Primeira Ordem): Quando f é continuamente diferenciável, em uma vizinhança de um ponto x podemos mostrar que:

$$\forall d \in \mathbb{R}^n \text{ com } \|d\| = 1 \qquad \frac{\mathrm{d}f}{\mathrm{d}d} = \nabla f(x)^T d \tag{1}$$

e, além disso, temos:

$$\forall y \in \mathbb{R}^n \text{ na vizinhança} \qquad f(y) = f(x) + \nabla f(x)^T (y - x) + o(\|y - x\|) \tag{2}$$

Onde $o : \mathbb{R}_+ \to \mathbb{R}$ satisfaz $\lim_{t \to 0^+} \frac{o(t)}{t} = 0$

Teorema 1.2 (Aproximação Linear): Seja $f : U \to \mathbb{R}$ uma função duas vezes continuamente diferenciável e $U \subseteq \mathbb{R}^n$, e seja $x \in U$ e $r > 0$ tais que $B(x, r) \subset U$ então:

$$\begin{gathered} \forall y \in B(x, r)\ \exists \xi \in [x, y] \text{ tal que} \\ f(y) = f(x) + \nabla f(x)^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(\xi)(y - x) \end{gathered} \tag{3}$$

Teorema 1.3 (Aproximação de Segunda Ordem): Seja $f : U \to \mathbb{R}$ uma função duas vezes continuamente diferenciável e $U \subseteq \mathbb{R}^n$, e seja $x \in U$ e $r > 0$ tais que $B(x, r) \subset U$ então:

$$\begin{gathered} \forall y \in B(x, r) \text{ vale} \\ f(y) = f(x) + \nabla f(x)^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x) + o(\|y - x\|^2) \end{gathered} \tag{4}$$

Método do Gradiente

2.1 Caso Global

É o método de otimização mais clássico que existe! Vamos supor que queremos resolver o problema:

$$\min_{x \in \mathbb{R}^n} f(x) \tag{5}$$

Lembra do que vimos em cálculo? Que $\nabla f(x)$ é o vetor que aponta pra direção em que $f(x)$ aumenta? Então que tal a gente seguir na direção contrária a $f(x)$? Isso faz bastante sentido, e funciona! Mas deve ter um motivo mais matemático por trás, não é? Vamos primeiro mostrar o algoritmo:

```
func GradientDescent(f) {
    x^(0) ∈ ℝ^n
    α > 0
    for t ∈ [T] do {
        x^(t+1) = x^(t) − α∇f(x^(t))
    }
    return x^(T)
}
```

Algoritmo 1: Gradient Descent

Antes de entender um motivo mais matemático por trás do algoritmo, vamos ver algumas definições

Definição 2.1.1 (Funções M-Lipschitz): Dizemos que uma função $f : \mathbb{R}^n \to \mathbb{R}^m$ é M-Lipschitz quando:

$$\|f(x) - f(y)\| \leq M \ \|x - y\| \qquad \forall x, y \in \mathbb{R}^n \tag{6}$$

Definição 2.1.2 (Função L-Suave): Uma função $f : \mathbb{R}^n \to \mathbb{R}$ é L-suave quando seu gradiente é L-Lipschitz:

$$\|\nabla f(x) - \nabla f(y)\| \leq L \ \|x - y\| \qquad \forall x, y \in \mathbb{R}^n \tag{7}$$

Essa definição de suavidade tem uma interpretação, imagine que, se eu estou na posição x e vou pra posição y, a variação que eu vou ter na função, dentro dessa passada, não ultrapassa o quanto eu andei vezes uma constante L. Então funções muito onduladas, e com ondulações

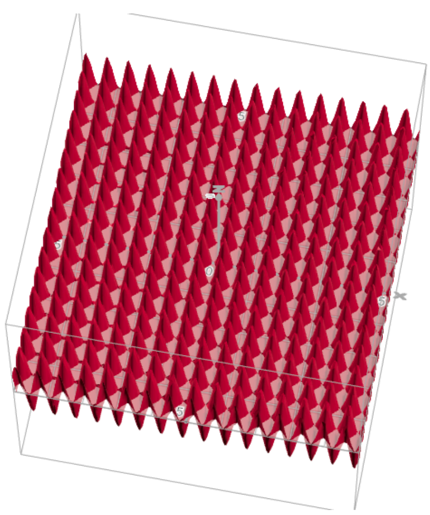

Figura 2: Função bem desregular, mas suave, $f(x) = \sin(10x) + \cos(10y)$

Teorema 2.1.1 (Aproximação linear de funções suaves): Seja $f : \mathbb{R}^n \to \mathbb{R}$ uma função diferenciável. Então f é L-suave se, e somente se, $\forall x, y \in \mathbb{R}^n$:

$$|f(y) - f(x) + \nabla f(x)^T (y - x)| \leq \frac{L}{2} \|y - x\|_2^2 \tag{8}$$

Usando esse teorema, a gente pode escrever isso:

$$\begin{aligned} f\left(x^{(t+1)}\right) &\leq f\left(x^{(t)}\right) + \nabla f\left(x^{(t)}\right)^T \left(x^{(t+1)} - x^{(t)}\right) + \frac{L}{2} \|x^{(t+1)} - x^{(t)}\|^2 \\ &\leq f\left(x^{(t)}\right) - \alpha \|\nabla f\left(x^{(t)}\right)\|^2 + \frac{\alpha^2 L}{2} \|\nabla f\left(x^{(t)}\right)\|^2 \\ &\leq f\left(x^{(t)}\right) - \alpha \left(1 - \frac{\alpha L}{2}\right) \|\nabla f\left(x^{(t)}\right)\|^2 \end{aligned} \tag{9}$$

E isso vale quando $\alpha \in \left(0, \frac{2}{L}\right)$, ou seja, se o ponto atual $x^{(t)}$ **NÃO É ESTACIONÁRIO**, o valor da função no próximo ponto será **menor** que o valor do ponto atual menos o tamanho do gradiente ao quadrado vezes um termo de regulação. Parece complicado, mas o que isso quer dizer? Eu vou usar esse fato para mostrar que, independente do ponto que eu iniciar o método do gradiente, eu **sempre vou encontrar um ponto mínimo local utilizando o método do gradiente**

Teorema 2.1.2 (Convergência do Gradient Descent): Suponha que $f : \mathbb{R}^n \to \mathbb{R}$ é L-suave. Tome qualquer passo:

$$\alpha = \frac{\beta}{L} \tag{10}$$

para algum $\beta \in (0, 2)$. Então:

$$\min_{t \in [T]} \|\nabla f\left(x^{(t)}\right)\|_2^2 \leq \frac{1}{T} \sum_{t=1}^{T} \|\nabla f\left(x^{(t)}\right)\|_2^2 \leq \left(\frac{2/\beta}{2 - \beta}\right) \frac{L\left(f\left(x^{(1)}\right) - f^*\right)}{T} \tag{11}$$

Demonstração: Sabemos que, dado n pontos x_i, a média $\frac{1}{n} \sum_{i=1}^{n} x_i \in [\min(x_i), \max(x_i)]$, então a desigualdade inicial já está provada. Vamos provar a segunda. Usando a equação (9), temos que:

$$\alpha \left(1 - \frac{\alpha L}{2}\right) \|\nabla f\left(x^{(t)}\right)\|^2 \leq f\left(x^{(t)}\right) - f\left(x^{(t+1)}\right) \tag{12}$$

isso para todo $t \in [T]$, então vamos somar todos os termos para obter:

$$\begin{aligned} \alpha \left(1 - \frac{\alpha L}{2}\right) \sum_{t=1}^{T} \|\nabla f\left(x^{(t)}\right)\|_2^2 &\leq \sum_{t=1}^{T} \left(f\left(x^{(t)}\right) - f\left(x^{(t+1)}\right)\right) \\ &\leq f\left(x^{(1)}\right) - f\left(x^{(T+1)}\right) \\ &= f\left(x^{(1)}\right) - f^* + f^* + f\left(x^{(T+1)}\right) \\ &\leq f\left(x^{(1)}\right) - f^* \end{aligned} \tag{13}$$

A primeria desigualdade eu fiz uma soma telescópica, depois eu somei 0 ($f^* - f^*$) e, como f^* é o valor mínimo da função, com certeza subtrair a parte que eu somei f^* vai dar um valor maior, então eu obtenho o resultado do enunciado do teorema dividindo tudo por $\alpha\left(1 - \frac{\alpha L}{2}\right)T$ □

Por que esse teorema mostra que, independentemente do lugar, o algoritmo converge para um ponto estacionário? Ele ta me dizendo isso daqui:

$$\min_{t\in[T]} \|\nabla f(x^{(t)})\|_2^2 = O\left(\frac{1}{T}\right) \tag{14}$$

Ou seja, o mínimo **converge para 0** conforme $T \to \infty$ **independentemente do ponto inicial** $x^{(0)}$

Só que se pararmos para pensar, se $\nabla f(x)$ é uma direção de subida, então $-\nabla f(x)$ é de descida, mas será que é a melhor? Será que **precisa** ser a melhor para o método funcionar? Se eu escolher **uma direção de descida arbitrária** ele funciona? Mas antes disso, vamos entender o que é uma direção de descida

Definição 2.1.3 (Direção de Descida): Dizemos que $d \in \mathbb{R}^n$ é uma direção de descida para a funçaõ $f : \mathbb{R}^n \to \mathbb{R}$ no ponto $x \in \mathbb{R}^n$ quando:

$$d^T \nabla f(x) < 0 \tag{15}$$

então podemos considerar um novo algoritmo

func GradientDescentWithArbitraryDescentDirection(f) {
  $x^{(0)} \in \mathbb{R}^n$
  $\alpha > 0$
  for $t \in [T]$ **do** {
    Encontrar **uma** direção de descida $d^{(t)}$
    $x^{(t+1)} = x^{(t)} + \alpha d^{(t)}$
  }
  return $x^{(T)}$
}

Algoritmo 3: Gradient Descent com Direção de Descida Arbitrária

Podemos provar, analogamente ao Teorema 2.1.2, que o algoritmo converge para um ponto estacionário

2.2 Caso Convexo

Antes, não assumimos nada além da suavidade da função, agora vamos mostrar que, assumindo que f é convexa, o algoritmo converge para uma solução global. Primeiro, nós sabemos que:

$$\begin{aligned} x^{(t+1)} &= x^{(t)} - \alpha \nabla f(x^{(t)}) \\ \Leftrightarrow x^{(t+1)} - x^* &= x^{(t)} - x^* - \alpha \nabla f(x^{(t)}) \\ \Leftrightarrow \|x^{(t+1)} - x^*\|_2^2 &= \|x^{(t)} - x^* - \alpha \nabla f(x^{(t)})\|_2^2 \end{aligned} \tag{16}$$

Pelas propriedades da convexidade:

$$\left(x^* - x^{(t)}\right)^T \nabla f(x^{(t)}) \leq f(x^*) - f(x^{(t)}) \tag{17}$$

e pelo que vimos na equação (9), se $\alpha \in \left(0, \frac{2}{L}\right)$, podemos chegar que:

$$\|x^{(t+1)} - x^*\|_2^2 \leq \|x^{(t)} - x^*\|_2^2 - \alpha\left(2 - \frac{1}{1 - \frac{\alpha L}{2}}\right)\left(f(x^{(t)}) - f^*\right) \tag{18}$$

ou seja, a distância do próximo iterado pro ponto ótimo é menor a distância atual, menos um termo proporcional a distância dos resultados de $x^{(t)}$ e do ponto ótimo. Vamos usar isso para provar a convergência global do resultado

Teorema 2.2.1 (Convergência Convexa do Método do Gradiente): Suponha que $f : \mathbb{R}^n \to \mathbb{R}$ é L-suave e convexa e tome qualquer passo

$$\alpha = \frac{\beta}{L} \tag{19}$$

para algum $\beta \in (0, 1)$, então:

$$f\left(x^{(T)}\right) - f^* \leq \frac{1}{T} \sum_{t=1}^{T} \left(f\left(x^t\right) - f^*\right) \leq \frac{\beta^{-1} - \frac{1}{2}}{1 - \beta} \frac{L\|x^{(1)} - x^*\|_2^2}{T} \tag{20}$$

Demonstração: Somando-se em $tin[T]$ a recorrência:

$$\alpha(2 - \frac{1}{1 - \alpha\frac{L}{2}}\left(f\left(x^t\right) - f^*\right) \leq \|x^{(t)} - x^*\|_2^2 - \|x^{(t+1)} - x^*\|_2^2 \tag{21}$$

obtemos novamente uma soma telescópica:

$$\begin{aligned} \alpha\left(\frac{1 - \alpha L}{1 - \alpha\frac{L}{2}}\right) \sum_{t=1}^{T} \left(f\left(x^{(t)}\right) - f^*\right) &\leq \sum_{t=1}^{T} \left(\|x^{(t)} - x^*\|_2^2 - \|x^{(t+1)} - x^*\|_2^2\right) \\ &\leq \|x^{(1)} - x^*\|_2^2 - \|x^{(T+1)} - x^*\|_2^2 \\ &\leq \|x^1 - x^*\|_2^2. \end{aligned} \tag{22}$$

Dividindo-se por $\alpha\left(\frac{1-\alpha L}{1-\alpha\frac{L}{2}}\right)T$, e lembrando que, pelo , a sequência $\left\{f\left(x^{(t)}\right)\right\}$ é decrescente:

$$f\left(x^{(T)}\right) - f^* \leq \frac{1}{T} \sum_{\{t=1\}}^{T} \left(f\left(x^{(t)}\right) - f^*\right) \leq \frac{\alpha^{-1}\left(1 - \alpha\frac{L}{2}\right)}{T} \ \|x^1 - x^*\|_2^2. \tag{23}$$

□

2.3 Interpretação via regularização

Outra formas que podemos ver e interpretar o algoritmo do gradiente é resolver a seguinte fórmula:

$$x^{(t+1)} = \operatorname{argmin}_{x \in \mathbb{R}^n} \left(\underbrace{f\left(x^{(t)}\right) + \left(x - x^{(t)}\right)^T \nabla f\left(x^{(t)}\right)}_{\text{Aproximação Linear}} + \underbrace{\frac{1}{2\alpha^{(t)}} \|x - x^{(t)}\|_2^2}_{\text{Regularização Proximal}} \right) \tag{24}$$

Ou seja, eu vou pegar qual que é o valor que minimiza a aproximação linear regularizada por um termo quadrático

Método do Subgradiente

Até agora vimos um método que usa o Gradiente da função, logo, para que ele funcione, estamos assumindo que a função é diferenciável em todos os pontos, porém em aplicações reais muitas funções não são diferenciáveis em todos os pontos. Então faz sentido utilizar esse método? Claro que não, porém, podemos utilizar uma versão muito parecida!

Definição 3.1 (Subgradiente): Seja uma função $f : \mathbb{R}^n \to \mathbb{R}$ e dado $x \in \mathbb{R}^n$, um vetor $g_x \in \mathbb{R}^n$ é chamado de **subgradiente** de f em x quando

$$f(y) \geq f(x) + (y - x)^T g_x \qquad \forall y \in \mathbb{R}^n \tag{25}$$

Definição 3.2 (Subdiferencial): O conjunto de todos os subgradientes de f em x é chamado de **subdiferencial** de f em x (Denotado por $\partial f(x)$)

$$\partial f(x) := \{g_x \in \mathbb{R}^n : f(y) \geq f(x) + (y - x)^T g_x \ \forall y \in \mathbb{R}^n\} \tag{26}$$

O que seria um subgradiente **intuitivamente** então? Note que, se eu definir $y = x^*$ (Sendo o ponto mínimo), vamos obter o seguinte:

$$f(x^*) \geq f(x) + g^T(x^* - x) \tag{27}$$

sabendo que $f(x^*) \leq f(x)$, temos que:

$$f(x) \geq f(x) + g^T(x^* - x) \Leftrightarrow 0 \geq g^T(x^* - x) \Leftrightarrow 0 \leq g^T(x - x^*) \tag{28}$$

Ou seja, o subgradiente faz um ângulo **maior que 90º** com o vetor que aponta de x para x^*. O que isso quer dizer? Que os subgradientes são direções que, se seguirmos na **direção oposta**, nós **não** estamos indo eu uma direção em que nós temos **certeza** que ela sobe

Nem sempre existirão subgradientes, porém, quando falamos das **funções convexas**, mesmo elas não sendo **diferenciáveis**, elas possuem um subgradiente. Logo, um subgradiente é uma direção que, se eu vou na direção contrária a ela, eu tenho **certeza** que minha função não está aumentando

Teorema 3.1: Seja $f : \mathbb{R}^n \to \mathbb{R}$, então vale que:

$$f \text{ é convexa} \Leftrightarrow \partial f(x) \neq \emptyset \ \forall x \tag{29}$$

Vale ressaltar que funções subdiferenciáveis podem não ser suaves. E por que isso é importante? Acontece que antes, no método do gradiente e na direções de descida, utilizamos o fato das funções serem suaves para provar a convergência do método, porém, aqui estamos trabalhando com funções que não necessariamente são diferenciáveis, logo, intuitivamente, elas podem ter vários picos, ou paradas bruscas, etc.

func SubgradientMethod(f) {
  $x^{(1)} \in \mathbb{R}^n$
  $\{\alpha^{(t)}\} \subset (0, \infty)$
  for $t \in [T]$ **do** {
    Compute um subgradiente $g^{(t)}$ de f em $x^{(t)}$
    $x^{(t+1)} = x^{(t)} - \alpha^{(t)} g^{(t)}$
  }
  return $x^{(T)} := \sum_{t=1}^{T} \frac{\alpha^{(t)}}{\sum_{l=1}^{T} \alpha^{(l)}} x^{(t)}$
}

Algoritmo 4: Método do Subgradiente

Esse algoritmo parece até que “ingênuo”, tipo, nada garante que o subgradiente vai fazer com que a função desça né?? Vamos mostrar que na verdade esse método converge sim! Porém, vamos assumir algumas coisas também. Para esse caso, vamos assumir que a função é M-Lipschitz (Definição 2.1.1)

Usando o que foi mostrado na introdução (Teorema 1.2), podemos mostrar o seguinte teorema:

Teorema 3.2 (Aproximação Linear de funções M-Lipschitz): Seja $f : \mathbb{R}^n \to \mathbb{R}$ uma função convexa, então f é M-Lipschitz contínua se, e somente se, $\forall x, y \in \mathbb{R}^n$ e todo subgradiente $g_x \in \mathbb{R}^n$ de f em x:

$$|f(y) - f(x) + g_x^T(y - x)| \leq M \, \|y - x\|_2^2 \tag{30}$$

E no que isso me é útil? Só parece um bando de complicação esquisita. Na verdade, o que será útil na demonstração é uma **consequência** desse teorema

Corolário 3.2.1: Seja $f : \mathbb{R}^n \to \mathbb{R}$ uma função convexa, então f é M-Lipschitz se, e somente se, $\forall x \in \mathbb{R}^n$ e todo subgradiente g_x de f em x, vale que $\|g_x\|_2 \leq M$

Vamos então mostrar a convergência do algoritmo. Um ponto interessante a se dizer é que, como você **talvez** possa ter pensado, nem sempre um subgradiente vai ser uma direção de **descida**. O que isso quer dizer? Isso quer dizer que, não necessariamente, a cada iteração, $f\left(x^{(t+1)}\right) \leq f\left(x^{(t)}\right)$, porém, ele decresce o valor sobre a **média de todas iterações**.

Teorema 3.3 (Convergência do Subgradiente): Suponha que $f : \mathbb{R}^n \to \mathbb{R}$ é M-Lipschitz contínua e convexa. Então:

$$f\left(\overline{x}^{(T)}\right) - f^* \leq \frac{1}{T} \sum_{t=1}^{T} \left(f\left(x^{(t)}\right) - f^*\right) \leq \frac{\|x^{(1)} - x^*\|_2^2 + M^2 \sum_{t=1}^{T} \left(\alpha^{(t)}\right)^2}{\sum_{t=1}^{T} \alpha^{(t)}} \tag{31}$$

em particular, para qualquer $\beta > 0$, consideramos:

$$\alpha^{(t)} = \frac{\beta}{\sqrt{T}} \qquad t \in [T] \tag{32}$$

Demonstração: Primeiramente, temos que:

$$\begin{aligned} \|x^{(t+1)} - x^*\|_2^2 &= \|x^{(t)} - x^* - \alpha^{(t)} g^{(t)}\|_2^2 \\ &\leq \|x^{(t)} - x^*\|_2^2 + 2\alpha^{(t)}(x - x^*)^T g^{(t)} + \left(\alpha^{(t)}\right)^2 \, \|g^{(t)}\|_2^2 \end{aligned} \tag{33}$$

e pela definição de subgradiente (f, por ser convexa, é garantida de ter subgradientes pelo Teorema 3.1), temos que:

$$\left(x^* - x^{(t)}\right)^T g^{(t)} \leq f(x^*) - f\left(x^{(t)}\right) \tag{34}$$

A partir disso, também podemos escrever:

$$\|x^{(t+1)} - x^*\|_2^2 \leq \|x^{(t)} - x^*\|_2^2 - \alpha^{(t)}\left(f\left(x^{(t)}\right) - f^*\right) + \left(M\alpha^{(t)}\right)^2 \tag{35}$$

Agora nós vamos fazer novamente a soma por recursão:

$$\begin{aligned} \sum_{t=1}^{T} \alpha^{(t)}\left(f\left(x^{(t)}\right) - f^*\right) &\leq \sum_{t=1}^{T} \left(\|x^{(t)} - x^*\|_2^2 - \|x^{(t+1)} - x^*\|_2^2\right) + M^2 \sum_{t=1}^{T} \left(\alpha^{(t)}\right)^2 \\ &\leq \|x^{(1)} - x^*\|_2^2 + M^2 \sum_{t=1}^{T} \left(\alpha^{(t)}\right)^2 \end{aligned} \tag{36}$$

Dividindo por $\sum_{t=1}^{T} \alpha^{(t)}$ e, usando a desigualdade de Jensen

$$f\left(\overline{x}^{(T)}\right) - f^* \leq \frac{1}{T}\sum_{t=1}^{T}\left(f\left(x^{(t)}\right) - f^*\right) \leq \frac{\|x^{(1)} - x^*\|_2^2 + M^2 \sum_{t=1}^{T}\left(\alpha^{(t)}\right)^2}{\sum_{t=1}^{T}\alpha^{(t)}} \tag{37}$$

Ou seja, a média das iterações converge para próximo de x^* □

Nós assumimos que $\alpha^{(t)}$ está em um conjunto de passos, e não que é um único passo, mas e se assumirmos que é um único, qual seria o melhor passo? Tomando $\alpha^{(t)} = \alpha$:

$$f\left(\overline{x}^{(T)}\right) - f^* \leq \frac{\|x^{(1)} - x^*\|_2^2}{T\alpha} + M^2\alpha \tag{38}$$

perceba também que:

$$\min_{\alpha > 0}\left\{\frac{\|x^{(1)} - x^*\|_2^2}{T\alpha} + M^2\alpha\right\} = \frac{M\ \|x^{(1)} - x^*\|_2}{\sqrt{T}} \tag{39}$$

logo, temos que:

$$\alpha^{(t)} = \frac{\|x^{(1)} - x^*\|_2}{M\sqrt{T}} \tag{40}$$

então teremos a taxa de convergência:

$$f\left(\overline{x}^{(T)}\right) - f^* \leq \frac{2M\ \|x^{(1)} - x^*\|_2}{\sqrt{T}} \tag{41}$$

Porém, na maioria esmagadora das vezes, não sabemos M e $\|x^{(1)} - x^*\|$, então utilizamos o passo sequencial já definido anteriormente.

Como vimos antes, também conseguimos uma fórmula recursiva para o método do subgradiente, assim como para o do gradiente

$$x^{(t+1)} = \operatorname{argmin}_{x \in \mathbb{R}^n}\left\{f\left(x^{(t)}\right) + < x - x^{(t)}, g^{(t)} > + 1/\left(2\alpha^{(t)}\right) \cdot \|x - x^{(t)}\|_2^2\right\} \tag{42}$$

Gradiente Projetado

Certo, perceba que, até o momento, nós utilizamos algoritmos aplicados apenas em funções em TODO o plano $\mathbb{R}^n$, mas e se, como em muitos casos, temos um conjunto limitado? Agora vamos considerar o seguinte problema de otimização

$$\min_{x \in C} f(x) \tag{43}$$

para funções $f : \mathbb{R}^n \to \mathbb{R}$ e o conjunto viável $C \subset \mathbb{R}^n$ **convexo** e vamos assumir também que f tem gradientes ou pelo menos subgradientes em qualquer ponto. A gente viu a forma recursiva nos problemas anteriores, que tal a gente tentar aplicar aqui? Só que em vez de fazer no $\mathbb{R}^n$, nós fazemos em C?

$$x^{(t+1)} = \mathrm{argmin}_{x \in C} \left\{ f\left(x^{(t)}\right) + < x - x^{(t)}, g^{(t)} > + \frac{1}{2\alpha^{(t)}} \ \|x - x^{(t)}\|_2^2 \right\} \tag{44}$$

Podemos mostrar que essa fórmula é equivalente a

$$x^{(t+1)} = \mathrm{argmin}_{x \in C} \ \|x - \left(x^{(t)} - \alpha^{(t)} g^{(t)}\right)\|_2^2 \tag{45}$$

Só que perceba que isso é equivalente a **projetar ortogonalmente** $x^{(t)} - \alpha^{(t)} g^{(t)}$ em C, então concluímos que:

$$x^{(t+1)} = \Pi_{x \in C} \left[x^{(t)} - \alpha^{(t)} g^{(t)}\right] \tag{46}$$

de forma que Π é o projetor ortogonal a C. Então temos o seguinte algoritmo:

func SubgradientMethod(f) {
    $x^{(1)} \in \mathbb{R}^n$
    $\left\{\alpha^{(t)}\right\} \subset (0, \infty)$
    for $t \in [T]$ **do** {
        Compute um subgradiente $g^{(t)}$ de f em $x^{(t)}$
        $z^{(t+1)} = x^{(t)} - \alpha^{(t)} g^{(t)}$
        $x^{(t+1)} = \Pi_{x \in C} \left[z^{(t+1)}\right]$
    }
    return $x^{(T)} := \frac{\sum_{t=1}^{T} \alpha^{(t)} x^{(t)}}{\sum_{l=1}^{T} \alpha^{(l)}}$
}

Algoritmo 5: Método do Subgradiente Projetado

Gradiente Proximal

Esses métodos resolvem naturalmente algumas limitações do método projetado. Um dos principais problemas é que, para conjuntos não triviais, calcular as projeções pode ser **muito** custoso, então o que fazer? Algo muito comum, é adicionar uma função de custo, que **penaliza** conforme a resposta de **afasta** do conjunto viável C. Então vamos considerar o novo problema:

$$\min_{x \in \mathbb{R}^n} f(x) + g(x) \tag{47}$$

para funções $f : \mathbb{R}^n \to \mathbb{R}$, onde f possui gradientes ou pelo menos subgradientes em todos os pontos e $g : \mathbb{R}^n \to \mathbb{R}$ convexa. Nós poderíamos tentar aplicar o método do subgradiente diretamente à $f + g$, só que essa função pode ser mais complexa e nem ser diferenciável, então utilizamos uma família de métodos onde mantemos g preservada e usamos subgradientes apenas de f

$$x^{(t+1)} = \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ f\left(x^{(t)}\right) + < g^{(t)}, x - x^{(t)} > + g(x) + \frac{1}{2\alpha^{(t)}} \ \|x - x^{(t)}\|_2^2 \right\} \tag{48}$$

onde $g^{(t)}$ é o subgradiente de f no ponto corrente $x^{(t)}$. Perceba que, ao adicionar $g(x)$ dentro do argmin, ele acaba por regularizar a função **dependendo** da escolha de $g(x)$

Exemplo (Restrição por Penalização): Uma penalização muito comum é transformar as condições de restrições de um conjunto viável em uma função, de forma que:

$$I_C(x) = \begin{cases} 0 & x \in C \\ +\infty & x \notin C \end{cases} \tag{49}$$

De forma que os problemas:

$$\min_{x \in C} f(x) \qquad \min_{x \in \mathbb{R}^n} f(x) + I_C(x) \tag{50}$$

são equivalentes

Podemos fazer uma definição útil para expressar modularmente o algoritmo descrito anteriormente

Definição 5.1 (Operador Proximal): O operador proximal de uma função **convexa** g é o operador que, para cada $x \in \mathbb{R}^n$, associa o vetor:

$$\operatorname{prox}_g(x) := \operatorname{argmin}_{y \in \mathbb{R}^n} \left\{ g(y) \frac{1}{2} \|y - x\|_2^2 \right\} \tag{51}$$

Com essa definição, é fácil ver que a equação (48) é equivalente a:

$$x^{(t+1)} = \operatorname{prox}_g\left(x^{(t)} - \alpha^{(t)} g^{(t)}\right) \tag{52}$$

func ProximalGradientMethod($f = g + r$) {
  $x^{\{(1)\}} \in \mathbb{R}^n$
  $\left\{\alpha^{(t)}\right\} \subset (0, \infty)$
  for $t \in [T]$ **do** {
    Compute $\nabla g\left(x^{(t)}\right)$
    $z^{(t+1)} = x^{(t)} - \alpha^{(t)} \nabla g\left(x^{(t)}\right)$
    $x^{(t+1)} = \operatorname{prox}_{\alpha^{(t)} r}\left(z^{(t+1)}\right)$
  }
  return $\overline{x}^{(T)} := \frac{\sum_{t=1}^{T} \alpha^{(t)} x^{(t)}}{\sum_{t=1}^{T} \alpha^{(t)}}$
}

Algoritmo 6: Método do Gradiente Proximal

Método de Newton

Até agora, vimos métodos que utilizam de aproximações de primeira ordem das funções, porém, e se tentarmos utilizar mais informações além dessas? E se a função que estamos trabalhando for diferenciável duas vezes? Será que não poderiamos usar sua **Hessiana** para auxiliar? Lembra no primeiro resumo que falamos, inutitivamente, como a Hessiana carrega informações sobre para quais lados a função cresce e decresce? Poderíamos tentar utilizar essas informações! Vamos tentar aplicar uma fórmula recursiva igual fizemos no último?

$$x^{(t+1)} = \text{argmin}_{x\in\mathbb{R}^n}\left\{f\left(x^{(t)}\right) + \left(x - x^{(t)}\right)^T \nabla f\left(x^{(t)}\right) + \frac{1}{2}\left(x - x^{(t)}\right)^T \nabla^2 f\left(x^{(t)}\right)\left(x - x^{(t)}\right)\right\} \tag{53}$$

onde aqui utilizamos o Teorema 1.3. Aqui, estamos assumindo algumas coisas:
- $\nabla^2 f(x) \succ 0 \ \forall x$
- $\nabla^2 f(x)$ é L-Lipschitz, ou seja: $\|\nabla^2 f(x) - \nabla^2 f(y)\| < L \ \|x - y\| \ \ \forall x, y$

Como resultado dessa operação, obtemos:

$$x^{(t+1)} = x^{(t)} - \alpha^{(t)}\left(\nabla^2 f\left(x^{(t)}\right)\right)^{-1} \nabla f\left(x^{(t)}\right) \tag{54}$$

Assim, temos:

```
func NewtonMethod(f) {
    x^{(1)} ∈ ℝ^n
    α^(t) ∈ ℝ
    for t ∈ [T] do {
        x^(t+1) = x^(t) − α^(t) (∇²f(x^(t)))^(−1) ∇f(x^(t))
    }
    return x^(T)
}
```

Algoritmo 7: Método de Newton

Por conta da "adição" de informações ao método convencional, esse método costuma convergir **MUITO** mais rápido que os já vistos anteriormente. Porém, ele não tem uma garantia **global** de convergência. Como assim? Os métodos anteriores, independente de qual fosse o ponto inicial $x^{(0)}$, convergiam para uma solução local, porém, dependendo do ponto que iniciarmos o método de newton, ele pode **divergir**.

Gradiente Conjugado

Esse método é muito útil pois **garante a convergência globalmente em uma quantidade limitada de passos**. Porém, ele só pode ser aplicado no seguintes problemas:

$$\min_{x \in \mathbb{R}^n} c + b^T x + x^T A x \tag{55}$$

Porém, antes de aplicarmos esse método, temos que fazer uma definição:

Definição 7.1 (Direção conjugada): Dizemos que os vetores $\{d_1, ..., d_n\} \subseteq \mathbb{R}^n$ são direções conjugadas de A se:

$$d_i^T \left(A d_j\right) = 0 \qquad \forall i, j = 1, ..., n \tag{56}$$

Então vamos tentar, ingenuamente, aplicar o método:

$$x^{(t+1)} = x^{(t)} - \alpha d^{(t)} \tag{57}$$

de forma que d_t é uma direção conjugada de A. Que tal tentarmos encontrar o melhor passo α? Então temos que resolver:

$$\begin{aligned} & \min_{\alpha \in \mathbb{R}} f\left(x^{(t)} - \alpha d^{(t)}\right) \\ = & \min_{\alpha \in \mathbb{R}} \left\{ c + b^T \left(x^{(t)} - \alpha d^{(t)}\right) + \left(x^{(t)} - \alpha d^{(t)}\right)^T A \left(x^{(t)} - \alpha d^{(t)}\right) \right\} \end{aligned} \tag{58}$$

Resolvendo esse problema, obtemos:

$$\alpha^{(t)} = -\frac{\nabla f\left(x^{(t)}\right) d^{(t)}}{< d^{(t)}, A d^{(t)} >} \tag{59}$$

Também podemos mostrar que, tomando esse passo, o algoritmo irá convergir **exatamente** para o mínimo em apenas n iterações

Dualidade