Non Convex Optimization for Data Science

Gasnikov Alexander

gasnikov.av@mipt.ru

Lecture 1. Introduction Lecture in Non Convex Optimization

Huawei

July 29, 2019

$x^0=u^0,\ A_0=\alpha_0$
$\alpha_{k+1}=\frac{1}{2L}+\sqrt{\frac{1}{4L^2}+\alpha_k^2},\ A_{k+1}=A_k+\alpha_{k+1}$
$y^{k+1}=\frac{\alpha_{k+1}u^k+A^kx^k}{A^{k+1}},\ u^k=\arg\min$

Гасников Александр Владимирович

- доктор физико-математических наук
- доцент кафедры Математических основ управления ФУПМ ПМИ МФТИ

https://arxiv.org/ftp/arxiv/papers/1711/1711.00394.pdf

(the most of the references below are from this book)

Main references:

Bottou L., Curtis F.E., Nocedal J. Optimization methods for large-scale machine learning // arXiv.org e-Print archive. 2016. – URL: https://arxiv.org/pdf/1606.04838.pdf

Jain P., Kar P. Non-convex optimization for machine learning // arXiv.org e-Print archive. 2017. – URL: https://arxiv.org/pdf/1712.07897.pdf

Lan G. Lectures on optimization. Methods for Machine Learning // e-print, 2019. – URL: http://pwp.gatech.edu/guanghui-lan/wp-content/uploads/sites/330/2019/02/LectureOPTML.pdf

Ruder A. An overview of gradient descent optimization algorithms // arXiv.org e-Print archive. 2017. – URL: https://arxiv.org/pdf/1609.04747.pdf

Wright S. Optimization algorithms for Data Science // IAS/Park City Mathematics Series. – 2016. URL: http://www.optimization-online.org/DB_FILE/2016/12/5748.pdf

Structure of Lecture 1

- Gradient descent (GD)
- Convergence to local minimum
- Penalty function
- Global optimization is NP-hard
- The meaning of criteria
- Lower bound for global optimization
- How we should find global minimum in practice?
- Practical acceleration by partially convex behavior of target function
- Stochastic optimization
- Sum-type minimization

Gradient descent

$$\boxed{f\left(x\right)\to\min_{x\in\mathbb{R}^n}.}$$

$$\frac{dx}{dt}=-\nabla f\left(x\right).$$

Let's show that $W\left(x\right)=f\left(x\right)$ is Lyapunov function:

$$\frac{dW\left(x(t)\right)}{dt}=\left\langle\nabla f\left(x(t)\right),\frac{dx(t)}{dt}\right\rangle=$$

$$,$$

$$=\left\langle\nabla f\left(x(t)\right),-\nabla f\left(x(t)\right)\right\rangle=-\left\|\nabla f\left(x(t)\right)\right\|_2^2\le 0.$$

Gradient descent

$$f\left(x\right) \to \min_{x\in\mathbb{R}^n}.$$

Euler discretization of $dx/dt = -\nabla f\left(x\right)$ has the form

$$\boxed{x^{k+1} = x^k - h\nabla f\left(x^k\right).}$$

What is about h ? Assume that for $x, y \in \left\{x \in \mathbb{R}^n : f\left(x\right) \leq f\left(x^0\right)\right\}$

$$\left\|\nabla f\left(y\right) - \nabla f\left(x\right)\right\|_2 \leq L\left\|y - x\right\|_2.$$

Then (this result can also be obtained from physical dimension considerations)

$$\boxed{h = 1/L.}$$

Gradient descent

Since

$$f\left(x^{k+1}\right) \leq f\left(x^{k}\right) - \frac{1}{2L}\left\|\nabla f\left(x^{k}\right)\right\|_2^2$$

we can conclude that (ε-extremum)

$$\boxed{\min_{k=1,\ldots,N}\left\|\nabla f\left(x^{k}\right)\right\|_2 \leq \varepsilon}$$

when

$$N = \frac{2L\cdot\left(f\left(x^{0}\right) - f\left(x^{extr}\right)\right)}{\varepsilon^2}.$$

If there available inexact gradient and (or) we have weaker assumption on smoothness, see P. Dvurechensky (2017) and references therein.

Gradient descent

This bound

$$N \sim \varepsilon^{-2} \text{ (first-order method, Lipschitz gradient)}$$

is unimprovable up to a constant factor for arbitrary first-order method. But if we assume high-order smoothness of $f(x)$, then lower bound for ε-extremum will have the form (*Carmon Y., Duchi J.C., Hinder O., Sidford A., 2017*)

$$N \sim \varepsilon^{-8/5} \text{ (first-order method, Lipschitz high-order derivatives).}$$

Moreover, if we additionally can use p-order derivatives in algorithm, then optimal bound will have the form (*Birgin E., Gardenghi J., Martinez J. et al., 2016*)

$$N \sim \varepsilon^{-(p+1)/p} \text{ (} p\text{-order method, Lipschitz } p\text{-order derivatives).}$$

Gradient descent

Let’s indicate the main drawbacks of these types of results:

1. The results are not global. Moreover, we cannot guarantee that we can find even local minimum. Indeed (Yu. Nesterov, 2004), for

$$f\left(x_1,x_2\right)=\frac{1}{2}\left(x_1\right)^2+\frac{1}{2}\left(x_2\right)^4-\frac{1}{2}\left(x_2\right)^2$$

   choose $x^0=\left(1,0\right)^T$, then $x^k\xrightarrow[k\to\infty]{}\left(0,0\right)^T$ – saddle-point.

2. We consider unconstraint optimization problem. It’s significant! For example, if we consider $f\left(x\right)\to\min\limits_{x\in\left[0,1\right]^n}$, the lower bound for ε-extremum has the form $N\sim\varepsilon^{-n/2}$ (Yu. Nesterov, 2012).

Gradient descent

The good news here is that GD (and other GD type methods) is typically (for almost all starting point) converges to local minimum and GD with small noise is also typically swing across saddle-point and stop at local minimum point (J.D. Lee, M.I. Jordan. B. Recht et al., 2016–2017, see also Langevin dynamics below). Moreover, by using gradient mapping and idea of composite optimization one can generalize the results obtained for unconstraint case to constraint ones with simple constraints (P. Dvurechensky, 2017).

But the really bad news is that the problem of finding global optimum is provably very difficult in general (A.S. Nemirovski, 1979).

Convergence to local minimum

$$f\left(x\right)\to\min_{x\in\mathbb{R}^n}.$$

Following by Yu. Nesterov and B. Polyak (2006) let's denote $x^N - \left(\varepsilon,\delta\right)$ local minimum if

$$\left\|\nabla f\left(x^N\right)\right\|_2\le\varepsilon,\ \lambda_{\min}\left(\nabla^2 f\left(x^N\right)\right)\ge-\delta.$$

Assume that for $x,y\in\left\{x\in\mathbb{R}^n:\ f\left(x\right)\le f\left(x^0\right)\right\}$

$$\left\|\nabla f\left(y\right)-\nabla f\left(x\right)\right\|_2\le L_1\left\|y-x\right\|_2,$$

$$\left\|\nabla^2 f\left(y\right)-\nabla^2 f\left(x\right)\right\|_2\le L_2\left\|y-x\right\|_2.$$

Convergence to local minimum

$$x^{k+1} = x^k - \frac{1}{L_1}\nabla f\left(x^k\right), \text{ if } \left\|\nabla f\left(x^k\right)\right\|_2 > \varepsilon;$$

$$x^{k+1} = x^k + hp^k, \text{ if } \left\|\nabla f\left(x^k\right)\right\|_2 \le \varepsilon \text{ and } \lambda_{\min}^k = \lambda_{\min}\left(\nabla^2 f\left(x^k\right)\right) < -\delta,$$

where $h = 2\delta/L_2$, p^k – eigen vector of $\nabla^2 f\left(x^k\right)$ corresponds to $\lambda_{\min}^k$:

$$\left\langle \nabla f\left(x^k\right), p^k \right\rangle \le 0, \ \left\|p^k\right\|_2 = 1.$$

This method will stop (find $\left(\varepsilon,\delta\right)$ local minimum) after N iterations

$$N \le N\left(\varepsilon,\delta\right) = \left(f\left(x^0\right) - f\left(x^{\text{local min}}\right)\right)\cdot \max\left\{\frac{2L_1}{\varepsilon^2}, \frac{3L_2^2}{2\delta^3}\right\}.$$

This is not an optimal method, but it is good for explanation the idea.

Convergence to local minimum

Indeed,

$$f\left(x^{k+1}\right) \le f\left(x^k\right) + h\underbrace{\left\langle \nabla f\left(x^k\right), p^k \right\rangle}_{\le 0} + \frac{h^2}{2!}\underbrace{\left\langle \nabla^2 f\left(x^k\right) p^k, p^k \right\rangle}_{\le -\frac{h^2\delta}{2}} + \frac{h^3}{3!} L_2 \underbrace{\left\| p^k \right\|_2^3}_{=M_2} \le$$

$$\le f\left(x^k\right) - \frac{1}{2}\left(\frac{2\delta}{L_2}\right)^2 \delta + \frac{L_2}{6}\left(\frac{2\delta}{L_2}\right)^3 = f\left(x^k\right) - \frac{2}{3}\frac{\delta^3}{L_2^2}.$$

Note, that it's sufficiently to indicate that $\lambda_{\min}\left(\nabla^2 f\left(x^k\right)\right) \ge -\delta$, or to find such p^k ($\left\| p^k \right\|_2 = 1$), that $\left\langle \nabla^2 f\left(x^k\right) p^k, p^k \right\rangle \le -\delta/2$. One can do it by Lanczos method with $\sim 1/\sqrt{\delta}$ matrix-vector $\nabla^2 f\left(x^k\right) p$ multiplications.

Convergence to local minimum

Note, that by using (Krylov's type methods)

$$\nabla^2 f(x)\mathrm{v} = \nabla\left\langle \nabla f(x), \mathrm{v}\right\rangle,$$

$$\nabla^2 f(x)\mathrm{v} \approx \frac{\nabla f(x+\tau \mathrm{v}) - \nabla f(x)}{\tau}$$

and automatic differentiations one can calculate $f(x)$, $\nabla f(x)$ and $\nabla^2 f(x)\mathrm{v}$ for almost the same time (up to a constant 4 – 16) for all reasonable function with known computational tree.

Note: Automatic differentiation is now widespread: *Torch*, *Caffe*, *Theano*, *TensorFlow*. But only the last two libraries support the desired high-order differentiation.

Convergence to local minimum

One can improve the ε-part of the bound on $N\left(\varepsilon,\delta\right)$ from $\sim\varepsilon^{-2}$ to $\sim\varepsilon^{-3/2}$ by using cubic regularized Newton method (Yu. Nesterov, B. Polyak, 2006; Grapilia–Netserov, 2019). For that, we need to replace

$$x^{k+1}=x^k-\frac{1}{L_1}\nabla f\left(x^k\right)=\arg\min_{x\in\mathbb{R}^n}\left\{f\left(x^k\right)+\left\langle\nabla f\left(x^k\right),x-x^k\right\rangle+\frac{L_1}{2!}\left\|x-x^k\right\|_2^2\right\}$$

by (the complexity of this iteration is comparable with Newton's iteration)

$$x^{k+1}=\arg\min_{x\in\mathbb{R}^n}\left\{f\left(x^k\right)+\left\langle\nabla f\left(x^k\right),x-x^k\right\rangle+\frac{1}{2!}\left\langle\nabla^2 f\left(x^k\right)\left(x-x^k\right),x-x^k\right\rangle+\frac{L_2}{3!}\left\|x-x^k\right\|_2^3\right\}.$$

Recall that for second-order methods this bound $\sim\varepsilon^{-3/2}$ is unimrovable. Note, that for cubic regularized Newton method we can skip $x^{k+1}=x^k+hp^k$.

Penalty function (reduction to unconstraint optimization)

One of the ways to reduce constraint optimization problem, say (this is rather general form; analogously one can consider $g(x) \leq 0$)

$$f(x) \to \min_{g(x)=0},$$

to unconstraint one

$$F(x) \to \min_{x \in \mathbb{R}^n}$$

is penalty function approach. The basic idea is to solve another problem:

$$F(x) = f(x) + \frac{K}{2}\|g(x)\|_2^2 \to \min_{x \in \mathbb{R}^n} \text{ (} x^K \text{ – is solution).}$$

Penalty function and vicinities

To interpret K let's relax initial problem

$$f\left(x\right)\to \min_{\frac{1}{2}\left\|g\left(x\right)\right\|_2^2\le\frac{1}{2}\varepsilon^2}$$

and put $K := K\left(\varepsilon\right)$ – Lagrange multiplier to $\frac{1}{2}\left\|g\left(x\right)\right\|_2^2 \le \frac{1}{2}\varepsilon^2$.

So if λ – Lagrange multiplier to $g\left(x\right)=0$ in initial problem, then

$$Kg\left(x^K\right)\xrightarrow[K\to\infty]{}\lambda, \text{ i.e. } g\left(x^K\right)\simeq\frac{\lambda}{K},\ f\left(x^K\right)-f\left(x_*\right)=\mathrm{O}\left(\frac{\left\|\lambda\right\|_2^2}{K}\right).$$

Feasibility problem: Brigin–Bueno–Martinez, 2019.

ε-KKT point (proximal point method): Boob–Deng–Lan, 2019.

Riemannian manifolds: S. Sra et al.; Ferreira–Louzeiro–Prudente, 2018.

Global optimization is NP-hard (K. Murty, S. Kabadi, 1987)

$$f\left(x\right)=\sum_{i=1}^{n}x_i^4-\frac{1}{n}\left(\sum_{i=1}^{n}x_i^2\right)^2+\left(\sum_{i=1}^{n}a_i x_i\right)^4+\left(1-x_1\right)^4\to\min_{x\in\mathbb{R}^n},$$

equivalent to minimization of convex polynomial on a unit sphere

$$P_4\left(x\right)=\sum_{i=1}^{n}x_i^4+\left(\sum_{i=1}^{n}a_i x_i\right)^4+\left(1-x_1\right)^4\to\min_{x\in\mathbb{R}^n:\ \|x\|_2=1}.$$

The last problem can be equivalently reformulated as knapsack problem:

$$a_1+\sum_{i=2}^{n}a_i x_i=0,\ x_i=\pm1.$$

By using FFT this problem can be solved by $\mathrm{O}\left(\ln n\cdot\sum_{i=1}^{n}|a_i|\right)$ a.o., but it’s NP.

Lower bound for global optimization

$$f\left(x\right)\to \min_{x\in\left[0,1\right]^n} .$$

Assume that for $x, y \in \left[0,1\right]^n$

$$\left\|\nabla f\left(y\right)-\nabla f\left(x\right)\right\|_2 \le L\left\|y-x\right\|_2 .$$

Let’s divide cube $\left[0,1\right]^n$ on small sub-cubes with side length

$$4\sqrt{2\varepsilon/L} .$$

Let’s put $f\left(x\right)\equiv 0$ everywhere except one sub-cubes that was observed by considered algorithm at the very end.

Lower bound for global optimization

In this sub-cube we can determine $f(x)$ such that:

$$\min_{x\in[0,1]^n} f(x) = -2\varepsilon .$$

So to find such x^N that guarantee (ε-solution)

$$\boxed{f\left(x^N\right) - f\left(x_*\right) \le \varepsilon}$$

algorithm need to visit each sub-cubes where $f(x) \equiv 0$ at least on time. Therefore the lower bound on N is

$$\boxed{N \sim \left(1/\sqrt{\varepsilon}\right)^n = \varepsilon^{-n/2}.}$$

The meaning of criteria

For the moment for

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

we've considered two criteria of quality of approximate solution:

$$\min_{k=1,\dots,N} \left\| \nabla f\left(x^k\right) \right\|_2 \le \varepsilon$$

and

$$f\left(x^N\right) - f\left(x_*\right) \le \varepsilon.$$

It seems that if $f(x)$ has unique extremum = local minimum (that is consequently global minimum), then these criteria are close to each other.

The meaning of criteria

But this is not true! (Yu. Nesterov, V. Skokov, 1980)

$$f(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n}\left(x_{i+1} - 2x_i^2 + 1\right)^2 =$$

$$= \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n}(x_{i+1} - \underbrace{P_2(x_i)}_{\substack{\text{Chebyshev}\\ \text{polynom}}})^2.$$

At point

$$x_* = (1,1,\ldots,1) \; (f(x_*) = 0),$$

we have unique extremum = global minimum of $f(x)$.

The meaning of criteria

If we put

$$x^0 = \left(-1,1,...,1\right)^T \ (\ f\left(x^0\right) = 1),$$

then gradient type methods guarantee small norm of the gradient, but not small discrepancy in function value. In all experiments (with different GD type methods and second-order type methods) under $n = 15$ we have

$$\left\|\nabla f\left(x^N\right)\right\|_2 \approx 10^{-8},$$

but

$$f\left(x^N\right) - f\left(x_*\right) \approx 1/2.$$

How we should find global minimum in practice?

1. Local algorithms + multistart
2. Multidimensional bisection (Yu. Evtushenko, R. Brent, G. Wood)
3. Markov search (simulated annealing)

But first two approach can be efficiently used in practice generally only in dimensions: $n \sim 10 - 10^2$.

Details see in books of A. Zilinskas and A. Zhigljavsky.

Local algorithms + multistart

$$f(x) \to \min_{x \in [0,1]^n} .$$

If the measure of pool of attraction of global minimum is $p > 0$, then it's sufficiently to choose (up to a logarithmic factors)

$$m = \tilde{O}(1/p)$$

random starting points $\left\{x^{0,i}\right\}_{i=1}^{m}$.

The quality of network $\left\{x^{0,i}\right\}_{i=1}^{m}$ can be also characterized by

$$d_n\left(\left\{x^{0,i}\right\}_{i=1}^{m}\right) = \max_{x \in [0,1]^n} \min_{i=1,\dots,m} \left\|x - x^{0,i}\right\|_2 = O\left(\sqrt{n} m^{-1/(2n)}\right).$$

Local algorithms + multistart (quasi Monte Carlo)

One of the best known way to generate points $\left\{x^{0,k}\right\}_{k=1}^{m}$ adaptively on m is quasi Monte Carlo scheme.

Let $\left\{p_i\right\}_{i=1}^{n}$ – sequence of different prim numbers.

Let $\varphi_{p_i}\left(k\right)=\sum_{j=0}^{l_{k,p_i}} a_j p_i^{-j-1}$, where $k=\sum_{j=0}^{l_{k,p_i}} a_j p_i^{j}$ (van der Corput).

Put $x^{0,k}=\left(\varphi_{p_1}\left(k\right),...,\varphi_{p_n}\left(k\right)\right)^T$, $k=1,...,m$.

In this case $d_n\left(\left\{x^{0,i}\right\}_{i=1}^{m}\right)=\mathrm{O}\left(\sqrt{n}m^{-1/n}\ln m\right)$.

Multidimensional bisection

The main drawback of described above approach is that network $\left\{x^{0,k}\right\}_{k=1}^{m}$ doesn’t take into account the structure of $f\left(x\right)$.

Assume that $n=1$ and

$$\left|f\left(y\right)-f\left(x\right)\right|\le M\left|y-x\right|.$$

Assume that we have already had $\left\{x^{0,k}\right\}_{k=1}^{m}$. Define

$$x^{0,m+1}=\arg\min_{x}\max_{k=1,\dots,m}\left\{f\left(x^{0,k}\right)-M\left|x-x^{0,k}\right|\right\}.$$

Generalizations on $n>1$ exist. But have many drawbacks …

Simulated annealing

$$f(x) \to \min_{x \in \mathbb{R}^n}.$$

The main idea: consider noisy dynamic $dx/dt = -\nabla f(x)$:

$$dx(t) = -\nabla f(x) + \sqrt{2T}\, dW(t),$$

where $W(t)$ – Winner process. When $t \to \infty$ one can prove that the distribution of vector $x(t)$ tends to distribution with density function

$$\frac{\exp\left(-f(x)/T\right)}{\int \exp\left(-f(y)/T\right) dy}.$$

When $T \to 0+$ this distribution concentrate around global minimum x_*.

Simulated annealing (Langevin dynamics)

So we can consider

$$dx(t) = -\nabla f(x) + \sqrt{2T(t)}dW(t)$$

with different policy of choosing $T(t)$, such that,

$$T(t) \xrightarrow[t\to\infty]{} 0+.$$

For example, in practice rather popular is the following strategy

$$T(t) = \frac{c}{\ln(2+t)}.$$

Structural parameter c significantly affected on the global convergence.

Non asymptotic results see in Xu–Chen–Zou–Gu, NeurIPS, 2018 (7575).

How we should find global minimum in practice? Almost convex cases

Unfortunately, all the described above approaches don't allow to solve global optimization problem better than with $N \sim \varepsilon^{-n/2}$ gradient calculations. But, if we have specific structure one can significantly reduce this complexity.

For example, for convex functions we have efficiently algorithms (see books of B. Polyak, A. Nemirovski, Yu. Nesterov).

But there exists many other situations where to find global optimum is not too difficult. For example (see also Yu. Nesterov, B. Polyak, 2006),

- $f(x)$ satisfies Polyak(–Kurdyka)–Lojasiewicz(Losiewicz) conditions;
- $f(x)$ – 1-weakly quasi-convex.
- Gauss–Newton + Taylor-like model; Drusvyatskiy–Ioffe–Lewis, 2016.

Polyak-Lojasiewicz condition

Assume that we'd like to solve system of nonlinear equations

$$g(x) = 0.$$

Let's reformulate this problem as optimization problem

$$f(x) = \frac{1}{2}\left\|g(x)\right\|_2^2 \to \min_{x\in\mathbb{R}^n}.$$

Assume that

$$\lambda_{\min}\left(\partial g(x)/\partial x \cdot \left[\partial g(x)/\partial x\right]^T\right) \geq \mu > 0.$$

Polyak-Lojasiewicz condition

This means that (gradient domination or PL-condition)

$$f\left(x\right)-f\left(x_*\right)\le\frac{1}{2\mu}\left\|\nabla f\left(x\right)\right\|_2^2.$$

So if additionally for $x,y\in\left\{x\in\mathbb{R}^n : f\left(x\right)\le f\left(x^0\right)\right\}$

$$\left\|\nabla f\left(y\right)-\nabla f\left(x\right)\right\|_2\le L\left\|y-x\right\|_2,$$

one can prove that for standard GD

$$x^{k+1}=x^k-\frac{1}{L}\nabla f\left(x^k\right),$$

$$f\left(x^N\right)-f\left(x_*\right)\le\exp\left(-\frac{\mu}{L}N\right)\left(f\left(x^0\right)-f\left(x_*\right)\right).$$

Weakly quasi-convexity

Assume that

$$f(x) - f(x_*) \le \left\langle \nabla f(x), x - x_* \right\rangle$$

and

$$\left\| \nabla f(y) - \nabla f(x) \right\|_2 \le L \left\| y - x \right\|_2 .$$

Then (A. Turin, 2017), similar triangles method (special variant of fast gradient method of Yu. Nesterov) converges with the same rate as $f(x)$ would be convex:

$$f(x^N) - f(x_*) \le \frac{4LR^2}{N^2}.$$

How we should find global minimum in practice? Rough solution

Semidefinite Relaxation (MAX CUT)

$$f(x) = \frac{1}{2}\sum_{i,j=1,1}^{n,n} A_{ij}\left(x_i - x_j\right)^2 \to \max_{x\in\{-1,1\}^n},$$

where

$$A = \left\| A_{ij} \right\|_{i,j=1,1}^{n,n} \ (A = A^T).$$

On this example we try to demonstrate general ides:

Sometimes there exists such threshold, that to find the solution better than this threshold is much more difficult than to find the solution corresponds to this threshold. So let’s restrict ourselves only by threshold solution.

Semidefinite Relaxation (MAX CUT)

Let’s introduce

$$L = \operatorname{diag}\left\{ \sum_{j=1}^{n} A_{ij} \right\}_{i=1}^{n} - A,$$

ς – random vector, uniformly distributed on a Hamming cube $\{-1,1\}^n$.

Note, that

$$f(x) = \langle x, Lx \rangle.$$

Simple observation:

$$E\langle \varsigma, L\varsigma \rangle \geq 0.5 \max_{x\in\{-1,1\}^n} \langle x, Lx \rangle.$$

Semidefinite Relaxation (MAX CUT)

Could we do better?

$$\max_{x\in\{-1,1\}^n}\left\langle x, Lx\right\rangle = \max_{x\in\{-1,1\}^n}\left\langle L, xx^T\right\rangle \le \max_{\substack{X\in S_+^n \\ X_{ii}=1,\, i=1,\dots,n}}\left\langle L, X\right\rangle \text{ // SDP problem!}$$

The book of Goemans–Williamson, 1995

Let Σ be the solution of SDP problem. Let

$$\xi\in N\left(0,\Sigma\right),\ \varsigma=\operatorname{sign}\left(\xi\right).$$

Then (the constant is unimprovable if $P\neq NP$ – Unique Games Conjecture)

$$E\left\langle \varsigma, L\varsigma\right\rangle \ge 0.878\max_{x\in\{-1,1\}^n}\left\langle x, Lx\right\rangle.$$

How we should find global minimum in practice? Narrowing the class

Compressed Sensing and L1-optimization (Donoho, Candes, Tao)

There are many areas where linear systems arise in which a sparse solution is unique. One is in plant breading. Consider a breeder who has a number of apple trees and for each tree observes the strength of some desirable feature. He wishes to determine which genes are responsible for the feature so he can cross bread to obtain a tree that better expresses the desirable feature. This gives rise to a set of equations $Ax = b$ where each row of the matrix A corresponds to a tree and each column to a position on the genome. The vector b corresponds to the strength of the desired feature in each tree. The solution x tells us the position on the genome corresponding to the genes that account for the feature. So one can hope that NP-hard problem $\|x\|_0 \to \min\limits_{Ax=b}$ can be replaced by convex problem $\|x\|_1 \to \min\limits_{Ax=b}$.

Compressed Sensing and L1-optimization

$$\|x\|_1 \to \min_{Ax=b}.$$

Due to Lagrange multipliers principle we can relax this problem as

$$\frac{1}{2}\|Ax-b\|_2^2+\lambda\|x\|_1 \to \min_x.$$

Under special $\lambda > 0$ this problem is equivalent to

$$\|x\|_1 \to \min_{Ax=b}.$$

What are the sufficient conditions for:

$$\|x\|_0 \to \min_{Ax=b} \quad \Leftrightarrow \quad \|x\|_1 \to \min_{Ax=b}?$$

Compressed Sensing and L1-optimization

Restricted Isometry Property (RIP)

$$(1-\delta_s)\|x\|_2^2 \le \|Ax\|_2^2 \le (1+\delta_s)\|x\|_2^2 \;\; \textit{for any } s\textit{-sparse } x.$$

Sufficient condition. Suppose that x_0 (solution of $\|x\|_0 \to \min\limits_{Ax=b}$) has at most s nonzero coordinates, matrix A satisfy RIP with $\delta_s \le \left(5\sqrt{s}\right)^{-1}$, then x_0 is the unique solution of the convex optimization problem $\|x\|_1 \to \min\limits_{Ax=b}$.

Example RIP matrix: for all $x \in \mathbb{R}^n$

$$P\Big((1-\varepsilon)\|x\|_2^2 \le \|Ax\|_2^2 \le (1+\varepsilon)\|x\|_2^2\Big) \ge 1-2\exp\left(-\varepsilon^2 n/6\right),$$

where i.i.d. $A_{ij} \in N\left(0,n^{-1}\right)$ ($0<\varepsilon<1$). If A is $(\varepsilon, 2s)$-RIP and $\|\tilde{x}\|_0 \le s$ satisfy $Ax=b$ then $\tilde{x}=x_0$.

How we should find global minimum in practice? Local minimums

In several statistics and engineering problems, including regression models with non-convex penalties and objectives, phase retrieval, non-convex (low-rank) reformulations of semidefinite programs and matrix completion it is possible to show that all stationary points are (near) global minima. The same things sometimes take place in Deep Learning (see Deep Learning book https://www.deeplearningbook.org/). So we come back to the problem of finding local minimums. We'll demonstrate three approaches:

- Practical acceleration by partially convex behavior of $f(x)$;
- Practical acceleration by adaptive choosing of parameters (1D-search);
- Theoretical acceleration by using the sum-type structure of $f(x)$.

Practical acceleration by partially convex behavior of $f(x)$

$$f(x) \to \min_{x \in \mathbb{R}^n}.$$

Assume that $\left\|\nabla f(y) - \nabla f(x)\right\|_2 \le L\left\|y - x\right\|_2$. The basic idea is to apply optimal (fast/accelerated/momentum) methods from convex optimization to non convex problems (in a small vicinity of x_* we may expect such a behavior). This idea in little bit different manner was firstly announced by Gelfand–Zetlin (1962) and B. Polyak (1964). Our plan is as follows:

- Polyak's heavy ball method;
- Nesterov's momentum methods;
- Conjugates gradient methods;
- BFGS and LBFGS.

Polyak's heavy ball method

The considered above dynamical system $dx/dt = -\nabla f\left(x\right)$ don't have a mechanical intuition. In 1964 B. Polyak propose the following generalization

$$\mu \frac{d^2 x}{dt^2} = -\nabla f\left(x\right) - p\frac{dx}{dt}.$$

The discrete variant of this system has a form

$$x^{k+1} = x^k - h\nabla f\left(x^k\right) + \beta\left(x^k - x^{k-1}\right).$$

For (strongly) convex functions this method under proper choice of parameters locally converges with best possible rates (see below). Globally it converges in Chesaro sense like standard GD (E. Ghadimi et al., 2014).

Polyak’s heavy ball method (non convex case)

In 1981 A. Grienwank show that if we consider

$$\mu(t)\frac{d^2x}{dt^2} = -\nabla f(x) - p(t)\frac{dx}{dt},$$

where $\mu(t) \sim f(x(t)) - c$ $(c > f(x_*))$ and $p(t) = F(\nabla f(x(t)))$ with special choice of $F(\ )$, then $x(t)$ when $t \to \infty$ converges to such local minimum x^{loc}, that $f(x^{loc}) \le c$.

In 2019 E. Diakonikolas and M. Jordan show that for discrete dynamic under special parameters selection one can guarantee

$$\min_{k=1,\dots,N} \left\| \nabla f(x^k) \right\|_2^2 \le \frac{2L\Delta f}{N}.$$

Nesterov's fast gradient (momentum) method

In 1983 in PhD thesis Yu. Nesterov (supervisor was B. Polyak) proposed the methods of type

$$x^1 = x^0 - h\nabla f\left(x^0\right),$$

$$x^{k+1} = x^k - h\nabla f\left(x^k + \beta_k\left(x^k - x^{k-1}\right)\right) + \beta_k\left(x^k - x^{k-1}\right).$$

For convex function this method converges under

$$h = \frac{1}{L},\ \beta_k = \frac{k-1}{k+2}:$$

$$f\left(x^N\right) - f\left(x_*\right) \le \frac{2LR^2}{N^2}.$$

Nesterov’s fast gradient (momentum) method

$$x^{k+1} = x^k - h\nabla f\left(x^k + \beta\left(x^k - x^{k-1}\right)\right) + \beta\left(x^k - x^{k-1}\right).$$

For μ-strongly convex function this method converges under

$$h = \frac{1}{L},\ \beta = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}:$$

$$f\left(x^N\right) - f\left(x_*\right) \le LR^2 \exp\left(-\frac{1}{2}\sqrt{\frac{\mu}{L}}N\right).$$

Unfortunately, to the best of our knowledge there are no results about this variant of fast gradient methods (FGM) for non convex problems. Fortunately, there exist many other variants of FGM that have the same theoretical properties in convex case and converge to extremum in non convex one.

Fast gradient method with 1D-search (Yu. Netserov et al., 2018)

$$x^{k+1} = \tau_{k+1} z^k + \left(1 - \tau_{k+1}\right) y^k,\ \tau_{k+1} \in \operatorname{Arg}\min_{\tau\in[0,1]} f\left(\tau z^k + \left(1-\tau\right) y^k\right),$$

$$y^{k+1} = x^{k+1} - h_{k+1} \nabla f\left(x^{k+1}\right),\ h_{k+1} \in \operatorname{Arg}\min_{h\geq 0} f\left(x^{k+1} - h\nabla f\left(x^{k+1}\right)\right),$$

$$z^{k+1} = z^k - \alpha_{k+1} \nabla f\left(x^{k+1}\right),$$

$$\alpha_{k+1} = \frac{1}{2L_{k+1}} + \sqrt{\frac{1}{4L_{k+1}^2} + \alpha_k^2},\ \alpha_0 = 0,$$

$$L_{k+1} = \frac{\left\|\nabla f\left(x^{k+1}\right)\right\|_2^2}{2\left(f\left(x^{k+1}\right) - f\left(y^{k+1}\right)\right)} \leq L,\ \ f\left(y^{k+1}\right) \leq f\left(x^{k+1}\right) - \frac{1}{2L}\left\|\nabla f\left(x^{k+1}\right)\right\|_2^2.$$

Convergence result of Nesterov’s method

For convex function $f(x)$ (there exists also strongly convex variant)

$$f\left(y^N\right) - f\left(x_*\right) \le \frac{2LR^2}{N^2}.$$

For non convex function $f(x)$ (see also Ghadimi–Lan, 2013; Catalyst, 2018)

$$\min_{k=1,\dots,N} \left\| \nabla f\left(y^k\right) \right\|_2^2 \le \frac{2L\Delta f}{N}.$$

In both cases these bounds unimprovable up to a constant factors. Note that close approaches proposed A. Nemirovski in the period 1978–1984.

In practice the behavior of this method is close to the family of conjugate gradients methods.

Conjugate gradients methods (with restarts)

$$x^{k+1} = x^k + h_k p^k,\ h_k \in \operatorname{Arg}\min_{h\in\mathbb{R}} f\left(x^k + hp^k\right)$$

$$p^{k+1} = \nabla f\left(x^{k+1}\right) - \beta_k p^k,\ p^{rn} = \nabla f\left(x^{rn}\right),\ r = 0,1,2,\ldots$$

$$\beta_k = -\frac{\left\|\nabla f\left(x^{k+1}\right)\right\|_2^2}{\left\|\nabla f\left(x^k\right)\right\|_2^2}, \text{ (Fletcher–Reeves)}$$

$$\beta_k = -\frac{\left\langle \nabla f\left(x^{k+1}\right), \nabla f\left(x^{k+1}\right) - \nabla f\left(x^k\right)\right\rangle}{\left\|\nabla f\left(x^k\right)\right\|_2^2}. \text{ (Polak–Ribiere–Polyak)}$$

To the best of our knowledge there are no bounds of global convergence for these methods in general. Local convergence corresponds to FGM.

BFGS

$$x^{k+1} = x^k - h_k H_k \nabla f\left(x^k\right),\ h_k \in \operatorname{Arg}\min_{h\in\mathbb{R}} f\left(x^k - hH_k \nabla f\left(x^k\right)\right)$$

$$H_{k+1} = H_k + \frac{H_k\gamma_k\delta_k^T + \delta_k\gamma_k^T H_k}{\left\langle H_k\gamma_k,\gamma_k\right\rangle} - \beta_k \frac{H_k\gamma_k\gamma_k^T H_k}{\left\langle H_k\gamma_k,\gamma_k\right\rangle},$$

$$\beta_k = 1 + \frac{\left\langle \gamma_k,\delta_k\right\rangle}{\left\langle H_k\gamma_k,\gamma_k\right\rangle},\ \gamma_k = \nabla f\left(x^{k+1}\right) - \nabla f\left(x^k\right),\ \delta_k = x^{k+1} - x^k,\ H_0 = I\,.$$

For local convergence of modified BFGS (Nesterov–Rodomanov, 2018)

$$\left\|x^k - x_*\right\|_2 \sim c^{k^2},\ c\in\left(0,1\right) \text{ depends on } \lambda_{\min}\left(\nabla^2 f\left(x_*\right)\right) > 0.$$

LBFGS

$$H_{k-q} = I,\ q \simeq 3-5.$$

Accelerated alternating minimizations method (S. Guminov et al., 2019)

This method looks the same as Nesterov’s FGM with 1D-search, but

$$y^{k+1} = x^{k+1} - h_{k+1}\nabla f\left(x^{k+1}\right),\ z^{k+1} = z^k - \alpha_{k+1}\nabla f\left(x^{k+1}\right),$$

are replaced by

$$y^{k+1} \in \operatorname{Arg}\min_{S_{i(k+1)}\left(x^{k+1}\right)} f\left(x\right),\ z^{k+1} = z^k - \alpha_{k+1}\nabla_{i(k+1)} f\left(x^{k+1}\right),$$

where

$$S_i\left(x^{k+1}\right) = x^{k+1} + \operatorname{span}\left\{e_j,\ j \in I_i\right\},\ i\left(k+1\right) \in \operatorname{Arg}\max_{i=1,\ldots,m}\left\|\nabla_i f\left(x^{k+1}\right)\right\|_2^2.$$

The rates of convergence remain the same up to a

$L \to mL,\ m$ is a number of blocks ($i = 1,\ldots,m$).

Accelerated alternating minimizations method

The theoretical rate of convergence is worthier than for Nesterov’s FGM with 1D-search. But in practice such type of methods can converges much faster than theoretical bounds. The reason explains in

http://blog.mrtz.org/2013/12/04/power-method.html

Note, that the described above approach allows to accelerate standard alternating minimization algorithms (typically used, when $m = 2$). In particular, for Matrix Completion problem and Matrix (Tensor) Factorization.

Note: That the complexity of iteration up to a logarithmic factor remains the same as for non accelerated variant. That is, here we also have almost pure acceleration. Without any payment for that!

Stochastic optimization

$$f(x) \to \min_{x \in \mathbb{R}^n}.$$

Assume that instead of real gradient we have an access to unbiased stochastic gradient $\nabla_x f(x,\xi)$ where

$$E_\xi\left[\nabla_x f(x,\xi)\right] \equiv \nabla f(x),\; E_\xi\left[\left\|\nabla_x f(x,\xi) - \nabla f(x)\right\|_2^2\right] \le \sigma^2.$$

The main idea: *Use instead of* $\nabla f(x)$ *in described above approaches its mini-batched version (can be calculated in a parallel manner)*

$$\overset{r}{\nabla}_x f\left(x,\left\{\xi^l\right\}_{l=1}^r\right) = \frac{1}{r}\sum_{l=1}^r \nabla_x f\left(x,\xi^l\right),$$

$$E_\xi\left[\overset{r}{\nabla}_x f\left(x,\left\{\xi^l\right\}_{l=1}^r\right)\right] \equiv \nabla f(x),\; E_\xi\left[\left\|\overset{r}{\nabla}_x f\left(x,\left\{\xi^l\right\}_{l=1}^r\right) - \nabla f(x)\right\|_2^2\right] \le \frac{\sigma^2}{r}.$$

Stochastic optimization

The main question: *How to choose batch-size and step-size?*

To answer for this question note, that

$$\left\|\nabla f\left(y\right)-\nabla f\left(x\right)\right\|_2 \le L\left\|y-x\right\|_2 .$$

$$\Downarrow$$

$$f\left(y\right)\le f\left(x\right)+\left\langle \nabla f\left(x\right),y-x\right\rangle+\frac{L}{2}\left\|y-x\right\|_2^2 .$$

$$\left(y=x^{k+1},\, x=x^k\right) \quad \Downarrow \quad \left(x^{k+1}=x^k-\frac{1}{L}\nabla f\left(x^k\right)\right)$$

$$f\left(x^{k+1}\right)\le f\left(x^k\right)-\frac{1}{2L}\left\|\nabla f\left(x^k\right)\right\|_2^2 .$$

Stochastic optimization

Note also that arbitrary $L > 0$ and all $x^k, \mathrm{v} = x^{k+1} - x^k$

$$\left\langle \nabla f\left(x^k\right) - \overset{r}{\nabla}_x f\left(x^k, \left\{\xi^l\right\}_{l=1}^r\right), \mathrm{v} \right\rangle \leq \underbrace{\frac{1}{2L}\left\| \overset{r}{\nabla}_x f\left(x^k, \left\{\xi^l\right\}_{l=1}^r\right) - \nabla f\left(x^k\right)\right\|_2^2}_{\delta^{k+1}} + \frac{L}{2}\|\mathrm{v}\|_2^2.$$

Hence

$$f\left(x^{k+1}\right) \leq f\left(x^k\right) + \left\langle \overset{r}{\nabla}_x f\left(x^k, \left\{\xi^{k,l}\right\}_{l=1}^r\right), x^{k+1} - x^k \right\rangle + \frac{2L}{2}\left\|x^{k+1} - x^k\right\|_2^2 + \delta^{k+1}.$$

Therefore for Stochastic Gradient Descent (SGD)

$$\boxed{x^{k+1} = x^k - \frac{1}{2L}\overset{r}{\nabla}_x f\left(x^k, \left\{\xi^{k,l}\right\}_{l=1}^r\right):}$$

$$f\left(x^{k+1}\right) \leq f\left(x^k\right) - \frac{1}{4L}\left\| \overset{r}{\nabla}_x f\left(x^k, \left\{\xi^{k,l}\right\}_{l=1}^r\right)\right\|_2^2 + \delta^{k+1}.$$

Stochastic optimization

That is

$$x^{k+1} = x^k - \frac{1}{2L}\overset{r}{\nabla}_x f\left(x^k,\left\{\xi^{k,l}\right\}_{l=1}^{r}\right):$$

$$f\left(x^{k+1}\right) \le f\left(x^k\right) - \frac{1}{8L}\left\|\nabla f\left(x^k\right)\right\|_2^2 + \frac{3}{2}\delta^{k+1}.$$

So if $E\left[\delta^{k+1}\right] \simeq \varepsilon^2 / (24L)$ then SGD converges like (skip expectation)

$$\min_{k=1,\ldots,N}\left\|\nabla f\left(x^k\right)\right\|_2 \le \varepsilon, \text{ where } N = \frac{16L\Delta f}{\varepsilon^2}.$$

So it is natural to choose r such that

$$\frac{\varepsilon^2}{24L} = E\left[\delta^{k+1}\right] = E\left[\frac{1}{2L}\left\|\overset{r}{\nabla}_x f\left(x^k,\left\{\xi^{l}\right\}_{l=1}^{r}\right) - \nabla f\left(x^k\right)\right\|_2^2\right] = \frac{\sigma^2}{2Lr} \Rightarrow \boxed{r = \frac{12\sigma^2}{\varepsilon^2}}.$$

Stochastic optimization

So, the bad news here is the total number of stochastic gradient calculations will be of order $\sim \varepsilon^{-4}$ (see also S. Ghadimi, G. Lan, 2015).

But the good news is that we can parallelize calculation $\sim \varepsilon^{-2}$ on processors.

Note, that in practice one typically uses other strategies to choose batch-size and step-size (see, for example, DL book). The batch-size are typically chooses much smaller and the step-size are typically chooses smaller.

In 2017 Z. Allen-Zhu proposed Natasha2, that allows to find (ε,δ) local minimum with the complexity $\tilde{O}\left(\varepsilon^{-3.25}+\delta^{-1}\varepsilon^{-3}+\delta^{-5}\right)$ of stochastic gradient calculations (under additional smoothness assumptions). In the core of Natasha2 lies Neon2 based on Oja's algorithm (stochastic version of Lanczos method), required $\sim 1/\delta^2$ matrix-vector $\nabla_x^2 f\left(x^k,\xi^k\right)p$ multiplications.

AdaGrad-Norm (R. Ward et al., 2019)

The problem of parameters selection can be partially solved by using adaptive version of SGD (this line starts in stochastic convex optimization from AdaGrad, 2011). For example (b_0 – large enough),

$$b_{k+1}^2 = b_k^2 + \left\| \overset{r}{\nabla}_x f\left(x^k, \left\{\xi^{k,l}\right\}_{l=1}^r\right) \right\|_2^2, \; x^{k+1} = x^k - \frac{\eta}{b_{k+1}} \overset{r}{\nabla}_x f\left(x^k, \left\{\xi^{k,l}\right\}_{l=1}^r\right),$$

converges as (skip expectation)

$$\min_{k=1,...,N} \left\| \nabla f\left(x^k\right) \right\|_2^2 = \tilde{\text{O}}\left(\left(\frac{\sigma}{\sqrt{r}} + \frac{\Delta f}{\eta} + L\eta\right)\left(\frac{b_0}{N} + \frac{\sigma}{\sqrt{rN}}\right)\right).$$

This result corresponds to what we’ve obtained above by proper SGD!

Most popular in practice nowadays are accelerated versions of SGD (see Deep Learning book and http://ruder.io/optimizing-gradient-descent/): Adam, AdaMax, Nadam, AMSGrad, ... Unfortunately, there are lacks of theoretical results for these methods. This direction is actively developed now in the works of: K. Levy, V. Chevher, F. Bach, G. Lan et al.

Sum-type minimization

$$f(x) = \frac{1}{m}\sum_{k=1}^{m} f_k(x) \to \min_{x\in\mathbb{R}^n},$$

$f_k(x)$ – smooth enough. Choose $\boxed{\nabla_x f(x,\xi) = \nabla f_\xi(x),\ \xi \in \text{Uniform}[1,\dots,m]}$.

We have already known, that this problem can be solved (we can find (ε,δ) local minimum, with large enough δ) by GD type methods with (in theory one can try to reduce $\varepsilon^{-2} \to \varepsilon^{-8/5}$)

$$\mathrm{O}\left(m\varepsilon^{-2}\right)$$

$\nabla f_k(x)$ calculations and by SGD type methods with (in theory one can try to reduce $\varepsilon^{-4} \to \varepsilon^{-3.25}$ (is still unbeaten even for 2nd order methods; G. Lan et al., 2018, 2019; Park–Juang–Pardalos, 2019 and references there in))

$$\mathrm{O}\left(\varepsilon^{-4}\right)$$

$\nabla f_k(x)$ calculations. Note, that in data science applications we need $m \gg \varepsilon^{-2}$.

Are there exists anything between these two bounds? Variance reduction

The answer is positive. In 2017 Z. Allen-Zhu and Y. Li proposed general technique (envelop) Neon2. Based on this envelop and well-known variance-reduction incremental algorithms they proposed Neon2+SVRG with

$$\tilde{O}\left(m^{2/3}\varepsilon^{-2}\right)$$

$\nabla f_k\left(x\right)$ calculations and Neon2+CDHS with

$$\tilde{O}\left(m\varepsilon^{-1.5}+m^{3/4}\varepsilon^{-1.75}\right).$$

$\nabla f_k\left(x\right)$ calculations. These methods require additional smoothness assumptions on $\left\{f_k\left(x\right)\right\}_{k=1}^{m}$ and are not fully parallelized like SGD.

Unfortunately, for the moment it seems that Neon2 is more interesting in theory then in practice. That is, we don't know good experiments that demonstrate efficiency of this approach in practice.

For the moment there are no tight lower bounds for $\left(\varepsilon,\delta\right)$ local minimum or ε-extremum for this class of problems.

Variance reduction (almost convex case, D. Zhou, Q. Gu, 2019)

For simplicity assume that each $f_k(x)$ has 1-Lipschitz continuous gradient. If $f(x)$ is μ-strongly convex ($f_k(x)$ can be non convex!), than one can find ε-solution with (all these bounds are optimal up to red term)

$$\mathrm{O}\left(\min\left\{\left(m+m^{3/4}\mu^{-0.5}\right)\ln\left(\varepsilon^{-1}\right), m+m^{3/4}\varepsilon^{-0.5}\right\}\right)$$

$\nabla f_k(x)$ calculations. If each $f_k(x)$ is μ-strongly convex, than with

$$\mathrm{O}\left(\min\left\{\left(m+m^{0.5}\mu^{-0.5}\right)\ln\left(\varepsilon^{-1}\right), m+m^{0.5}\varepsilon^{-0.5}\right\}\right).$$

If $\lambda_{\min}\left(\nabla^2 f(x)\right)\ge-\sigma$, than one can find ε-extremum with

$$\mathrm{O}\left(\varepsilon^{-2}\max\left\{m^{3/4}\sigma^{0.5}, m^{0.5}\right\}\right)$$

$\nabla f_k(x)$ calculations. If for each $f_k(x)$ $\lambda_{\min}\left(\nabla^2 f(x)\right)\ge-\sigma$, than with

$$\mathrm{O}\left(\varepsilon^{-2}\max\left\{m\sigma+m^{0.5}\sigma^{0.5}, m^{0.5}+1\right\}\right).$$

Variance reduction (almost convex case)

To obtain these bounds in non convex case we should use combination of optimal convex variance reduction methods with the special iterative regularization technique (Z. Allen-Zhu, 2018; Paquette–Lin–Drusvyatskiy–Marial–Harchaoui, Catalyst 2018; Asi–Duchi, 2019)

$$x^{k+1} \approx \arg\min_x \left\{ f(x) + \sigma \left\| x - x^k \right\|_2^2 \right\}.$$

Note, that auxiliary problem is $(\sigma/2)$-strongly convex.

In the last 5 years the interest for the sum-type minimization is significantly increase. Additionally to what we've talked about in literature also rather popular are parallel and distributed algorithms for sum-type target function. Unfortunately, for the moment optimal parallel and distributed algorithms for such problems develop only in the case when all $f_k(x)$ are convex, see, for example, D. Dvinskikh, A. Gasnikov, 2019 and literature therein.

Possible directions of further research

Above we mainly concentrate on general approaches for wide classes of non convex optimization problems. Buy fixing the class of problems one may expect better practical and theoretical convergence. For example, general optimal control problems are significantly non convex, but there exist many efficient practical approaches use the specificity of control problems. So an interesting problem is to find new interesting classes of non convex problems that can be solved efficiently.

In this presentation we concentrate on gradient type methods. But sometimes it’s hardly possible to obtain even the exact value of target function. Finite differentiation or batching for randomly chosen orts allows to approximate gradient and to use the described above approaches. But more delicate strategies allow to reduce the worst case smoothness constants to the average ones (Larson–Menickelly–Wild, 2019). Moreover, inexactness in function value generate additional requirement to the sensitivity analysis of this type methods, Y. Li, A. Risteski NIPS, 2016 (convex case).

To be continued…

Lecture 2. Distributed and Parallel optimization

September 17, 2019

Lecture 3. Optimization in Machine Learning (Deep Learning)

October 14 – 19, 2019