@bibtex Files

Summary

A classical problem of function minimization is considered.

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

- 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 $\eta_k = \frac{1}{L}$.
- 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:

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

Направление антиградиента : направление наискорейшего локального убывания

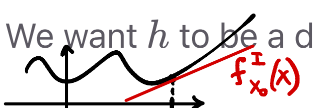
Intuition

Direction of local steepest descent

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

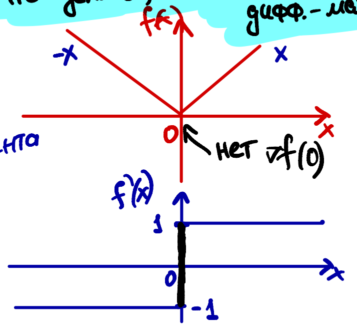
$$f(x + \eta h) = f(x) + \eta \langle f'(x), h \rangle + o(\eta)$$

We want h to be a decreasing direction:



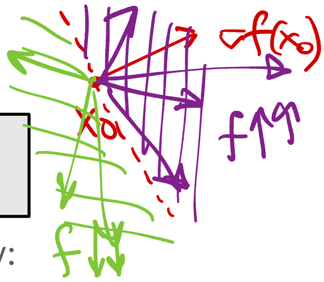
$$f(x + \eta h) < f(x)$$

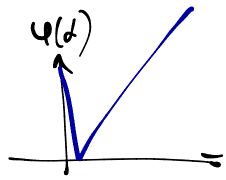
$$f(x) + \eta \langle f'(x), h \rangle + o(\eta) < f(x)$$



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

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





Also from Cauchy–Bunyakovsky–Schwarz inequality:

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

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.

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

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.

Necessary local minimum condition

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

необх. усл. лок. экстр

градиент не может меняться скачком

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

Липшицевость градиента

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\|$$

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 $\forall x, y \in \mathbb{R}^n$:

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

$L_1 = 1$

$L_2 = 1000$

с какой функцией удобнее работать

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

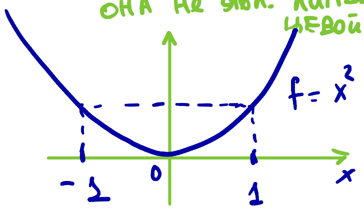
макс. шаг град. спуска, при котором метод сходится → $\alpha_{MAX} \approx \frac{1}{L}$

$\alpha \leq \alpha_{MAX}$

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

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

Функция, у которой везде есть ∇f, но она не явл. липшицевой



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.

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

Пусть $f(x) = \frac{1}{2}\|Ax - b\|_2^2$, $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$

$L = ?$

$\nabla f(x) = A^T(Ax - b)$

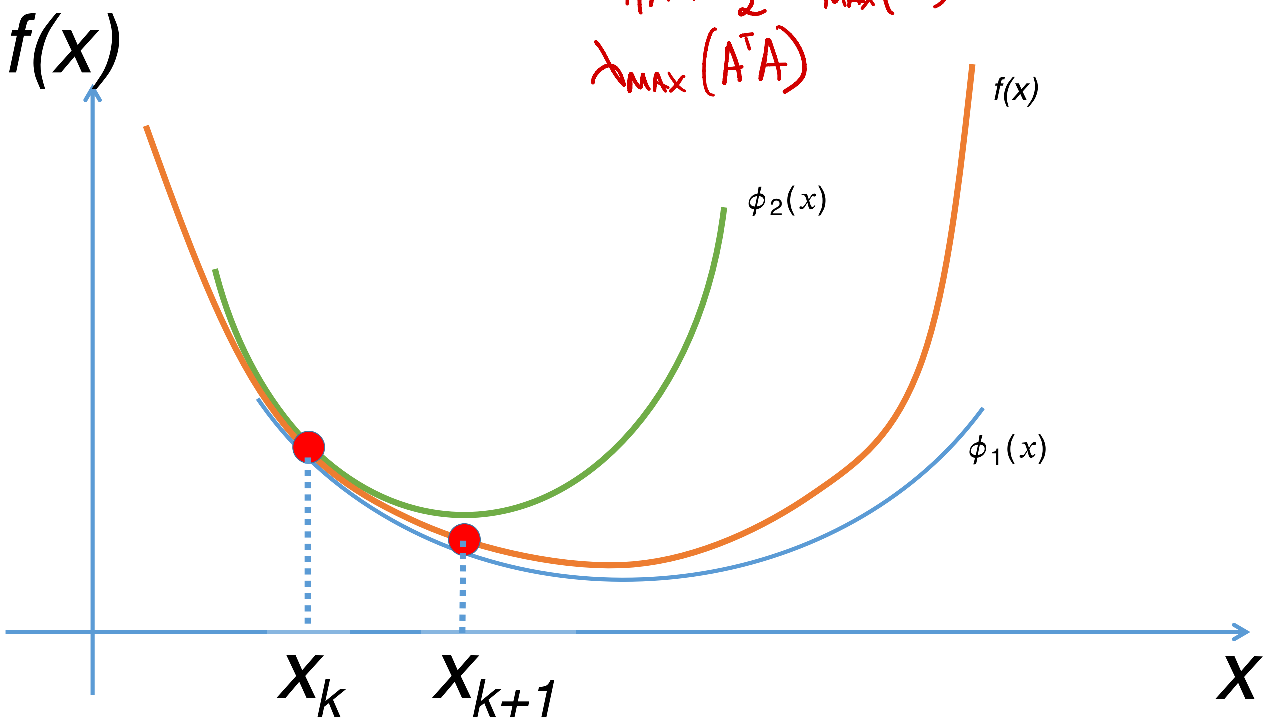
$\|\nabla f(x) - \nabla f(y)\| =$

$= \|A^TAx - A^Tb - A^TAy + A^Tb\| =$

$= \|A^TA(x-y)\| \leq \|A^TA\| \|x-y\|$

$$\alpha = \frac{1}{\lambda_{MAX}(A^TA)}$$

$L = \|A^T A\|_2 = \sigma_{MAX}(A^T A) = \lambda_{MAX}(A^T A)$



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 $\forall x \in \mathbb{R}^n$:

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

Если речь идёт про произвольную функцию $f(x)$, то

$$L = \lambda_{MAX}(\nabla^2 f(x))$$

or

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

$$\alpha \approx \frac{1}{\lambda_{MAX}(\nabla^2 f(x))}$$

Макс. λ матрицы A можно оценить быстро с помощью POWER METHOD $A \cdot A \cdot A \cdot A \cdot A \cdot x_0 \to e$

Stepsize choosing strategies

Stepsize choosing strategy η_k significantly affects convergence. General Line search algorithms might help in choosing scalar parameter.

$$\lambda_k = \frac{x_k^T A x_k}{x_k^T x_k} \to \lambda_{MAX}$$

$$x_k = \frac{A \cdot x_{k-1}}{\|A x_{k-1}\|}$$

Constant stepsize

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

$$\eta_k = \eta$$

\+ EASY

\- можно не угадать

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

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

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

learning rate scheduler

Fixed sequence

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

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).

Exact line search aka steepest descent

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

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:

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

Optimality conditions:

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

Дано x_k, $-\nabla f(x_k)$

старт, направление

как далеко

$$f(x_k - \alpha \cdot \nabla f(x_k)) \to \min_{\alpha \in \mathbb{R}^{++}}$$

одномерная оптимизация

Goldstein-Armijo

Convergence analysis

Convex case

Lipischitz continuity of the gradient

Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable, and additionally $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \; \forall x, y \in \mathbb{R}^n$

i.e. , ∇f is Lipschitz continuous with constant $L > 0$.

Since ∇f Lipschitz with constant L, which means $\nabla^2 f \preceq LI$, we have $\forall x, y, z$:

$$(x-y)^\top(\nabla^2 f(z) - LI)(x-y) \leq 0$$

$$(x-y)^\top \nabla^2 f(z)(x-y) \leq L\|x-y\|^2$$

Now we'll consider second order Taylor approximation of $f(y)$ and Taylor's Remainder Theorem (we assum, that the function f is continuously differentiable), we have $\forall x, y, \exists z \in [x, y]:$

$$\begin{aligned} f(y) &= f(x) + \nabla f(x)^\top(y-x) + \frac{1}{2}(x-y)^\top \nabla^2 f(z)(x-y) \\ &\leq f(x) + \nabla f(x)^\top(y-x) + \frac{L}{2}\|x-y\|^2 \end{aligned}$$

For the gradient descent we have $x = x_k, y = x_{k+1}, x_{k+1} = x_k - \eta_k \nabla f(x_k)$:

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \nabla f(x_k)^\top(-\eta_k \nabla f(x_k)) + \frac{L}{2}(\eta_k \nabla f(x_k))^2 \\ &\leq f(x_k) - \left(1 - \frac{L\eta}{2}\right)\eta\|\nabla f(x_k)\|^2 \end{aligned}$$

Optimal constant stepsize

Now, if we'll consider constant stepsize strategy and will maximize $\left(1 - \frac{L\eta}{2}\right)\eta \to \max\limits_{\eta}$, we'll get $\eta = \frac{1}{L}$.

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

Convexity

$f(x_k) \leq f(x^*) + \nabla f(x_k)^\top(x_k - x^*)$

That's why we have:

$$
\begin{aligned}
f(x_{k+1}) &\leq f(x^*) + \nabla f(x_k)^\top (x_k - x^*) - \frac{1}{2L}\|\nabla f(x_k)\|^2 \\
&= f(x^*) + \frac{L}{2}\left(\|x_k - x^*\|^2 - \|x_k - x^* - \frac{1}{L}\nabla f(x_k)\|^2\right) \\
&= f(x^*) + \frac{L}{2}\left(\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2\right)
\end{aligned}
$$

Thus, summing over all iterations, we have:

$$
\begin{aligned}
\sum_{i=1}^{k}(f(x_i) - f(x^*)) &\leq \frac{L}{2}\left(\|x_0 - x^*\|^2 - \|x_k - x^*\|^2\right) \\
&\leq \frac{L}{2}\|x_0 - x^*\|^2 = \frac{LR^2}{2},
\end{aligned}
$$

where $R = \|x_0 - x^*\|$. And due to convexity:

$$
f(x_k) - f(x^*) \leq \frac{1}{k}\sum_{i=1}^{k}(f(x_i) - f(x^*)) \leq \frac{LR^2}{2k} = \frac{R^2}{2\eta k}
$$

Strongly convex case

If the function is strongly convex:

$$
f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|y - x\|^2 \; \forall x, y \in \mathbb{R}^n
$$

...

$\|x_{k+1} - x^*\|^2 \leq (1 - \eta\mu)\|x_k - x^*\|^2$

Bounds

Conditions	$\|f(x_k) - f(x^*)\| \leq$	Type of convergence	$\|x_k - x^*\| \leq$
Convex Lipschitz-continuous function(G)	$\mathcal{O}\left(\frac{1}{k}\right) \; \frac{GR}{k}$	Sublinear	
Convex			

Lipschitz-continuous gradient (L)	$\mathcal{O}\left(\frac{1}{k}\right)$ $\frac{LR^2}{k}$	Sublinear	
μ-Strongly convex Lipschitz-continuous gradient(L)		Linear	$(1-\eta\mu)^k R^2$
μ-Strongly convex Lipschitz-continuous hessian(M)		Locally linear $R < \overline{R}$	$\frac{\overline{R}R}{\overline{R}-R}\left(1-\frac{2\mu}{L+3\mu}\right)$

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

Materials

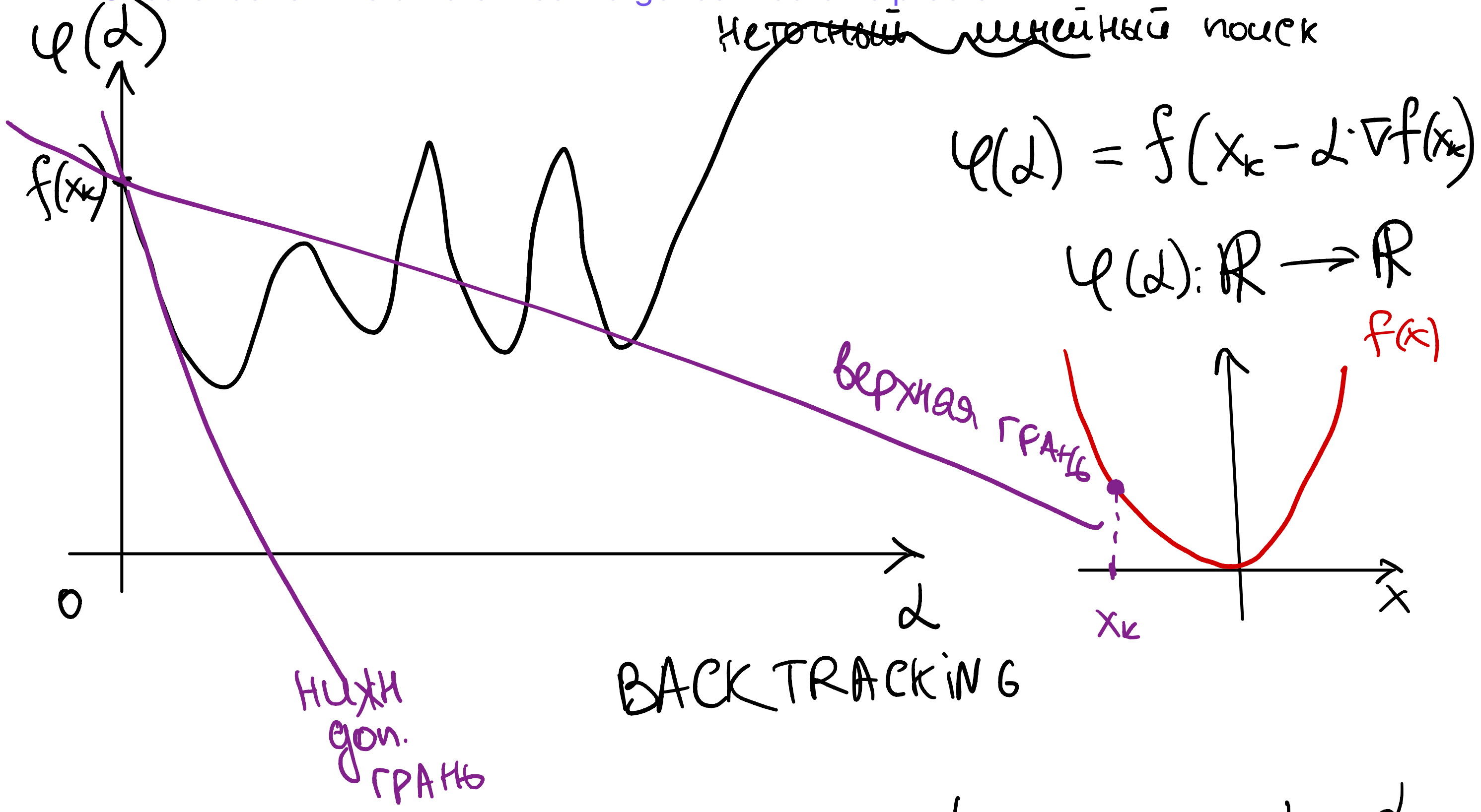
Armijo, Goldstein, Wolfe

- The zen of gradient descent. Moritz Hardt
- Great visualization
- Cheatsheet on the different convergence theorems proofs

Неточный линейный поиск

$\varphi(\alpha) = f(x_k - \alpha \cdot \nabla f(x_k))$

$\varphi(\alpha): \mathbb{R} \to \mathbb{R}$



BACKTRACKING

$\alpha_0 = 100$ не подходит; $\alpha_1 = \frac{\alpha_0}{2}$ не подх; $\alpha_2 = \frac{\alpha_1}{2}$

Можно ли используя лишь информацию 1-го порядка получить метод быстрее градиентного спуска.

ДА!

Б.Т. Поляк

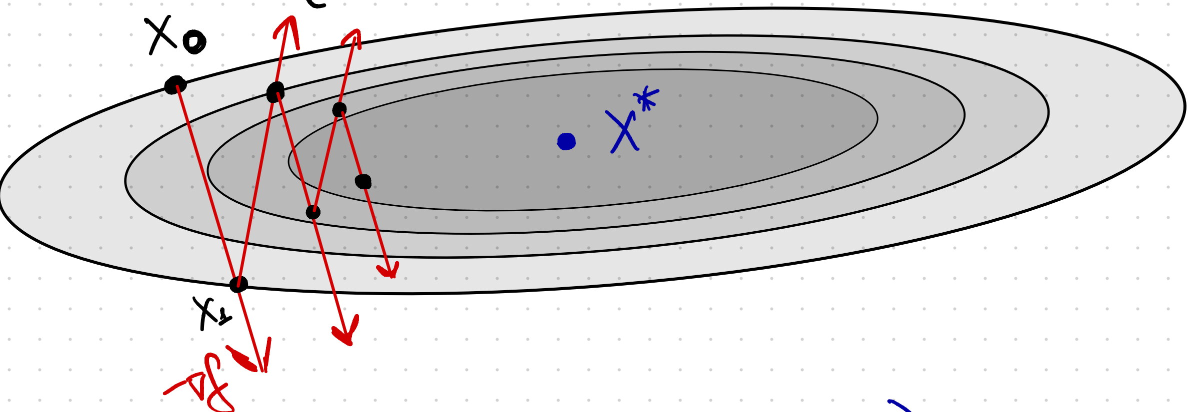
Ускоренные методы:

Heavy ball: (1) $x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$

$$\begin{cases} V_k = -\nabla f(x_k) + \beta \cdot V_{k-1} \\ x_{k+1} = x_k + \alpha \cdot V_k \end{cases}$$

$0 < \beta < 1$

MOMENTUM

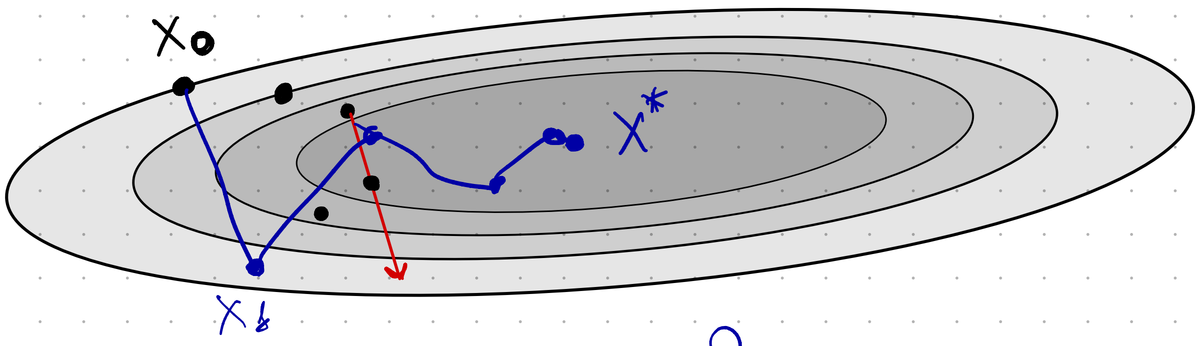


$x_{k+1} = x_k + \alpha \cdot V_k = x_k + \alpha\left(-\nabla f(x_k) + \beta V_{k-1}\right) =$

$= x_k + \alpha\left(-\nabla f(x_k) + \beta\left(-\nabla f(x_{k-1}) + \beta \cdot V_{k-2}\right)\right) =$

$= x_k + \alpha\left(-\nabla f(x_k) + \beta\left(-\nabla f(x_{k-1}) + \beta\left(-\nabla f(x_{k-2}) + \beta V_{k-3}\right)\right)\right)$

$$x_k - \alpha\left(\nabla f(x_k) + \beta \nabla f(x_{k-1}) + \beta^2 \nabla f(x_{k-2}) + \beta^3 \nabla f(x_{k-3}) + \dots\right)$$



Как выбирать α и β?

Практика: α как и в GD; $\beta \approx 0.99, 0.9$

Теория:

$$\lambda_{min}\left(\nabla^2 f(x)\right) = \mu \qquad \alpha = \frac{4}{(\sqrt{\mu}+\sqrt{L})^2}$$

$$\lambda_{max}\left(\nabla^2 f(x)\right) = L \qquad \beta = \frac{(\sqrt{L}-\sqrt{\mu})^2}{(\sqrt{L}+\sqrt{\mu})^2}$$

Nesterov Accelerated Gradient:

NESTEROV MOMENTU

$$\begin{cases} y_{k+1} = x_k + \beta(x_k - x_{k-1}) \\ x_{k+1} = x_k - \alpha \nabla f(y_{k+1}) \end{cases}$$

Теория: $\alpha = \frac{1}{L}$

$$\beta = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$$

$0 < \mu \le \lambda\left(\nabla^2 f(x)\right) \le L$

для сильно вып. функций

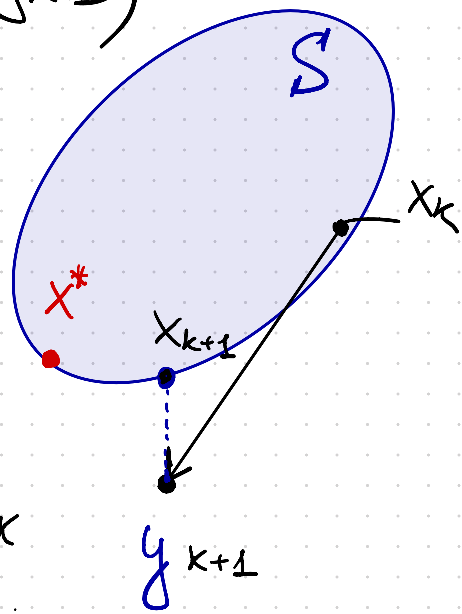
БЫСТРЕЕ НЕЛЬЗЯ!

Метод проекции градиента.

$$y_{k+1} = \text{Метод}(x_k, x_{k-1}, \alpha, \beta)$$

$$x_{k+1} = \Pi_S(y_{k+1})$$

↑ проекция



- С точки зрения кол-ва итераций методы, описанные выше, сходятся точно так же, как и в безусловной задаче. (если мн-во S — выпуклое и замкнутое)
- Но стоимость 1 итерации может стать очень большой из-за проекции

$$\Pi_S(y) = \arg\min_{x \in S} \|y - x\|^2$$

Nocedal Wright
Numerical Optimization

Intuition

Newton's method to find the equation' roots

Consider the function $\varphi(x) : \mathbb{R} \to \mathbb{R}$. Let there be equation $\varphi(x^*) = 0$. Consider a linear approximation of the function $\varphi(x)$ near the solution ($x^* - x = \Delta x$):

$$\varphi(x^*) = \varphi(x + \Delta x) \approx \varphi(x) + \varphi'(x)\Delta x.$$

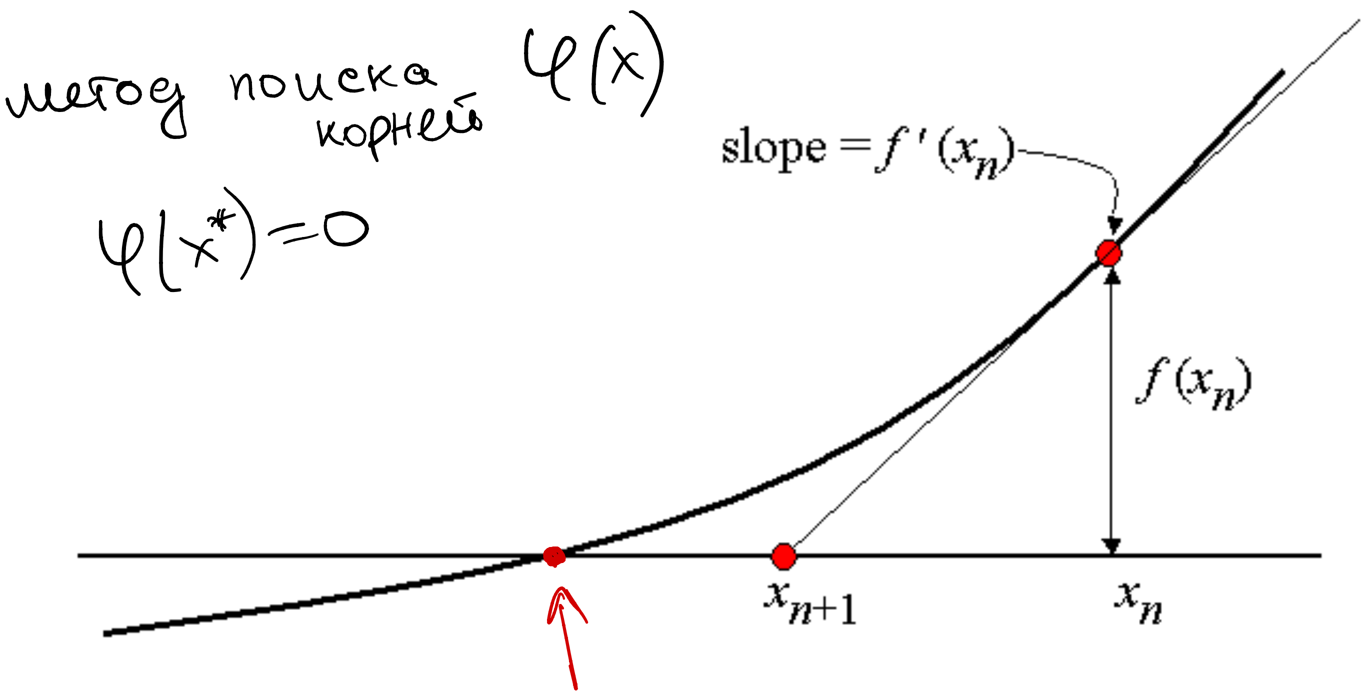
We get an approximate equation:

$$\varphi(x) + \varphi'(x)\Delta x = 0$$

We can assume that the solution to equation $\Delta x = -\dfrac{\varphi(x)}{\varphi'(x)}$ will be close to the optimal $\Delta x^* = x^* - x$.

We get an iterative scheme:

$$x_{k+1} = x_k - \frac{\varphi(x_k)}{\varphi'(x_k)}.$$



This reasoning can be applied to the unconditional minimization task of the $f(x)$ function by writing down the necessary extremum condition:

$$f'(x^*) = 0$$

Here $\varphi(x) = f'(x),\ \varphi'(x) = f''(x)$. Thus, we get the Newton optimization method in its classic form:

$x_{k+1} = x_k + d_{new} \rightarrow d_{new} = x_{k+1} - x_k$

$$x_{k+1} = x_k - \left[f''(x_k)\right]^{-1} f'(x_k). \qquad \text{(Newton)}$$

With the only clarification that in the multidimensional case:
$x \in \mathbb{R}^n,\ f'(x) = \nabla f(x) \in \mathbb{R}^n,\ f''(x) = \nabla^2 f(x) \in \mathbb{R}^{n \times n}$.

$x_{k+1} - x_k = -[\nabla^2 f]^{-1} \cdot \nabla f$

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

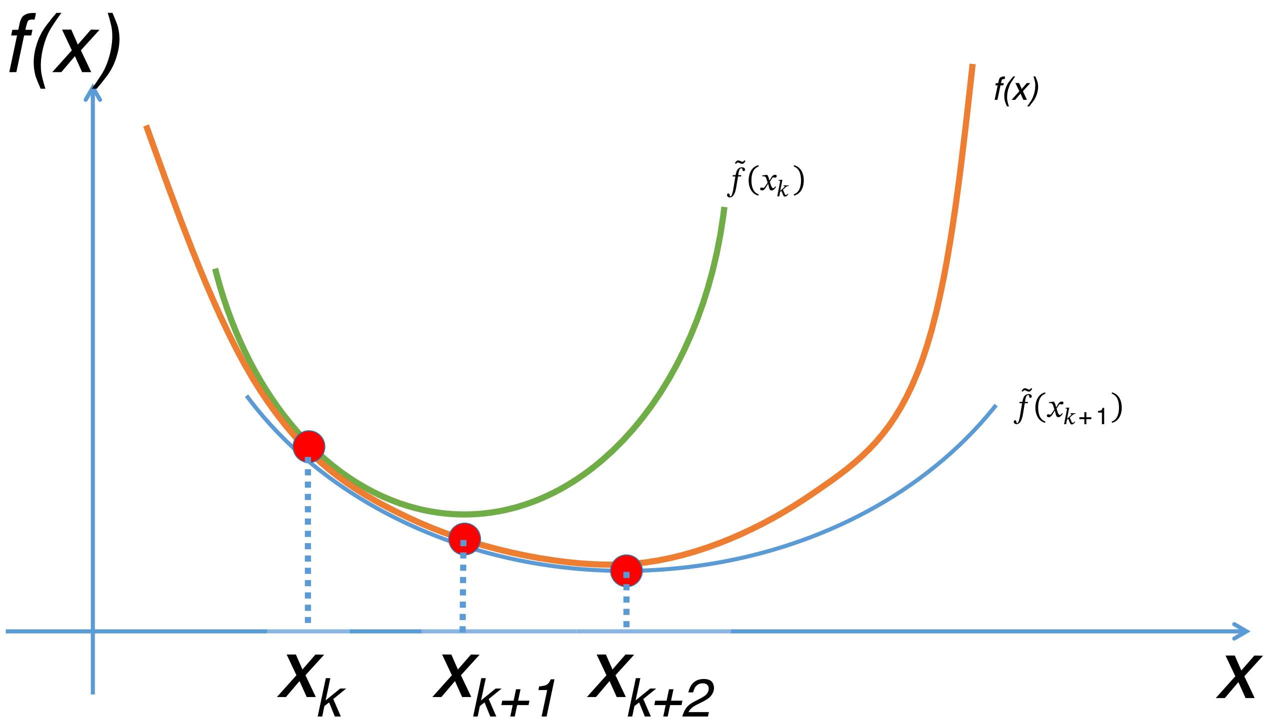
$\nabla^2 f(x_k) \cdot d_{new} = -\nabla f(x_k)$

Second order Taylor approximation of the function

Let us now give us the function $f(x)$ and a certain point x_k. Let us consider the square approximation of this function near x_k:

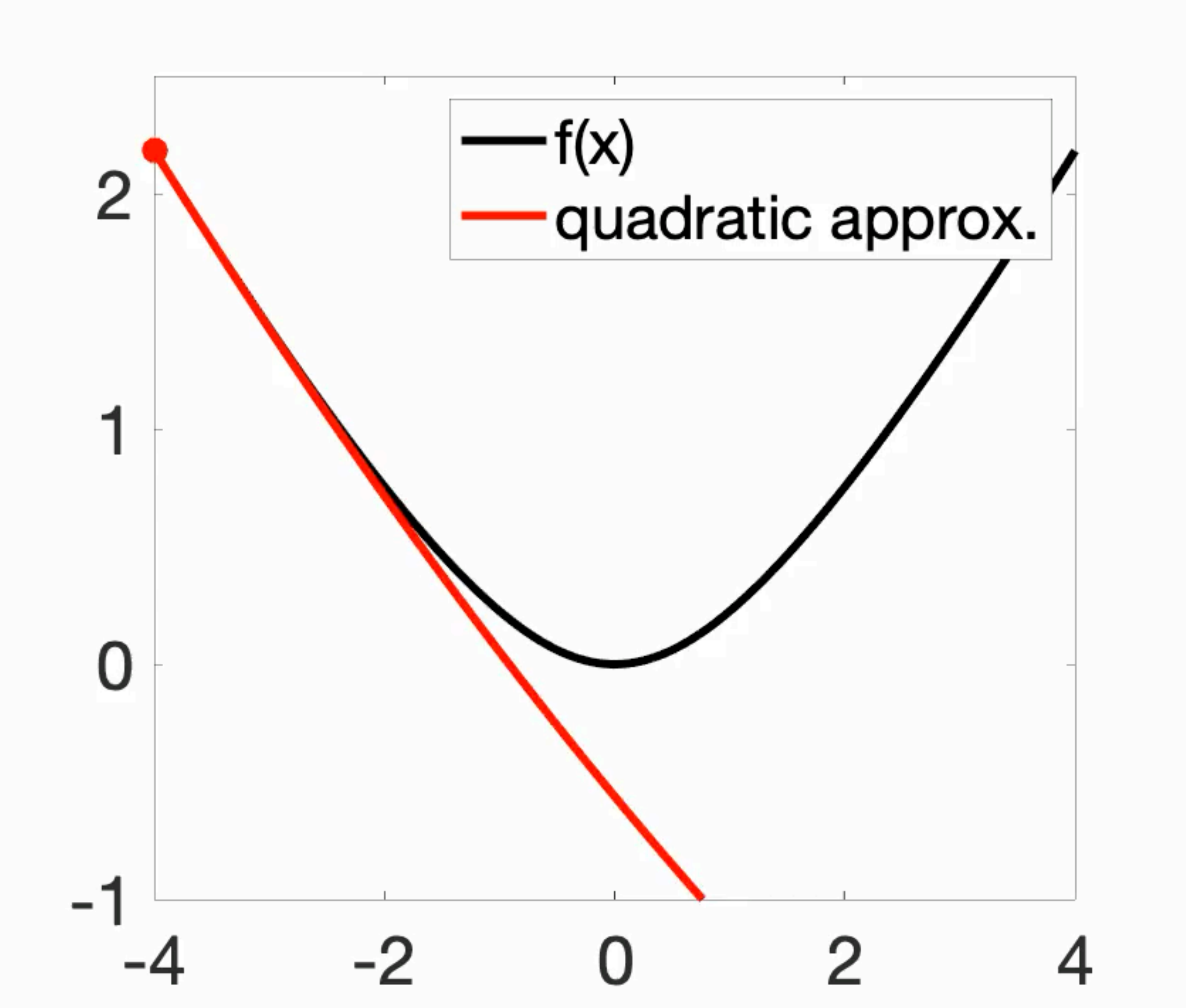
$$\tilde{f}(x) = f(x_k) + \langle f'(x_k), x - x_k \rangle + \frac{1}{2} \langle f''(x_k)(x - x_k), x - x_k \rangle.$$

The idea of the method is to find the point x_{k+1}, that minimizes the function $\tilde{f}(x)$, i.e. $\nabla \tilde{f}(x_{k+1}) = 0$.



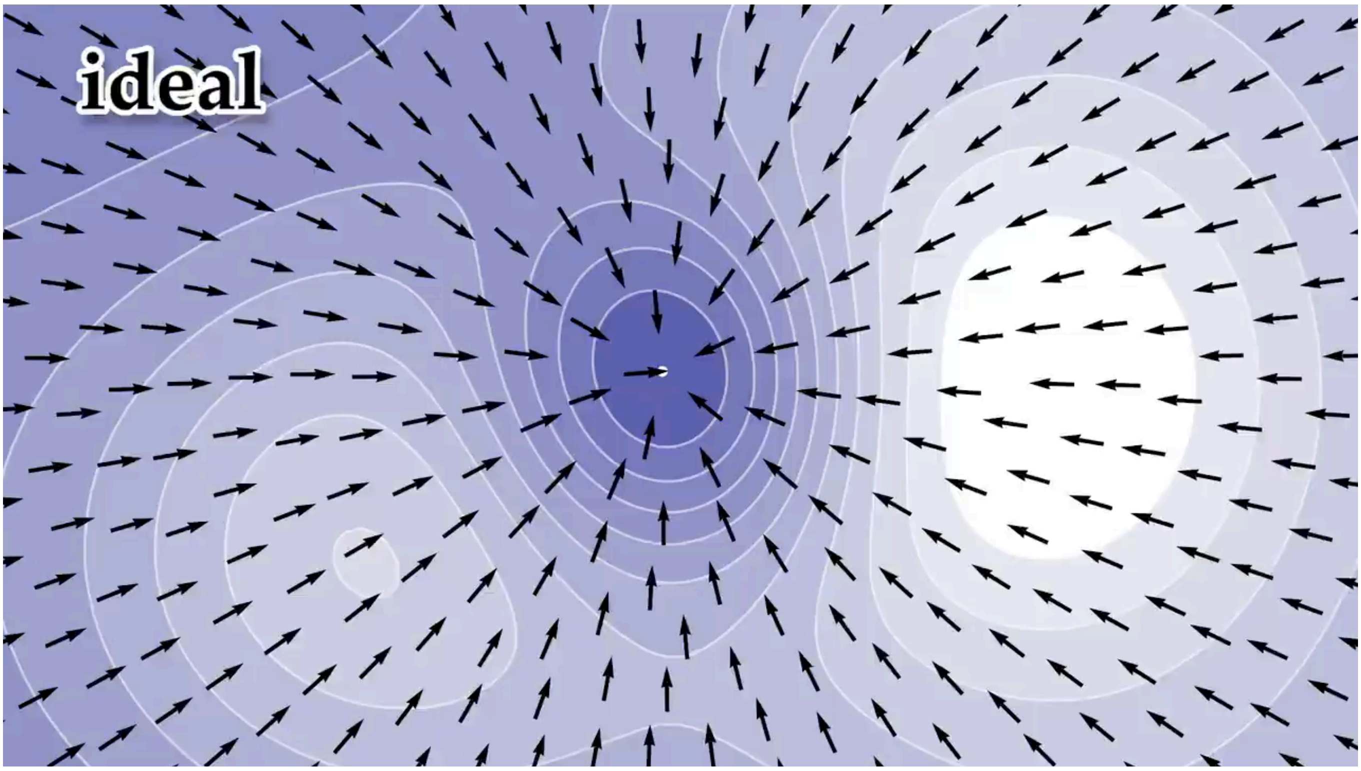
$$\begin{aligned}
\nabla \tilde{f}(x_{k+1}) &= f'(x_k) + f''(x_k)(x_{k+1} - x_k) = 0 \\
f''(x_k)(x_{k+1} - x_k) &= -f'(x_k) \\
\left[f''(x_k)\right]^{-1} f''(x_k)(x_{k+1} - x_k) &= -\left[f''(x_k)\right]^{-1} f'(x_k) \\
x_{k+1} &= x_k - \left[f''(x_k)\right]^{-1} f'(x_k).
\end{aligned}$$

Let us immediately note the limitations related to the necessity of the Hessian’s non-degeneracy (for the method to exist), as well as its positive definiteness (for the convergence guarantee).



Quadratic approximation and Newton step (in green) for varying starting points (in red). Note that when the starting point is far from the global minimizer (in 0), the Newton step totally overshoots the global minimizer. Picture was taken from the post.

Convergence



Let's try to get an estimate of how quickly the classical Newton method converges. We will try to enter the necessary data and constants as needed in the conclusion (to illustrate the methodology of obtaining such estimates).

$$
\begin{aligned}
x_{k+1} - x^* = x_k - \left[f''(x_k)\right]^{-1} f'(x_k) - x^* = x_k - x^* - \left[f''(x_k)\right]^{-1} f'(x_k) = \\
= x_k - x^* - \left[f''(x_k)\right]^{-1} \int_0^1 f''(x^* + \tau(x_k - x^*))(x_k - x^*) d\tau = \\
= \left(1 - \left[f''(x_k)\right]^{-1} \int_0^1 f''(x^* + \tau(x_k - x^*)) d\tau\right)(x_k - x^*) = \\
= \left[f''(x_k)\right]^{-1} \left(f''(x_k) - \int_0^1 f''(x^* + \tau(x_k - x^*)) d\tau\right)(x_k - x^*) = \\
= \left[f''(x_k)\right]^{-1} \left(\int_0^1 \left(f''(x_k) - f''(x^* + \tau(x_k - x^*)) d\tau\right)\right)(x_k - x^*) = \\
= \left[f''(x_k)\right]^{-1} G_k (x_k - x^*)
\end{aligned}
$$

Used here is: $G_k = \int_0^1 \left(f''(x_k) - f''(x^* + \tau(x_k - x^*)) d\tau\right)$. Let's try to estimate the size of G_k:

$$\|G_k\| = \left\| \int_0^1 \left(f''(x_k) - f''(x^* + \tau(x_k - x^*))d\tau\right) \right\| \leq$$

$$\leq \int_0^1 \|f''(x_k) - f''(x^* + \tau(x_k - x^*))\| d\tau \leq \qquad \text{(Hessian's Lipschitz continuity)}$$

$$\leq \int_0^1 M\|x_k - x^* - \tau(x_k - x^*)\| d\tau = \int_0^1 M\|x_k - x^*\|(1-\tau)d\tau = \frac{r_k}{2}M,$$

where $r_k = \|x_k - x^*\|$.

So, we have:

$$r_{k+1} \leq \left\| \left[f''(x_k)\right]^{-1} \right\| \cdot \frac{r_k}{2} M \cdot r_k$$

Already smells like quadratic convergence. All that remains is to estimate the value of Hessian's reverse.

Because of Hessian's Lipschitz continuity and symmetry:

$$\begin{aligned} f''(x_k) - f''(x^*) &\succeq -Mr_k I_n \\ f''(x_k) &\succeq f''(x^*) - Mr_k I_n \\ f''(x_k) &\succeq \mu I_n - Mr_k I_n \\ f''(x_k) &\succeq (\mu - Mr_k) I_n \end{aligned}$$

So, (here we should already limit the necessity of being $f''(x_k) \succ 0$ for such estimations, i.e. $r_k < \frac{\mu}{M}$).

$$\left\| \left[f''(x_k)\right]^{-1} \right\| \leq (\mu - Mr_k)^{-1}$$

$$r_{k+1} \leq \frac{r_k^2 M}{2(\mu - Mr_k)}$$

The convergence condition $r_{k+1} < r_k$ imposes additional conditions on $r_k: \quad r_k < \frac{2\mu}{3M}$

Thus, we have an important result: Newton's method for the function with Lipschitz positive Hessian converges **quadratically** near ($\|x_0 - x^*\| < \frac{2\mu}{3M}$) to the solution.

Theorem

Let $f(x)$ be a strongly convex twice continuously differentiated function at $\mathbb{R}^n$, for the second derivative of which inequalities are executed: $\mu I_n \preceq f''(x) \preceq L I_n$. Then

Newton's method with a constant step locally converges to solving the problem with superlinear speed. If, in addition, Hessian is Lipschitz continuous, then this method converges locally to x^* at a quadratic rate.

Summary

It's nice:

- quadratic convergence near the solution x^*
- affinity invariance
- the parameters have little effect on the convergence rate

It's not nice:

- it is necessary to store the hessian on each iteration: $\mathcal{O}(n^2)$ memory
- it is necessary to solve linear systems: $\mathcal{O}(n^3)$ operations
- the Hessian can be degenerate at x^*
- the hessian may not be positively determined $\to$ direction $-(f''(x))^{-1}f'(x)$ may not be a descending direction

Possible directions

- Newton's damped method (adaptive stepsize)
- Quasi-Newton methods (we don't calculate the Hessian, we build its estimate - BFGS)
- Quadratic evaluation of the function by the first order oracle (superlinear convergence)
- The combination of the Newton method and the gradient descent (interesting direction)
- Higher order methods (most likely useless)

Materials

- Going beyond least-squares – I : self-concordant analysis of Newton method
- Going beyond least-squares – II : Self-concordant analysis for logistic regression

- Picture with gradient and Newton field was taken from this tweet by Keenan Crane.
- About global damped Newton convergence issue. Open in Colab

Code

Open in Colab

Идея:
вместо гессиана использовать
его аппроксимацию.
Например, $H_0 = I$

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \cdot \nabla f(x_k)$$

Newton → GD с $\alpha = 1$

$H_1 = H_0 + \Delta H_0$ ← добавка, учитывающая накопл. статистику

BFGS

Intuition

For the classic task of unconditional optimization $f(x) \to \min\limits_{x \in \mathbb{R}^n}$ the general scheme of iteration method is written as:

$$x_{k+1} = x_k + \alpha_k s_k$$

In the Newton method, the s_k direction (Newton's direction) is set by the linear system solution at each step:

$$s_k = -B_k \nabla f(x_k), \quad B_k = f_{xx}^{-1}(x_k)$$

i.e. at each iteration it is necessary to **compensate** hessian and gradient and **resolve** linear system.

Note here that if we take a single matrix of $B_k = I_n$ as B_k at each step, we will exactly get the gradient descent method.

The general scheme of quasi-Newton methods is based on the selection of the B_k matrix so that it tends in some sense at $k \to \infty$ to the true value of inverted Hessian in the local optimum $f_{xx}^{-1}(x_*)$. Let's consider several schemes using iterative updating of B_k matrix in the following way:

$$B_{k+1} = B_k + \Delta B_k$$

Then if we use Taylor's approximation for the first order gradient, we get it:

$$\nabla f(x_k) - \nabla f(x_{k+1}) \approx f_{xx}(x_{k+1})(x_k - x_{k+1}).$$

Now let's formulate our method as:

$$\Delta x_k = B_{k+1} \Delta y_k, \text{ where } \quad \Delta y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

in case you set the task of finding an update ΔB_k:

$$\Delta B_k \Delta y_k = \Delta x_k - B_k \Delta y_k$$

Broyden method

The simplest option is when the amendment ΔB_k has a rank equal to one. Then you can look for an amendment in the form

$$\Delta B_k = \mu_k q_k q_k^\top.$$

where μ_k is a scalar and q_k is a non-zero vector. Then mark the right side of the equation to find ΔB_k for Δz_k:

$$\Delta z_k = \Delta x_k - B_k \Delta y_k$$

We get it:

$$\mu_k q_k q_k^\top \Delta y_k = \Delta z_k$$

$$\left(\mu_k \cdot q_k^\top \Delta y_k\right) q_k = \Delta z_k$$

A possible solution is: $q_k = \Delta z_k$, $\mu_k = \left(q_k^\top \Delta y_k\right)^{-1}$.

Then an iterative amendment to Hessian's evaluation at each iteration:

$$\Delta B_k = \frac{(\Delta x_k - B_k \Delta y_k)(\Delta x_k - B_k \Delta y_k)^\top}{\langle \Delta x_k - B_k \Delta y_k, \Delta y_k \rangle}.$$

Davidon–Fletcher–Powell method

$\Delta B_k = \mu_1 \Delta x_k (\Delta x_k)^\top + \mu_2 B_k \Delta y_k (B_k \Delta y_k)^\top.$

$$\Delta B_k = \frac{(\Delta x_k)(\Delta x_k)^\top}{\langle \Delta x_k, \Delta y_k \rangle} - \frac{(B_k \Delta y_k)(B_k \Delta y_k)^\top}{\langle B_k \Delta y_k, \Delta y_k \rangle}.$$

Broyden–Fletcher–Goldfarb–Shanno method

$$\Delta B_k = Q U Q^\top, \quad Q = [q_1, q_2], \quad q_1, q_2 \in \mathbb{R}^n, \quad U = \begin{pmatrix} a & c \\ c & b \end{pmatrix}.$$

$$\Delta B_k = \frac{(\Delta x_k)(\Delta x_k)^\top}{\langle \Delta x_k, \Delta y_k \rangle} - \frac{(B_k \Delta y_k)(B_k \Delta y_k)^\top}{\langle B_k \Delta y_k, \Delta y_k \rangle} + p_k p_k^\top.$$

Code

- Open in Colab
- Comparison of quasi Newton methods