GD $\rightarrow$ 0 порядок $f(x_0)$

CG $\rightarrow$ 1 порядок $f(x_0), \nabla f(x_0)$

квазиньютон. $\rightarrow$ 2 порядок $f(x_0), \nabla f(x_0), \nabla^2 f(x_0)$

метод Ньютона

Теор. база

1. Обзор.
2. Дифференцирование $\nabla f, \nabla^2 f$, Jacobian
3. Выпуклость

Gradient descent

$\nabla f(x_k)$ – направление наискорейш. локального роста функции

Summary

$-\nabla f(x)$

A classical problem of function minimization is considered.

$$\boxed{x_{k+1} = x_k - \eta_k \nabla f(x_k)} \tag{GD}$$

learning rate

- The bottleneck (for almost all gradient methods) is choosing step-size, which can lead to the dramatic difference in method's behavior.
- One of the theoretical suggestions: choosing stepsize inversly proportional to the gradient Lipschitz constant $\eta_k = \frac{1}{L}$

$\|\nabla f(x) - \nabla f(y)\|_{2} \leq L \cdot \|x-y\|_{2}$ $\overset{\infty}{1}$: $\frac{\|\nabla f(x) - \nabla f(y)\|}{\|x-y\|} \leq L$

- In huge-scale applications the cost of iteration is usually defined by the cost of gradient calculation (at least $\mathcal{O}(p)$)
- If function has Lipschitz-continious gradient, then method could be rewritten as follows:

$$
\begin{aligned}
x_{k+1} &= x_k - \frac{1}{L} \nabla f\left(x_k\right) = \\
&= \arg\min_{x \in \mathbb{R}^n} \left\{ f\left(x_k\right) + \left\langle \nabla f\left(x_k\right), x - x_k \right\rangle + \frac{L}{2} \|x - x_k\|_2^2 \right\}
\end{aligned}
$$

$\eta = \eta_k = \frac{1}{L}$ $\rightarrow$ $\|\nabla^2 f(x)\| \leq L$

$f(x) = \|Ax - b\|^2$

$\nabla^2 f = A^T A$

Intuition

Direction of local steepest descent

Let's consider a linear approximation of the differentiable function f along some direction $h, \|h\|_2 = 1$:

$$f(x + \eta h) = f(x) + \eta \langle f'(x), h \rangle + o(\eta)$$

We want h to be a decreasing direction:

$$f(x + \eta h) < f(x)$$

$$f(x) + \eta \langle f'(x), h \rangle + o(\eta) < f(x)$$

and going to the limit at $\eta \rightarrow 0$:

$$\langle f'(x), h \rangle \leq 0$$

Also from Cauchy–Bunyakovsky–Schwarz inequality:

$$|\langle f'(x), h \rangle| \leq \|f'(x)\|_2 \|h\|_2 \quad \rightarrow \quad \langle f'(x), h \rangle \geq -\|f'(x)\|_2 \|h\|_2 = -\|f'(x)\|_2$$

Thus, the direction of the antigradient

$$h = -\frac{f'(x)}{\|f'(x)\|_2}$$

gives the direction of the **steepest local** decreasing of the function f.

The result of this method is

$$x_{k+1} = x_k - \eta f'(x_k)$$

Gradient flow ODE

Let's consider the following ODE, which is referred as Gradient Flow equation.

$$\frac{dx}{dt} = -f'(x(t)) \tag{GF}$$

and discretize it on a uniform grid with η step:

$$\frac{x_{k+1} - x_k}{\eta} = -f'(x_k),$$

where $x_k \equiv x(t_k)$ and $\eta = t_{k+1} - t_k$ - is the grid step.

From here we get the expression for x_{k+1}

$$x_{k+1} = x_k - \eta f'(x_k),$$

which is exactly gradient descent.

$$P(\omega) = \sum_{i=1}^{N} (x_i - w)^2 \rightarrow \min_w$$

$$\frac{\partial P}{\partial w} = \sum_{i=1}^{N} \frac{\partial}{\partial w} (x_i - w)^2 \ldots\ldots = 0$$

$$\Rightarrow \sum_{i=1}^{N} \cancel{2}(x_i - w)\cdot(-\cancel{2}) = 0$$

$$\sum_{i=1}^{N} x_i - \sum_{i=1}^{N} w = 0$$

$$\sum_{i=1}^{N} w = \sum_{i=1}^{N} x_i$$

$$Nw = \sum x_i$$

$$\boxed{w = \frac{\sum x_i}{N}}$$

$$\frac{\partial x_i}{\partial w} = 0 \qquad \frac{\partial w}{\partial w} = 1 \qquad \frac{\partial (w)^2}{\partial w} = 2\cdot w \qquad \frac{\partial(-w)}{\partial w} = -1 \qquad \frac{\partial (x_i - w)}{\partial w} = -1$$

Necessary local minimum condition

$$\begin{aligned}
& f'(x) = 0 \\
& -\eta f'(x) = 0 \\
& x - \eta f'(x) = x \\
& x_k - \eta f'(x_k) = x_{k+1}
\end{aligned}$$

This is, surely, not a proof at all, but some kind of intuitive explanation.

$$\min f(x)$$

$$f'(x) = 0 \qquad f'(x) = \sum_{i=1}^{n} 2(w - x_i)$$

$$\eta = \frac{1}{L} \;\cancel{\approx}\; \frac{1}{2}$$

$$f''(x) = 2$$

$$x_{k+1} = x_k - \eta \nabla f(x_k)$$

$$\boxed{f(x_{k+1}) < f(x_k)}$$

$$(x_{k+1})^2 < (x_k)^2$$

$$(x_k - \eta \cdot 2x_k)^2 < x_k^2$$

$$\cancel{x_k^2} - 2x_k^2 \cdot 2\eta + 4\eta^2 x_k^2 < \cancel{x_k^2}$$

$$\eta^2 \cancel{x_k^2} - \eta \cancel{x_k^2} < 0 \;\Big|\; : x_k^2$$

$$\eta^2 - \eta < 0$$

$$\eta(\eta - 1) < 0$$

Minimizer of Lipschitz parabola

Some general highlights about Lipschitz properties are needed for explanation. If a function $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and its gradient satisfies Lipschitz conditions with constant L, then $\forall x, y \in \mathbb{R}^n$:

$$|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leq \frac{L}{2} \|y - x\|^2,$$

which geometrically means, that if we'll fix some point $x_0 \in \mathbb{R}^n$ and define two parabolas:

$$\phi_1(x) = f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle - \frac{L}{2} \|x - x_0\|^2,$$

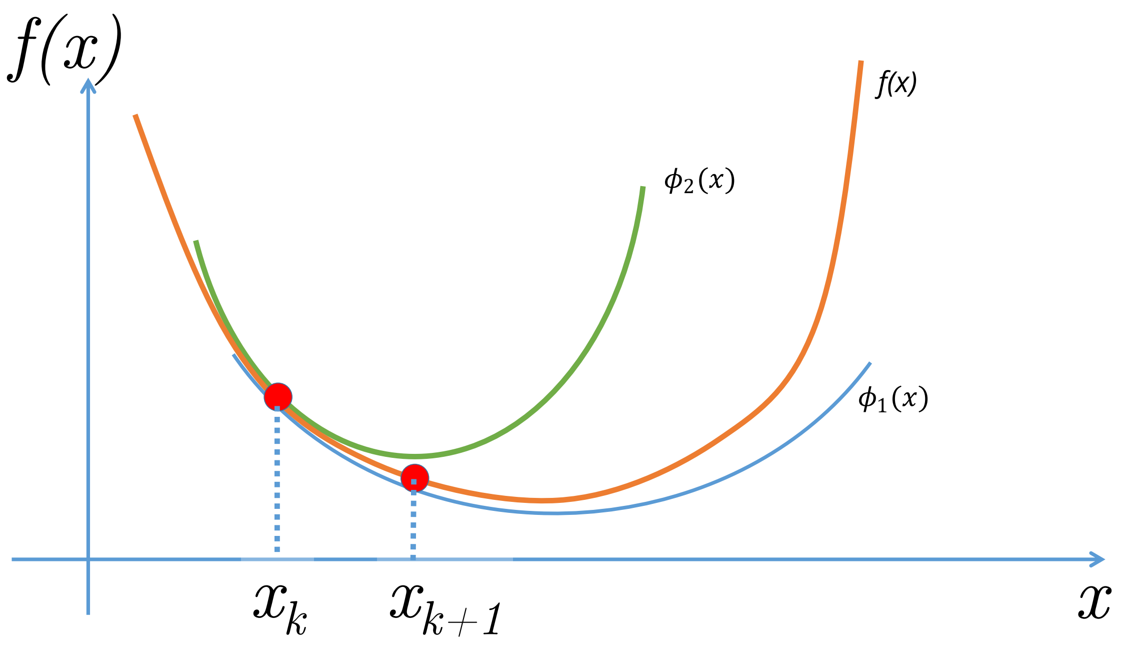
$$\phi_2(x) = f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle + \frac{L}{2} \|x - x_0\|^2.$$

Then

$$\phi_1(x) \leq f(x) \leq \phi_2(x) \quad \forall x \in \mathbb{R}^n.$$

Now, if we have global upper bound on the function, in a form of parabola, we can try to go directly to its minimum.

$$\begin{aligned}
& \phi_2(x) = 0 \\
& \nabla f(x_0) + L(x^* - x_0) = 0 \\
& x^* = x_0 - \frac{1}{L} \nabla f(x_0) \\
& x_{k+1} = x_k - \frac{1}{L} \nabla f(x_k)
\end{aligned}$$



This way leads to the $\frac{1}{L}$ stepsize choosing. However, often the L constant is not known.

But if the function is twice continuously differentiable and its gradient has Lipschitz constant L, we can derive a way to estimate this constant $\forall x \in \mathbb{R}^n$:

$$\|\nabla^2 f(x)\| \leq L$$

or

$$-LI_n \preceq \nabla^2 f(x) \preceq LI_n$$

Stepsize choosing strategies

Stepsize choosing strategy η_k significantly affects convergence. General {%include link.html title='Line search algorithms might help in choosing scalar parameter.

Constant stepsize

For $f \in C_L^{1,1}$:

$$\boxed{\eta_k = \eta} = 0.1 \;/ 0.01 \;/ 0.001$$

$$f(x_k) - f(x_{k+1}) \geq \eta \left(1 - \frac{1}{2}L\eta\right) \|\nabla f(x_k)\|^2$$

With choosing $\eta = \frac{1}{L}$, we have:

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2L}\|\nabla f(x_k)\|^2$$

Fixed sequence

$$\boxed{\eta_k = \frac{1}{\sqrt{k+1}}}$$

The latter 2 strategies are the simplest in terms of implementation and analytical analysis. It is clear that this approach does not often work very well in practice (the function geometry is not known in advance).

Метод наискорейшего спуска

Exact line search aka steepest descent

$$\eta_k = \arg\min_{\eta \in \mathbb{R}^+} f(x_{k+1}) = \arg\min_{\eta \in \mathbb{R}^+} f(x_k - \eta \nabla f(x_k)) = \varphi(\eta) = \arg\min_{\eta \in \mathbb{R}^+} \varphi(\eta)$$

More theoretical than practical approach. It also allows you to analyze the convergence, but often exact line search can be difficult if the function calculation takes too long or costs a lot.

Interesting theoretical property of this method is that each following iteration is orthogonal to the previous one:

есть x_k $f(x)$, $\nabla f(x)$
хочу найти шаг? η

$$\eta_k = \arg\min_{\eta \in \mathbb{R}^+} f(\underbrace{x_k - \eta \nabla f(x_k)}_{x_{k+1}})$$

$x_{k+1} = x_k - \eta \nabla f(x_k)$

$f(x_{k+1}) \to \min_\eta$

Optimality conditions:

$$\nabla f(x_{k+1})^\top \nabla f(x_k) = 0$$

Goldstein-Armijo

This strategy of inexact line search works well in practice, as well as it has the following geometric interpretation:

Let's consider the following scalar function while being at a specific point of x_k:

$$\phi(\eta) = f(x_k - \eta \nabla f(x_k)), \eta \geq 0$$

consider first order approximation of $\phi(\eta)$:

$$\phi(\eta) \approx f(x_k) - \eta \nabla f(x_k)^\top \nabla f(x_k)$$

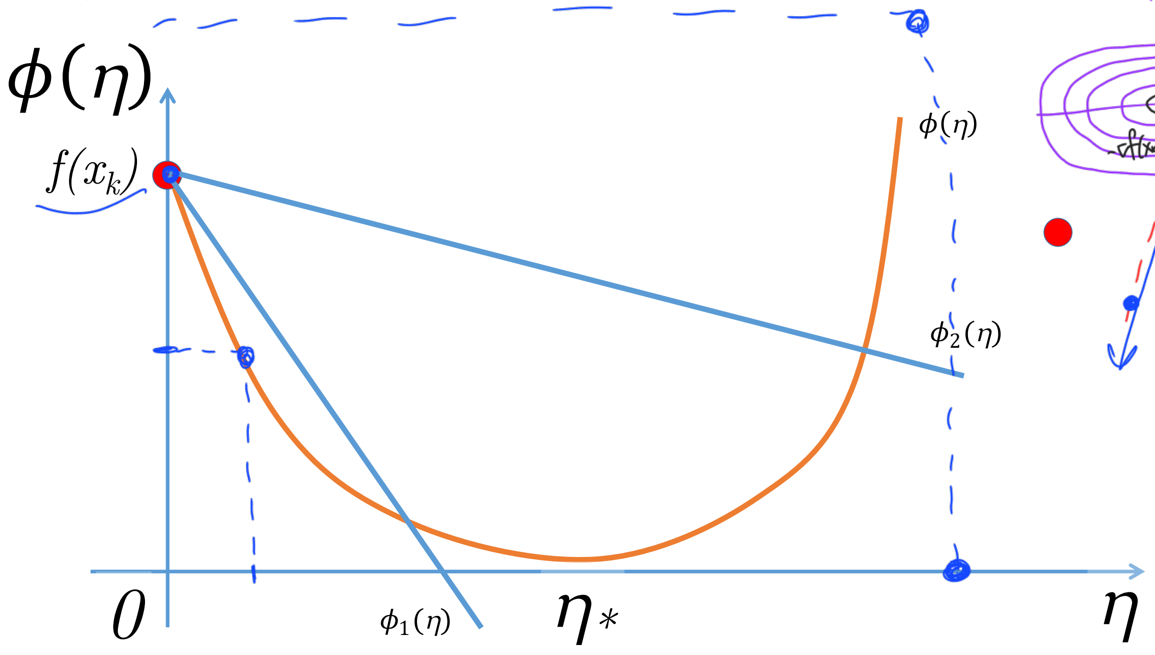
Let's consider also 2 linear scalar functions $\phi_1(\eta), \phi_2(\eta)$:

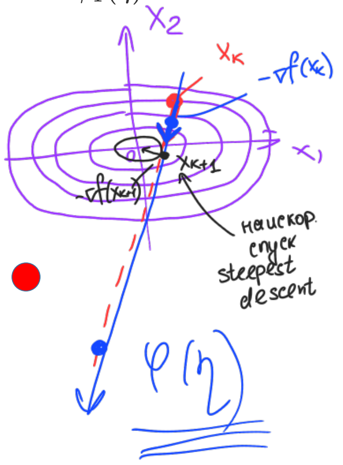
$$\phi_1(\eta) = f(x_k) - \alpha \eta \|\nabla f(x_k)\|^2$$

and

$$\phi_2(\eta) = f(x_k) - \beta \eta \|\nabla f(x_k)\|^2$$

Note, that Goldstein-Armijo conditions determine the location of the function $\phi(\eta)$ between $\phi_1(\eta)$ and $\phi_2(\eta)$. Typically, we choose $\alpha = \rho$ and $\beta = 1 - \rho$,while $\rho \in (0.5, 1)$





Convergence analysis

Quadratic case

Bounds

Conditions	$\|f(x_k) - f(x^*)\| \leq$	**Type of convergence**	$\|x_k - x^*\| \leq$
Convex Lipschitz-continuous function(G)	$\mathcal{O}\left(\dfrac{1}{k}\right) \; \dfrac{GR}{k}$	Sublinear	
Convex Lipschitz-continuous gradient (L)	$\mathcal{O}\left(\dfrac{1}{k}\right) \; \dfrac{LR^2}{k}$	Sublinear	
μ-Strongly convex Lipschitz-continuous gradient(L)		Linear	$(1 - \eta\mu)^k R^2$
μ-Strongly convex Lipschitz-continuous hessian(M)		Locally linear $R < \overline{R}$	$\dfrac{\overline{R}R}{\overline{R} - R}\left(1 - \dfrac{2\mu}{L + 3\mu}\right)$

- $R = \|x_0 - x^*\|$ - initial distance
- $\overline{R} = \dfrac{2\mu}{M}$

Materials

- The zen of gradient descent. Moritz Hardt
- Great visualization

Line search

Problem

Suppose, we have a problem of minimization of a function $f(x) : \mathbb{R} \to \mathbb{R}$ of scalar variable:

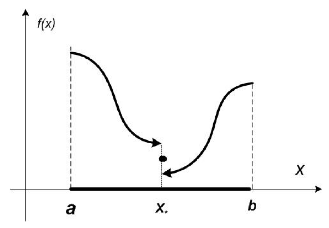
$$\boxed{f(x) \to \min_{x \in \mathbb{R}}}$$

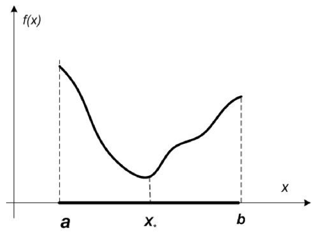
Sometimes, we refer to the similar problem of finding minimum on the line segment $[a, b]$:

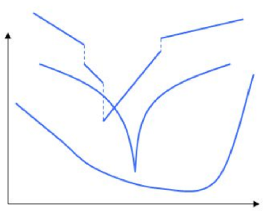
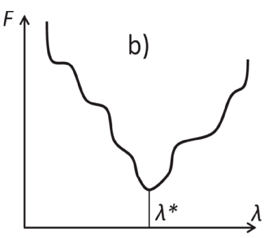
$$f(x) \to \min_{x \in [a,b]}$$

Line search is one of the simplest formal optimization problems, however, it is an important link in solving more complex tasks, so it is very important to solve it effectively. Let's restrict the class of problems under consideration where $f(x)$ is a *unimodal function*.

Function $f(x)$ is called **unimodal** on $[a, b]$, if there is $x_* \in [a, b]$, that $f(x_1) > f(x_2) \quad \forall a \leq x_1 < x_2 < x_*$ and $f(x_1) < f(x_2) \quad \forall x_* < x_1 < x_2 \leq b$



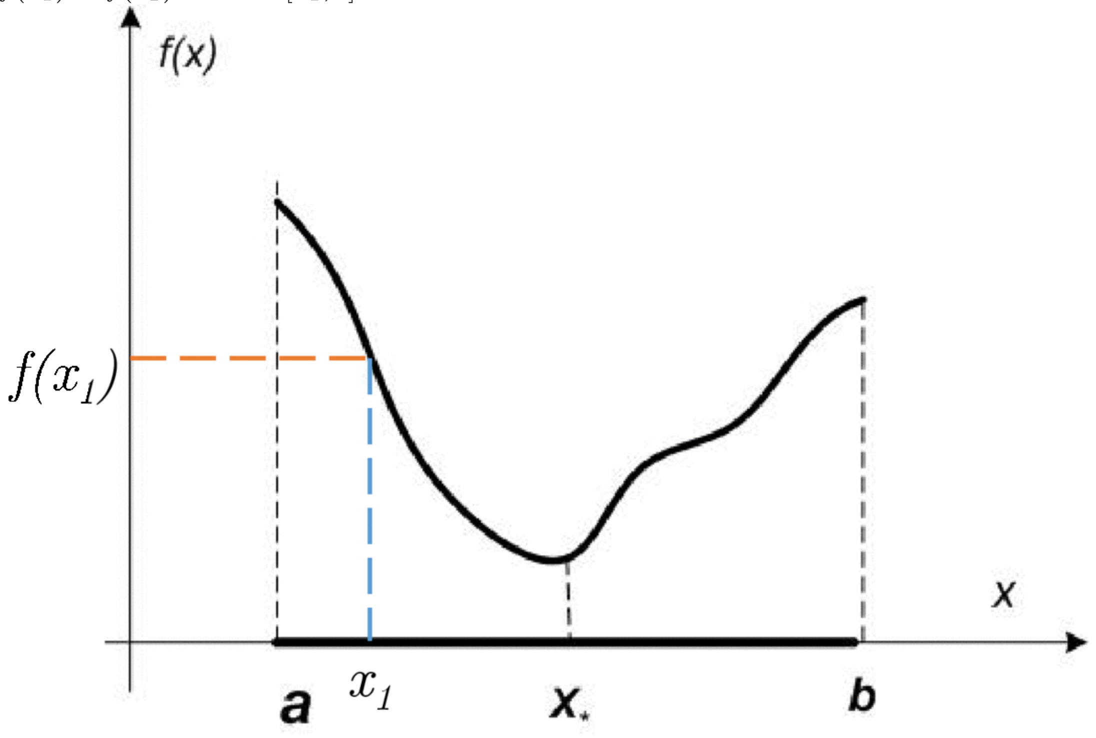


Key property of unimodal functions

Let $f(x)$ be unimodal function on $[a, b]$. Than if $x_1 < x_2 \in [a, b]$, then:

- if $f(x_1) \leq f(x_2) \to x_* \in [a, x_2]$
- if $f(x_1) \geq f(x_2) \to x_* \in [x_1, b]$



Code

[Open in Colab]

References

- [CMC seminars (ru)]

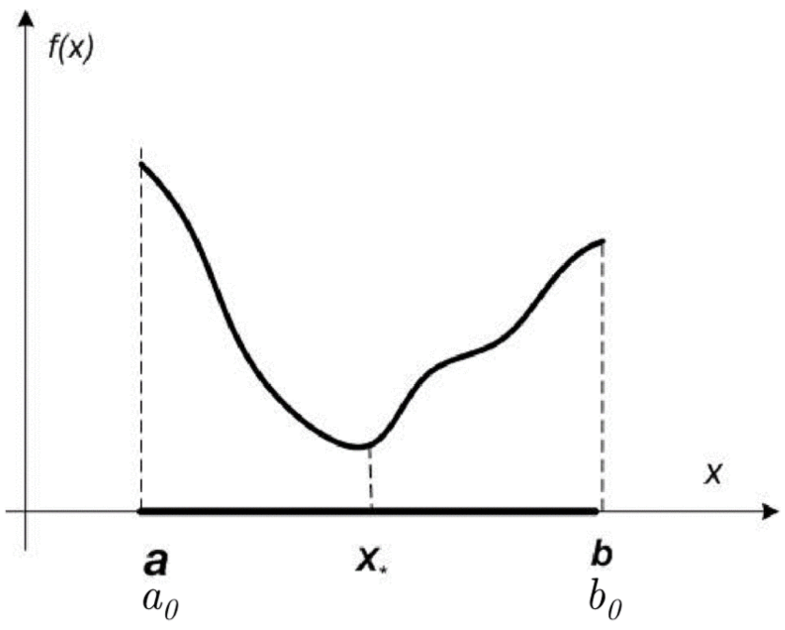
Binary search

Idea

We divide a segment into two equal parts and choose the one that contains the solution of the problem using the values of functions.

Algorithm

```
def binary_search(f, a, b, epsilon):
    c = (a + b) / 2
    while abs(b - a) > epsilon:
        y = (a + c) / 2.0
        if f(y) <= f(c):
            b = c
            c = y
        else:
            z = (b + c) / 2.0
            if f(c) <= f(z):
                a = y
                b = z
            else:
                a = c
                c = z
    return c
```



Bounds

The length of the line segment on $k+1$-th iteration:

$$\Delta_{k+1} = b_{k+1} - a_{k+1} = \frac{1}{2^k}(b-a)$$

For unimodal functions, this holds if we select the middle of a segment as an output of the iteration x_{k+1}:

$$|x_{k+1} - x_*| \leq \frac{\Delta_{k+1}}{2} \leq \frac{1}{2^{k+1}}(b-a) \leq (0.5)^{k+1} \cdot (b-a)$$

Note, that at each iteration we ask oracle no more, than 2 times, so the number of function evaluations is $N = 2 \cdot k$, which implies:

$$|x_{k+1} - x_*| \leq (0.5)^{\frac{N}{2}+1} \cdot (b-a) \leq (0.707)^N \frac{b-a}{2}$$

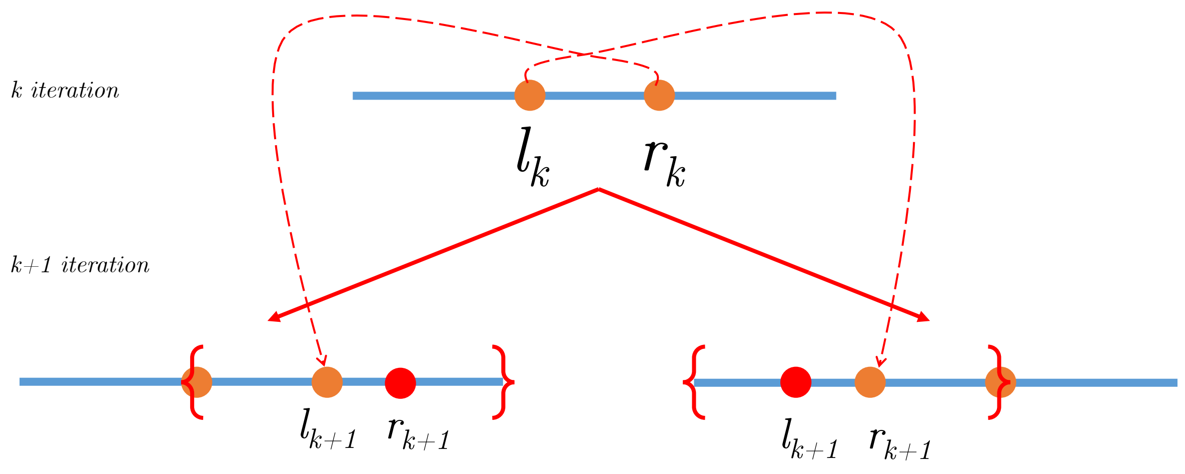
By marking the right side of the last inequality for ε, we get the number of method iterations needed to achieve ε accuracy:

$$K = \left\lceil \log_2 \frac{b-a}{\varepsilon} - 1 \right\rceil$$

Golden search

Idea

The idea is quite similar to the dichotomy method. There are two golden points on the line segment (left and right) and the insightful idea is, that on the next iteration one of the points will remains the golden point.



Algorithm

```
def golden_search(f, a, b, epsilon):
    tau = (sqrt(5) + 1) / 2
    y = a + (b - a) / tau**2
    z = a + (b - a) / tau
    while b - a > epsilon:
        if f(y) <= f(z):
            b = z
            z = y
            y = a + (b - a) / tau**2
        else:
            a = y
            y = z
            z = a + (b - a) / tau
    return (a + b) / 2
```

Bounds

$$|x_{k+1} - x_*| \leq b_{k+1} - a_{k+1} = \left(\frac{1}{\tau} \right)^{N-1} (b-a) \approx 0.618^k (b-a),$$

where $\tau = \frac{\sqrt{5}+1}{2}$.

- The geometric progression constant **more** than the dichotomy method - 0.618 worse than 0.5
- The number of function calls **is less** than for the dichotomy method - 0.707 worse than 0.618 - (for each iteration of the dichotomy method, except for the first one, the function is calculated no more than 2 times, and for the gold method - no more than one)

Successive parabolic interpolation

Idea

$$f(x_1, x_2) = x_1^2 + K x_2^2$$

$$x^T A x = (x_1 \; x_2) \begin{pmatrix} 1 & 0 \\ 0 & K \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} =$$

$$= (x_1 \; x_2) \begin{pmatrix} x_1 \cdot 1 + 0 \cdot x_2 \\ 0 \cdot x_1 + K x_2 \end{pmatrix} = x_1^2 + K x_2^2$$

Sampling 3 points of a function determines unique parabola. Using this information we will go directly to its minimum. Suppose, we have 3 points $x_1 < x_2 < x_3$ such that line segment $[x_1, x_3]$ contains minimum of a function $f(x)$. Than, we need to solve following system of equations:

$$ax_i^2 + bx_i + c = f_i = f(x_i), i = 1, 2, 3$$

Note, that this system is linear, since we need to solve it on a, b, c. Minimum of this parabola will be calculated as:

$$u = -\frac{b}{2a} = x_2 - \frac{(x_2 - x_1)^2(f_2 - f_3) - (x_2 - x_3)^2(f_2 - f_1)}{2\left[(x_2 - x_1)(f_2 - f_3) - (x_2 - x_3)(f_2 - f_1)\right]}$$

Note, that if $f_2 < f_1, f_2 < f_3$, than u will lie in $[x_1, x_3]$

Algorithm

```python
def parabola_search(f, x1, x2, x3, epsilon):
    f1, f2, f3 = f(x1), f(x2), f(x3)
    while x3 - x1 > epsilon:
        u = x2 - ((x2 - x1)**2*(f2 - f3) - (x2 - x3)**2*(f2 - f1))/(2*((x2 - 
x1)*(f2 - f3) - (x2 - x3)*(f2 - f1)))
        fu = f(u)

        if x2 <= u:
            if f2 <= fu:
                x1, x2, x3 = x1, x2, u
                f1, f2, f3 = f1, f2, fu
            else:
                x1, x2, x3 = x2, u, x3
                f1, f2, f3 = f2, fu, f3
        else:
            if fu <= f2:
                x1, x2, x3 = x1, u, x2
                f1, f2, f3 = f1, fu, f2
            else:
                x1, x2, x3 = u, x2, x3
                f1, f2, f3 = fu, f2, f3
    return (x1 + x3) / 2
```

Bounds

The convergence of this method is superlinear, but local, which means, that you can take profit from using this method only near some neighbour of optimum.

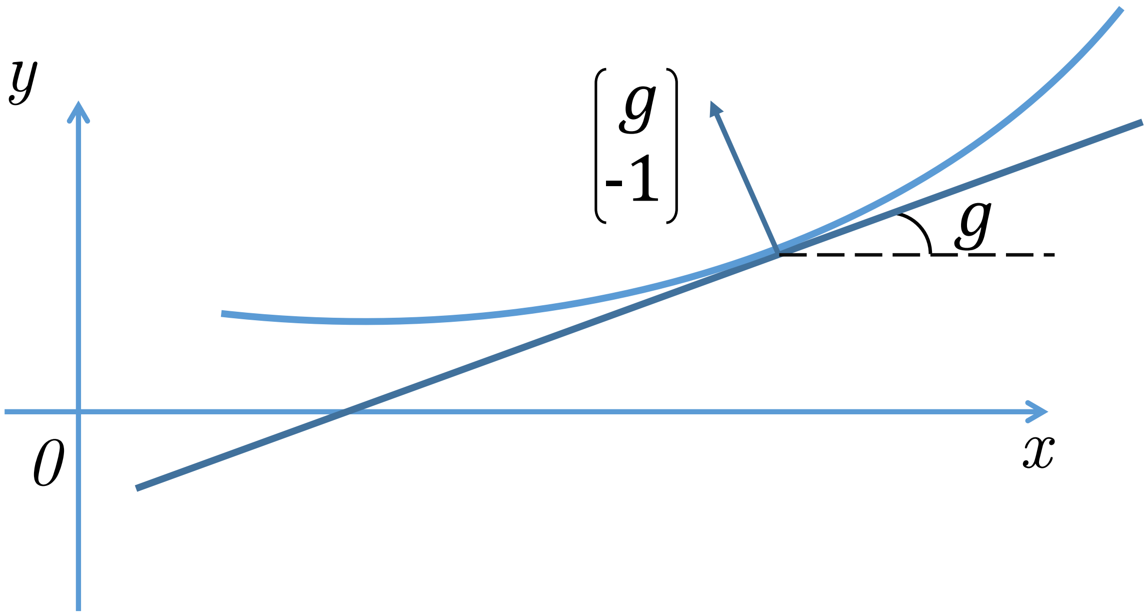
Subgradient and subdifferential

Motivation

Важным свойством непрерывной выпуклой функции $f(x)$ является то, что в выбранной точке x_0 для всех $x \in \text{dom } f$ выполнено неравенство:

$$f(x) \geq f(x_0) + \langle g, x - x_0 \rangle$$

для некоторого вектора g, то есть касательная к графику функции является *глобальной* оценкой снизу для функции.



- Если $f(x)$ - дифференцируема, то $g = \nabla f(x_0)$
- Не все непрерывные выпуклые функции дифференцируемы :)

Не хочется лишаться такого вкусного свойства.

Subgradient

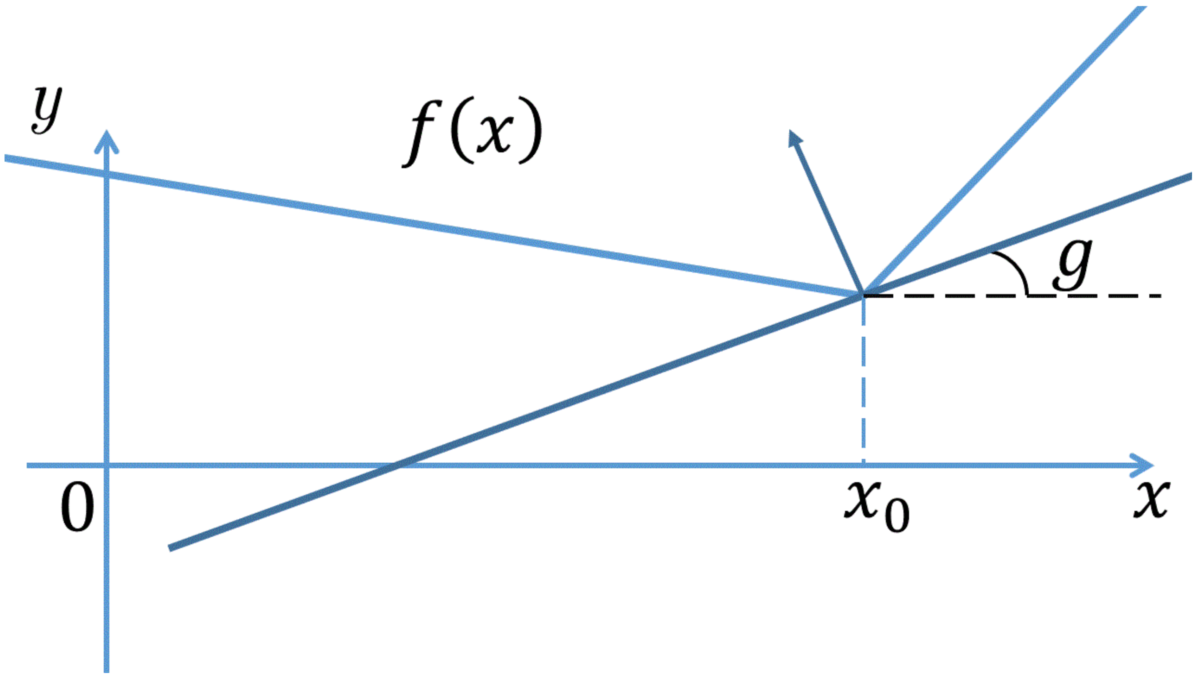
Вектор g называется **субградиентом** функции $f(x) : S \to \mathbb{R}$ в точке x_0, если $\forall x \in S$:

$$f(x) \geq f(x_0) + \langle g, x - x_0 \rangle$$

Subdifferential

Множество всех субградиентов функции $f(x)$ в точке x_0 называется **субдифференциалом** f в x_0 и обозначается $\partial f(x_0)$.

- Если $x_0 \in \mathbf{ri} S$, то $\partial f(x_0)$ выпуклое компактное множество.
- Выпуклая функция $f(x)$ дифференцируема в точке x_0 $\iff$ $\partial f(x_0) = \nabla f(x_0)$
- Если $\partial f(x_0) \neq \emptyset \quad \forall x_0 \in S$, то $f(x)$ - выпукла на S.



Moreau - Rockafellar theorem (subdifferential of a linear combination)

Пусть $f_i(x)$ - выпуклые функции на выпуклых множествах $S_i,\ i = \overline{1,n}$.
Тогда, если $\bigcap\limits_{i=1}^n \mathbf{ri} S_i \neq \emptyset$ то функция $f(x) = \sum\limits_{i=1}^n a_i f_i(x),\ a_i > 0$ имеет субдифференциал $\partial_S f(x)$ на множестве $S = \bigcap\limits_{i=1}^n S_i$ и

$$\partial_S f(x) = \sum_{i=1}^n a_i \partial_{S_i} f_i(x)$$

Dubovitsky - Milutin theorem (subdifferential of a point-wise maximum)

Пусть $f_i(x)$ - выпуклые функции на открытом выпуклом множестве $S \subseteq \mathbb{R}^n,\ x_0 \in S$, а поточечный максимум определяется как $f(x) = \max\limits_i f_i(x)$. Тогда:

$$\partial_S f(x_0) = \mathbf{conv} \left\{ \bigcup_{i \in I(x_0)} \partial_S f_i(x_0) \right\},$$

где $I(x) = \{ i \in [1:m] : f_i(x) = f(x) \}$

Chain rule for subdifferentials

Пусть $g_1, \ldots, g_m$ - выпуклые функции на открытом выпуклом множестве $S \subseteq \mathbb{R}^n$, $g = (g_1, \ldots, g_m)$ - образованная из них вектор - функция, φ - монотонно неубывающая выпуклая функция на открытом выпуклом множестве $U \subseteq \mathbb{R}^m$, причем $g(S) \subseteq U$. Тогда субдифференциал функции $f(x) = \varphi\left(g(x)\right)$ имеет вид:

$$\partial f(x) = \bigcup_{p \in \partial \varphi(u)} \left(\sum_{i=1}^m p_i \partial g_i(x) \right),$$

где $u = g(x)$

В частности, если функция φ дифференцируема в точке $u = g(x)$, то формула запишется так:

$$\partial f(x) = \sum_{i=1}^m \frac{\partial \varphi}{\partial u_i}(u) \partial g_i(x)$$

Subdifferential calculus

- $\partial(\alpha f)(x) = \alpha \partial f(x)$, for $\alpha \geq 0$
- $\partial(\sum f_i)(x) = \sum \partial f_i(x)$, f_i - выпуклые функции
- $\partial(f(Ax + b))(x) = A^T \partial f(Ax + b)$, f - выпуклая функция

Examples

Концептуально, различают три способа решения задач на поиск субградиента:

- Теоремы Моро - Рокафеллара, композиции, максимума
- Геометрически
- По определению

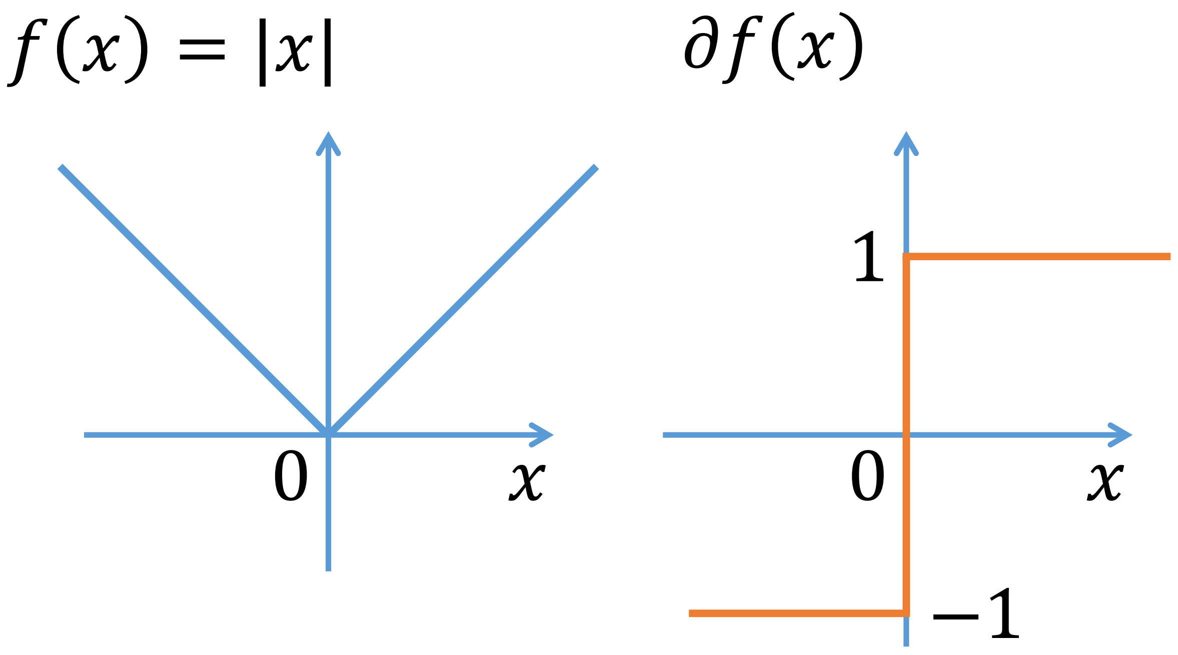
1

Найти $\partial f(x)$, если $f(x) = |x|$

Решение:

Решить задачу можно либо геометрически (в каждой точке числовой прямой указать угловые коэффициенты прямых, глобально подпирающих функцию снизу), либо по теореме Моро - Рокафеллара, рассмотрев $f(x)$ как композицию выпуклых функций:

$$f(x) = \max\{-x, x\}$$



2

Найти $\partial f(x)$, если $f(x) = |x-1| + |x+1|$

Решение:

Совершенно аналогично применяем теорему Моро - Рокафеллара, учитывая следующее:

$$\partial f_1(x) = \begin{cases} -1, & x < 1 \\ [-1;1], & x = 1 \\ 1, & x > 1 \end{cases} \qquad \partial f_2(x) = \begin{cases} -1, & x < -1 \\ [-1;1], & x = -1 \\ 1, & x > -1 \end{cases}$$

Таким образом:

$$\partial f(x) = \begin{cases} -2, & x < -1 \\ [-2;0], & x = -1 \\ 0, & -1 < x < 1 \\ [0;2], & x = 1 \\ 2, & x > 1 \end{cases}$$

3

Найти $\partial f(x)$, если $f(x) = [\max(0, f_0(x))]^q$. Здесь $f_0(x)$ - выпуклая функция на открытом выпуклом множестве S, $q \geq 1$.

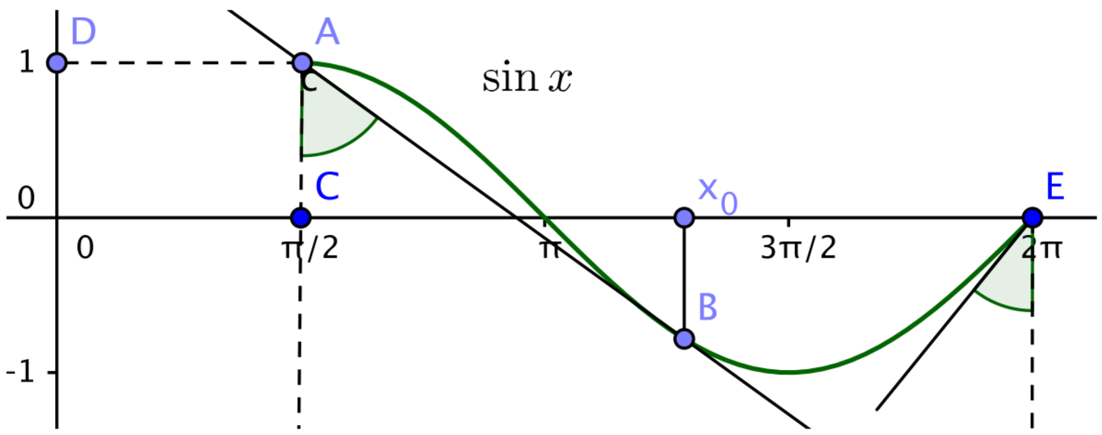
Решение:
Согласно теореме о композиции (функция $\varphi(x) = x^q$ - дифференцируема), а $g(x) = \max(0, f_0(x))$ имеем: $\partial f(x) = q(g(x))^{q-1}\partial g(x)$

По теореме о поточечном максимуме:

$$\partial g(x) = \begin{cases} \partial f_0(x), \quad f_0(x) > 0, \\ \{0\}, \quad f_0(x) < 0 \\ \{a \mid a = \lambda a', \ 0 \leq \lambda \leq 1, \ a' \in \partial f_0(x)\}, \quad f_0(x) = 0 \end{cases}$$

4

Найти $\partial f(x)$, если $f(x) = \sin x, x \in [\pi/2; 2\pi]$



$$\partial f_G(x) = \begin{cases} (-\infty, \cos x_0], & x = \pi/2; \\ \varnothing, & x \in (\pi/2, x_0); \\ \cos x, & x \in [x_0, 2\pi); \\ [1, +\infty), & x = 2\pi. \end{cases}$$

5

Найти $\partial f(x)$, если $f(x) = |c_1^\top x| + |c_2^\top x|$

Решение: Пусть $f_1(x) = |c_1^\top x|$, а $f_2(x) = |c_2^\top x|$. Так как эти функции выпуклы, субдифференциал их суммы равен сумме субдифференциалов. Найдем каждый из них:

$$\partial f_1(x) = \partial \left(\max\{c_1^\top x, -c_1^\top x\}\right) = \begin{cases} -c_1, & c_1^\top x < 0 \\ \mathbf{conv}(-c_1; c_1), & c_1^\top x = 0 \\ c_1, & c_1^\top x > 0 \end{cases}$$

$$\partial f_2(x) = \partial \left(\max\{c_2^\top x, -c_2^\top x\}\right) = \begin{cases} -c_2, & c_2^\top x < 0 \\ \mathbf{conv}(-c_2; c_2), & c_2^\top x = 0 \\ c_2, & c_2^\top x > 0 \end{cases}$$

Далее интересными представляются лишь различные взаимные расположения векторов c_1 и c_2, рассмотрение которых предлагается читателю.

6

Найти $\partial f(x)$, если $f(x) = \|x\|_1$

Решение: По определению

$$\|x\|_1 = |x_1| + |x_2| + \ldots + |x_n| = s_1 x_1 + s_2 x_2 + \ldots + s_n x_n$$

Рассмотрим эту сумму как поточечный максимум линейных функций по x: $g(x) = s^\top x$, где $s_i = \{-1, 1\}$. Каждая такая функция однозначно определяется набором коэффициентов $\{s_i\}_{i=1}^n$.

Тогда по теореме Дубовицкого - Милютина, в каждой точке $\partial f = \mathbf{conv}\left(\bigcup\limits_{i \in I(x)} \partial g_i(x)\right)$

Заметим, что $\partial g(x) = \partial\left(\max\{s^\top x, -s^\top x\}\right) = \begin{cases} -s, & s^\top x < 0 \\ \mathbf{conv}(-s; s), & s^\top x = 0 \\ s, & s^\top x > 0 \end{cases}$.

Причем, правило выбора "активной" функции поточечного максимума в каждой точке следующее:

- Если j-ая координата точки отрицательна, $s_i^j = -1$
- Если j-ая координата точки положительна, $s_i^j = 1$
- Если j-ая координата точки равна нулю, то подходят оба варианта коэффициентов и соответствующих им функций, а значит, необходимо включать субградиенты этих функций в объединение в теореме Дубовицкого - Милютина.

В итоге получаем ответ:

$$\partial f(x) = \left\{g \ : \ \|g\|_\infty \leq 1, \quad g^\top x = \|x\|_1\right\}$$

References

- Lecture Notes for ORIE 6300: Mathematical Programming I by Damek Davis