Universal accelerated envelop and some applications

Gasnikov Alexander (MIPT gasnikov.av@mipt.ru)

*Joint work with P. Dvurechensky, D. Kamzolov, C. Uribe
and from Yahoo side S. Lee, E. Ordentlich*

Main literature

Gasnikov A. Universal gradient descent. arXiv:1711.00394

Nesterov, Y. Lectures on convex optimization (Vol. 137). Berlin, Germany: Springer, 2018

Lan, G. First-order and Stochastic Optimization Methods for Machine Learning. Springer, 2020

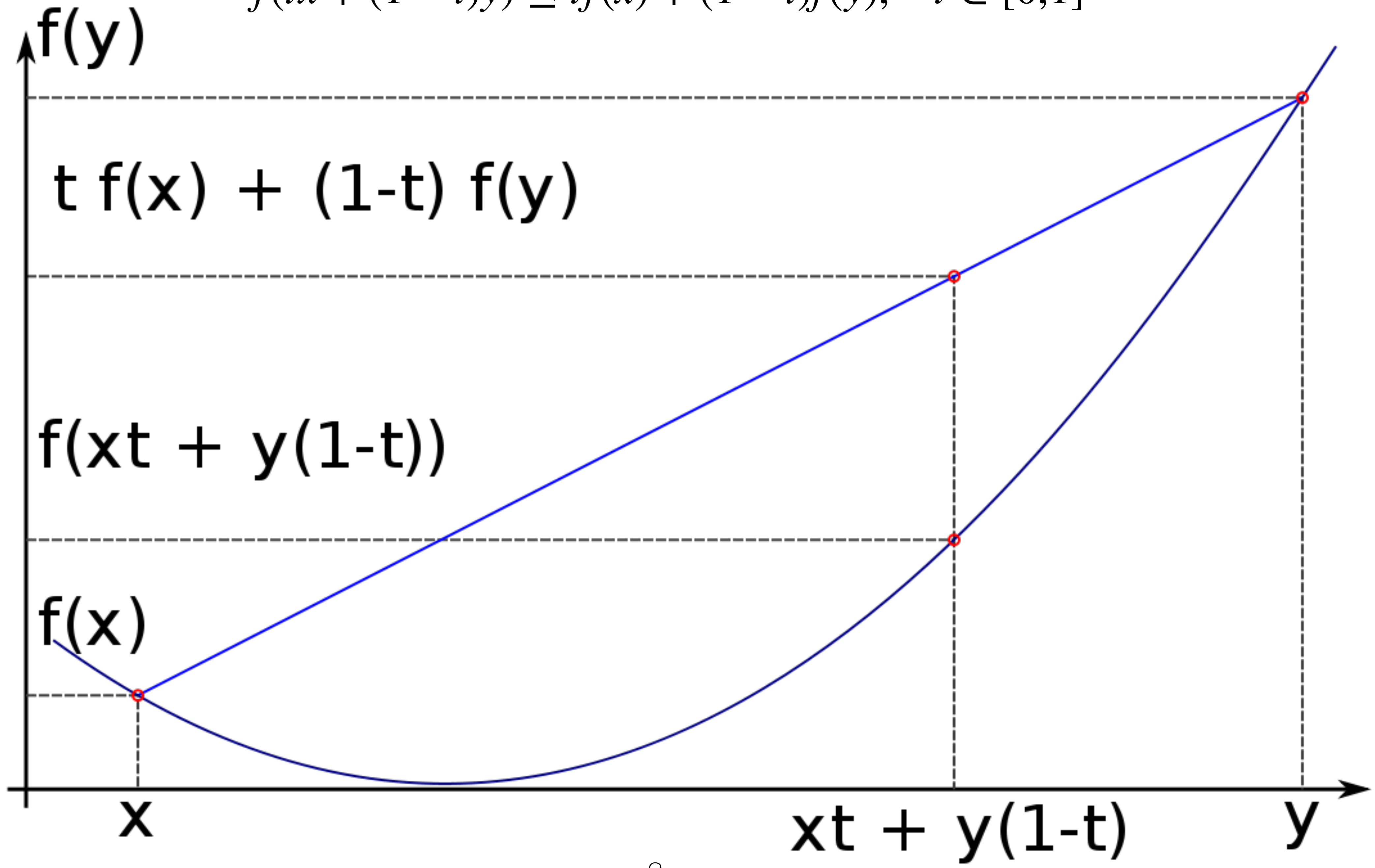
Lin, Z., Li, H., & Fang, C. Accelerated Optimization for Machine Learning. Springer, 2020

Dvinskikh, D., Kamzolov, D., Gasnikov, A., Dvurechensky, P., Pasechnyk, D., Matykhin, V., & Chernov, A. (2020). Accelerated meta-algorithm for convex optimization. arXiv:2004.08691

Smooth convex optimization

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

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y), \quad t \in [0,1]$$



Smooth convex optimization

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

Function f has L-Lipschitz continuous gradient if:

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

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 \;\; \forall x, y \in \mathbb{R}^d$$

Function f is convex if:

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle \;\; \forall x, y \in \mathbb{R}^d$$

Function f has is μ-strongly convex if:

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

Smooth convex optimization

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

How many times N we should calculate $\nabla f(x)$ to obtain x_N:

$$f(x_N) - \min_{x\in\mathbb{R}^d} f(x) = f(x_N) - f(x_*) \leq \varepsilon?$$

Let's introduce $R = \|x_0 - x_*\|$, where x_0 is a starting point of considered algorithm. For gradient descent (when μ is small enough or zero)

$$x_{k+1} = \text{argmin} \left\{ f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2}\|x - x_k\|^2 \right\} = x_k - \frac{1}{L}\nabla f(x_k)$$

$$f(x_N) - f(x_*) \leq \frac{LR^2}{2N}$$

Does this algorithm optimal? NO!

Smooth convex optimization

Does gradient descent optimal for convex optimization? NO!

Algorithm 1 Accelerated algorithm of Nesterov–Monteiro–Svaiter

```
1: Input: $f : \mathbb{R}^d \to \mathbb{R}$, $L > 0$.
2: $A_0 = 0, y_0 = x_0$.
3: for $k = 0$ to $k = K - 1$
4:     Define $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^d$ from
           $\lambda_{k+1} = \frac{1}{2L}$,
           $y_{k+1} = \operatorname{argmin}_{y \in \mathbb{R}^d} \left\{ f(\tilde{x}_k) + \langle \nabla f(\tilde{x}_k), y - \tilde{x}_k \rangle + \frac{L}{2}\|y - \tilde{x}_k\|^2 \right\} = \tilde{x}_k - \frac{1}{L}\nabla f(\tilde{x}_k)$
           $a_{k+1} = \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1}A_k}}{2}$, $A_{k+1} = A_k + a_{k+1}$,
           $\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k$.
6:     $x_{k+1} := x_k - a_{k+1}\nabla f(y_{k+1})$.
7: end for
8: return $y_K$
```

$$f(y_k) - f(x_*) \le \frac{4LR^2}{k^2}$$

The only thing we can improve in general by choosing another algorithm is to reduce $4 \to 1$.

$$f(y_N) - f(x_*) \le \frac{4LR^2}{k^2} = \varepsilon$$

$$N(\varepsilon) = \sqrt{\frac{4LR^2}{\varepsilon}} = O\left(\sqrt{\frac{LR^2}{\varepsilon}}\right)$$

Smooth convex composite optimization

$$\min_{x\in\mathbb{R}^d}\left\{F(x) := f(x) + g(x)\right\} \qquad F(y_k) - F(x_*) \le \frac{4LR^2}{k^2}$$

Algorithm 2 Composite accelerated algorithm of Nesterov–Monteiro–Svaiter

```
1: Input: $f : \mathbb{R}^d \to \mathbb{R}$, $g : \mathbb{R}^d \to \mathbb{R}$, $L > 0$.
2: $A_0 = 0, y_0 = x_0$.
3: for $k = 0$ to $k = K - 1$
4:     Define $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^d$ from
           $\lambda_{k+1} = \frac{1}{2L},$
           $y_{k+1} = \operatorname{argmin}_{y\in\mathbb{R}^d} \left\{ f(\tilde{x}_k) + \langle \nabla f(\tilde{x}_k), y - \tilde{x}_k \rangle + g(y) + \frac{L}{2}\|y - \tilde{x}_k\|^2 \right\}$
           $a_{k+1} = \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1}A_k}}{2}, \quad A_{k+1} = A_k + a_{k+1},$
           $\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k.$
6:     $x_{k+1} := x_k - a_{k+1}\nabla f(y_{k+1}) - a_{k+1}\nabla g(y_{k+1})$.
7: end for
8: return $y_K$
```

Smooth convex composite optimization

$$\min_{x\in\mathbb{R}^d} \left\{F(x) := f(x) + g(x)\right\} \qquad F(y_k) - F(x_*) \le \frac{4LR^2}{k^2}$$

We assume that auxiliary problem

$$y_{k+1} = \operatorname*{argmin}_{y\in\mathbb{R}^d} \left\{ f(\tilde{x}_k) + \langle \nabla f(\tilde{x}_k), y - \tilde{x}_k \rangle + g(y) + \frac{L}{2}\|y - \tilde{x}_k\|^2 \right\}$$

can be efficiently solved (g is proximal-friendly).

Example LASSO ($L = \lambda_{\max}(A^TA)$)

$$\min_{x\in\mathbb{R}^d} \left\{ F(x) := \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1 \right\}$$

Nesterov, Y. (2013). Gradient methods for minimizing composite functions. Mathematical Programming, 140(1), 125-161.

Accelerated gradient sliding

$$\min_{x \in \mathbb{R}^d} \left\{ F(x) := f(x) + g(x) \right\}$$

But if g is not proximal-friendly, what should we do?

Idea: To apply Accelerated Algorithm (AA) for next problem

$$y_{k+1} = \underset{y \in \mathbb{R}^d}{\operatorname{argmin}} \left\{ f(\tilde{x}_k) + \langle \nabla f(\tilde{x}_k), y - \tilde{x}_k \rangle + g(y) + \frac{L}{2} \|y - \tilde{x}_k\|^2 \right\}$$

This problem is L-strongly convex. So we should apply proper restarted version of AA (see below). In this case the complexity splits ($L_f \le L_g$):

$$O\left(\sqrt{\frac{L_f R^2}{\varepsilon}}\right) \quad \nabla f \text{ calls} \qquad \qquad \tilde{O}\left(\sqrt{\frac{L_g R^2}{\varepsilon}}\right) \quad \nabla g \text{ calls}$$

Lan, G. First-order and Stochastic Optimization Methods for Machine Learning. Springer, 2020

Data Science applications

$$\min_{x \in \mathbb{R}^d} \left\{ F(x) := f(x) + g(x) \right\}$$

$$f(x) := \frac{1}{m} \sum_{k=1}^{m} f_k(x)$$

If g is proximal-friendly, then Composite AA requires

$$O\left(m \sqrt{\frac{L_f R^2}{\varepsilon}} \right) \quad \nabla f_k \text{ calls}$$

But this is not an optimal bound. Optimal bound will be (variance reduction)

$$O\left(m + \sqrt{m \frac{\max L_k R^2}{\varepsilon}} \right) \quad \nabla f_k \text{ calls}$$

If g is not proximal-friendly (Kernel SVM) it's an open problem. See below!

Lan, G. First-order and Stochastic Optimization Methods for Machine Learning. Springer, 2020

Yahoo problem: Click-prediction model

$$\min_{x \in \mathbb{R}^d} \left\{ F(x) := f(x) + \frac{\mu}{2}\|x\|_2^2 \right\}$$

$$f(x) := \frac{1}{m}\sum_{k=1}^{m} f_k(x) \qquad f_k(x) = \log\Big(1 + \exp\left(-y_k\langle a_k, x\rangle\right)\Big)$$

Let s be a maximal number of nonzero elements in a_k and $L_k = O\left(\|a_k\|^2\right)$, $k = 1,\ldots,m$. The total complexity (arithmetic operations) of optimal first-order variance-reduced schemes will be (we consider $\mu \leq \varepsilon/\|x_*\|^2$)

$$O\left(sm + s\sqrt{m\frac{\max L_k R^2}{\varepsilon}}\right) \quad \text{a.o.}$$

Is it possible to solve this problem faster? If ε is small enough the answer is YES. For that we should use accelerated tensor methods!

Problem formulation

$$\min_{x\in\mathbb{R}^d}\{F(x) := f(x) + g(x)\}$$

where f, g - convex.

$$D^k f(x)[h]^k = \sum_{i_1,\dots,i_d\geq 0:\ \sum_{j=1}^d i_j = k} \frac{\partial^k f(x)}{\partial x_1^{i_1}\dots\partial x_d^{i_d}} h_1^{i_1}\cdot\ldots\cdot h_d^{i_d},$$

$$\|D^k f(x)\| = \max_{\|h\|\leq 1}\left\|D^k f(x)[h]^k\right\|.$$

$$\|D^p f(x) - D^p f(y)\| \leq L_{p,f}\|x-y\|.$$

$$\Omega_p(f,x;y) = f(x) + \sum_{k=1}^{p}\frac{1}{k!}D^k f(x)\left[y-x\right]^k, y\in\mathbb{R}^d.$$

$$|f(y) - \Omega_p(f,x;y)| \leq \frac{L_{p,f}}{(p+1)!}\|y-x\|^{p+1}.$$

Brief history of p-order acceleration

Nesterov, Y. E. (1983). A method for solving the convex programming problem with convergence rate O (1/k^ 2). In Dokl. akad. nauk Sssr (Vol. 269, pp. 543-547). $p = 1$

Nesterov, Y. (2008). Accelerating the cubic regularization of Newton's method on convex problems. Mathematical Programming, 112(1), 159-181. $p = 2$ (not optimal)

Monteiro, R. D., & Svaiter, B. F. (2013). An accelerated hybrid proximal extragradient method for convex optimization and its implications to second-order methods. SIAM Journal on Optimization, 23(2), 1092-1125. $p = 2$

Nesterov, Y. (2019). Implementable tensor methods in unconstrained convex optimization. Mathematical Programming, 1-27. $p \geq 2$ (not optimal); $p = 2{,}3$ are implementable!

Gasnikov, A., Dvurechensky, P., Gorbunov, E., Vorontsova, E., Selikhanovych, D., Uribe, C. A., ... & Jiang, Q. (2019, June). Near Optimal Methods for Minimizing Convex Functions with Lipschitz p-th Derivatives. In Conference on Learning Theory (pp. 1392-1393). $p \geq 2$

Reminder

Algorithm 2 Composite accelerated algorithm of Nesterov–Monteiro–Svaiter

```
1: **Input:** $f : \mathbb{R}^d \to \mathbb{R}$, $L > 0$.
2: $A_0 = 0, y_0 = x_0$.
3: **for** $k = 0$ **to** $k = K - 1$
4:     Define $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^d$ from
           $$\lambda_{k+1} = \frac{1}{2L},$$
           $$y_{k+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \left\{ f(\tilde{x}_k) + \langle \nabla f(\tilde{x}_k), y - \tilde{x}_k \rangle + g(y) + \frac{L}{2}\|y - \tilde{x}_k\|^2 \right\}.$$
           $$a_{k+1} = \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1}A_k}}{2}, \quad A_{k+1} = A_k + a_{k+1},$$
           $$\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k.$$
6:     $x_{k+1} := x_k - a_{k+1}\nabla f(y_{k+1}) - a_{k+1}\nabla g(y_{k+1})$.
7: **end for**
8: **return** $y_K$
```

Main Algorithm

Algorithm 3 Accelerated Methaalgorithm (AM) (AM(x_0,f,g,p,H,K))

```
1: Input: $p \in \mathbb{N}$, $f : \mathbb{R}^d \to \mathbb{R}$, $g : \mathbb{R}^d \to \mathbb{R}$, $H > 0$.
2: $A_0 = 0, y_0 = x_0$.
3: for $k = 0$ to $k = K - 1$
4:     Define $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^d$ from
           $$\frac{1}{2} \leq \lambda_{k+1} \frac{H \|y_{k+1} - \tilde{x}_k\|^{p-1}}{p!} \leq \frac{p}{p+1},$$
       where
           $$y_{k+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \left\{ f(\tilde{x}_k) + \sum_{k=1}^{p} \frac{1}{k!} D^k f(\tilde{x}_k) \left[y - \tilde{x}_k\right]^k + g(y) + \frac{H}{(p+1)!} \|y - \tilde{x}_k\|^{p+1} \right\}$$
           $$a_{k+1} = \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1} A_k}}{2}, \quad A_{k+1} = A_k + a_{k+1},$$
           $$\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k.$$
5:     $x_{k+1} := x_k - a_{k+1} \nabla f(y_{k+1}) - a_{k+1} \nabla g(y_{k+1})$.
6: end for
7: return $y_K$
```

Main Algorithm

Algorithm 3 Accelerated Methaalgorithm (AM) (AM(x_0,f,g,p,H,K))

```
1: Input: $p \in \mathbb{N}$, $f : \mathbb{R}^d \to \mathbb{R}$, $g : \mathbb{R}^d \to \mathbb{R}$, $H > 0$.
2: $A_0 = 0, y_0 = x_0$.
3: for $k = 0$ to $k = K - 1$
4:     Define $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^d$ from
           $$\frac{1}{2} \leq \lambda_{k+1} \frac{H\|y_{k+1} - \tilde{x}_k\|^{p-1}}{p!} \leq \frac{p}{p+1},$$
       where
           $$y_{k+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \left\{ \Omega_p(f, \tilde{x}_k; y) + g(y) + \frac{H}{(p+1)!}\|y - \tilde{x}_k\|^{p+1} \right\} \quad \text{(AP)}$$
           $$a_{k+1} = \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1}A_k}}{2}, \; A_{k+1} = A_k + a_{k+1},$$
           $$\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k.$$
5:     $x_{k+1} := x_k - a_{k+1}\nabla f(y_{k+1}) - a_{k+1}\nabla g(y_{k+1})$.
6: end for
7: return $y_K$
```

Main Theorem

Theorem. Let y_k – output of AM(x_0,f,g,p,H,k) after k iterations under $p \geq 1$ and $H \geq (p+1)L_{p,f}$. Then

$$F(y_k) - F(x_*) \leq \frac{c_p H R^{p+1}}{k^{\frac{3p+1}{2}}} \quad \text{(RC)}$$

where $c_p = 2^{p-1}(p+1)^{\frac{3p+1}{2}}/p!$, $R = \|x_0 - x^*\|$.

Moreover, if $p \geq 2$ for ε: $F(y_k) - F(x_*) \leq \varepsilon$ at each iteration of AM we have to solve auxiliary problem (AP) on (λ_{k+1}, y_{k+1}) at most $O\left(\ln\left(\varepsilon^{-1}\right)\right)$ times.

Reminder:

$$\Omega_p(f, x; y) = f(x) + \sum_{k=1}^{p} \frac{1}{k!} D^k f(x) \left[y - x\right]^k, y \in \mathbb{R}^d.$$

$$y_{k+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \left\{ \Omega_p(f, \tilde{x}_k; y) + g(y) + \frac{H}{(p+1)!} \|y - \tilde{x}_k\|^{p+1} \right\} \quad \text{(AP)}$$

Main Drawback

The main theorem assumes that we have to solve (AP) exactly!

$$y_{k+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \left\{ \widetilde{\Omega}^k(y) := \Omega_p(f, \tilde{x}_k; y) + g(y) + \frac{H}{(p+1)!}\|y - \tilde{x}_k\|^{p+1} \right\} \qquad \text{(AP)}$$

Is it possible to relax this requirement? YES!

Let's use instead of (AP) the following (practical) criteria:

$$\left\| \nabla \widetilde{\Omega}^k(\tilde{y}_{k+1}) \right\| \leq \frac{1}{4p(p+1)} \|\nabla F(\tilde{y}_{k+1})\|.$$

In this case the main theorem holds true with minor correction:

$$F(y_k) - F(x_*) \leq \frac{12}{5} \frac{c_p H R^{p+1}}{k^{\frac{3p+1}{2}}} \qquad \text{(RC)}$$

With what precision should we solve (AP)?

$$y_{k+1} = \operatorname*{argmin}_{y\in\mathbb{R}^d} \left\{ \widetilde{\Omega}^k(y) := \Omega_p(f, \tilde{x}_k; y) + g(y) + \frac{H}{(p+1)!}\|y - \tilde{x}_k\|^{p+1} \right\} \qquad \text{(AP)}$$

If F is r-uniformly convex with constant $\sigma_r > 0$

$$F(y) \geq F(x) + \langle \nabla F(x), y - x \rangle + \frac{\sigma_r}{r}\|y - x\|^r, \quad x, y \in \mathbb{R}^d.$$

Then

$$F(\tilde{y}_{k+1}) - F(x_*) \leq \frac{r-1}{r}\left(\frac{1}{\sigma_r}\right)^{\frac{1}{r-1}} \|\nabla F(\tilde{y}_{k+1})\|^{\frac{r}{r-1}}$$

Hence, if we want $F(y_k) - F(y_*) \leq \varepsilon$. Then it's sufficiently to satisfy

$$\left\|\nabla \widetilde{\Omega}^k(\tilde{y}_{k+1})\right\| = O\left(\left(\epsilon^{r-1}\sigma_r\right)^{\frac{1}{r}}\right).$$

With what precision should we solve (AP) when $p = 1$?

$$y_{k+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \left\{ \Omega_1(f, \tilde{x}_k; y) + g(y) + \frac{H}{2} \|y - \tilde{x}_k\|^2 \right\} \quad \text{(AP)}$$

Denote y^*_{k+1} - the exact solution of (AP) and

L_1^g is Lipschitz gradient constant of g.

Assume that

$$\|\tilde{y}_{k+1} - y^*_{k+1}\| \leq \frac{H}{3H + 2L_1^g} \|\tilde{x}_k - y^*_{k+1}\|$$

In this case the main theorem holds true with minor correction:

$$F(y_k) - F(x_*) \leq \frac{12}{5} \frac{4HR^2}{k^2} \quad \text{(RC)}$$

With what precision should we solve (AP) when $p = 1$?

$$y_{k+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \left\{ \Omega_1(f, \tilde{x}_k; y) + g(y) + \frac{H}{2}\|y - \tilde{x}_k\|^2 \right\} \quad \text{(AP)}$$

This condition

$$\|\tilde{y}_{k+1} - y^*_{k+1}\| \leq \frac{H}{3H + 2L_1^g}\|\tilde{x}_k - y^*_{k+1}\|$$

is very convenient from the theoretical point of view.

But its main advantage is evidence that we can solve H-strongly convex problem (AP) with desired accuracy with complexity that doesn't depend on the desired precision ε, where $F(y_k) - F(y_*) \leq \varepsilon$.

This is principal moment that differs our analysis form state of the art.

How to solve (AP) when $g \equiv 0$?

$$\min_{x \in \mathbb{R}^d} \{F(x) := f(x)\}.$$

$$y_{k+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \left\{ f(\tilde{x}_k) + \sum_{k=1}^{p} \frac{1}{k!} D^k f(\tilde{x}_k) [y - \tilde{x}_k]^k + \frac{H}{(p+1)!} \|y - \tilde{x}_k\|^{p+1} \right\} \qquad \text{(AP)}$$

Nesterov, Y., & Polyak, B. T. (2006). Cubic regularization of Newton method and its global performance. Mathematical Programming, 108(1), 177-205. $p = 2$ is implementable!
Nesterov, Y. (2019). Implementable tensor methods in unconstrained convex optimization. Mathematical Programming, 1-27. $p = 3$ is implementable!

If we have $D^2 f(\tilde{x}_k) = \nabla^2 f(\tilde{x}_k)$ (it costs $O(dT_{\nabla f(x)})$) then the complexity to solve (AP) by using automatic differentiation and gradient descent in relative smoothness assumption one can solve auxiliary problem with the complexity $\tilde{O}\left(T_{\nabla f(x)} + d^3\right)$ a.o.

Nesterov Y. Inexact basic tensor methods. – 2019/23. – CORE Preprint. $p = 2$

If we don't want to calculate $D^2 f(\tilde{x}_k) = \nabla^2 f(\tilde{x}_k)$ and want to solve (AP) with precision δ (in function), then the complexity will be $O\left(T_{\nabla f(x)} \delta^{-\frac{1}{6}}\right)$.

How to solve (AP) when $g \equiv 0$?
(Super)Hyper-fast Second-order method

$$y_{k+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \left\{ f(\tilde{x}_k) + \sum_{k=1}^{p} \frac{1}{k!} D^k f(\tilde{x}_k) \left[y - \tilde{x}_k\right]^k + \frac{H}{(p+1)!} \|y - \tilde{x}_k\|^{p+1} \right\} \quad \text{(AP)}$$

Nesterov, Y. (2019). Implementable tensor methods in unconstrained convex optimization. Mathematical Programming, 1-27.
Nesterov, Y. (2020). Superfast second-order methods for unconstrained convex optimization. CORE DP, 7, 2020.

For $p = 2{,}3$ (AP) has almost the same complexity according to the developed method $\tilde{O}\left(dT_{\nabla f(x)} + d^3\right)$ a.o. and we really need only the first and the second order oracle in both cases! So if we have 3-d order smoothness, we'd better to choose in AM $p = 3$, but to solve (AP) by using second-order information.

Kamzolov, D., & Gasnikov, A. (2020). Near-Optimal Hyperfast Second-Order Method for convex optimization and its Sliding. arXiv preprint arXiv:2002.09050.
Nesterov, Y. (2020). Inexact high-order proximal-point methods with auxiliary search procedure. CORE DP, 10, 2020.

Generalization to r-uniformly convex case $(r \leq p+1)$

Algorithm 2 Restarted AM(x_0,f,g,p,r,σ_r,H,K)

```
1: Input: $r$-uniformly convex function $F = f + g : \mathbb{R}^d \to \mathbb{R}$ with constant $\sigma_r$
   and AM($x_0$,$f$,$g$,$p$,$H$,$K$).
2: $z_0 = x_0$.
3: for $k = 0$ to $K$
4:     $R_k = R_0 \cdot 2^{-k}$,
       $$N_k = \max\left\{ \left\lceil \left( \frac{r c_p H 2^r}{\sigma_r} R_k^{p+1-r} \right)^{\frac{2}{3p+1}} \right\rceil, 1 \right\}.$$
5:     $z_{k+1} := y_{N_k}$, where $y_{N_k}$ – output of AM($z_k$,$f$,$g$,$p$,$H$,$N_k$).
6: end for
7: return $z_K$
```

Generalization to r-uniformly convex case ($r \leq p+1$)

If F is r-uniformly convex with constant $\sigma_r > 0$

$$F(y) \geq F(x) + \langle \nabla F(x), y - x \rangle + \frac{\sigma_r}{r}\|y - x\|^r, \quad x, y \in \mathbb{R}^d.$$

$$y_{k+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \left\{ \Omega_p(f, \tilde{x}_k; y) + g(y) + \frac{H}{(p+1)!}\|y - \tilde{x}_k\|^{p+1} \right\} \quad \text{(AP)}$$

Corollary. Let z_k – output of Restarted AM after k iterations. If $H \geq (p+1)L_{p,f}$, $\sigma_r > 0$, then the total number of times we required to solve (AP) for $F(z_k) - F(x_*) \leq \varepsilon$ can be estimated as follows:

$$N = \tilde{O}\left(\left(\frac{H R^{p+1-r}}{\sigma_r} \right)^{\frac{2}{3p+1}} \right)$$

Note: Unfortunately, for the moment we don't know how to use in general additional r-uniform convexity of (AP) to reduce its complexity.

The main novelty

1) We introduce general non proximal-friendly composite term g in a special manner in AM. As we will see later this term allows us to consider AM as universal accelerated envelop, that allows to accelerate different methods that applied to (AP).

2) We develop a precise theory that doesn't requires to solve (AP) with the precision determined by the desired precision for initial problem. As we will see later, this allows us to improve in $Poly(log)$ times different procedures like Catalyst, Accelerated Gradient Sliding.

Vicinities

1) Generalization to Hoelder smoothness of elder derivatives

Song, C., & Ma, Y. (2019). Towards Unified Acceleration of High-Order Algorithms under H\"{o}lder Continuity and Uniform Convexity. arXiv preprint arXiv:1906.00582.

2) Adaptive and Universal generalizations

Grapiglia, G. N., & Nesterov, Y. (2019). Tensor Methods for Minimizing Functions with H\"{o}lder Continuous Higher-Order Derivatives. arXiv preprint arXiv:1904.12559.
Difficulties with auxiliary problem

3) Convergence in norm of the gradient

Dvurechensky, P., Gasnikov, A., Ostroukhov, P., Uribe, C. A., & Ivanova, A. (2019). Near-optimal tensor methods for minimizing the gradient norm of convex function. arXiv preprint arXiv:1912.03381.

4) Generalization on variational inequalities

Bullins B. Highly smooth minimization of non-smooth problems // Conference on Learning Theory. – 2020. – P. 988-1030.
Author does not consider implementation issues of 3-d order schemes. One can try to do it by using Nesterov, Y. (2019). Implementable tensor methods in unconstrained convex optimization. Mathematical Programming, 1-27.

Applications of AM

Composite optimization: g is proximal-friendly

Beck, A., & Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on imaging sciences, 2(1), 183-202. p = 1

Nesterov, Y. (2013). Gradient methods for minimizing composite functions. Mathematical Programming, 140(1), 125-161. p = 1

Catalyst: $f \equiv 0$, $p = 1$. Accelerated proximal envelop, $H = L_1^g$

Lin, H., Mairal, J., & Harchaoui, Z. (2015). A universal catalyst for first-order optimization. In Advances in neural information processing systems (pp. 3384-3392).

Accelerated gradient sliding: g isn't proximal-friendly, we apply AM for (AP)

Lan, G., & Ouyang, Y. (2016). Accelerated gradient sliding for structured convex optimization. arXiv preprint arXiv:1609.04905. p = 1

Kamzolov, D., Gasnikov, A., & Dvurechensky, P. (2020). On the optimal combination of tensor optimization methods. arXiv preprint arXiv:2002.01004. p = 2, 3

Accelerated methods for composite saddle point problems: $p = 1$

Alkousa, M., Dvinskikh, D., Stonyakin, F., & Gasnikov, A. (2019). Accelerated methods for composite non-bilinear saddle point problem. arXiv preprint arXiv:1906.03620.

Lin, T., Jin, C., & Jordan, M. (2020). Near-optimal algorithms for minimax optimization. arXiv preprint arXiv:2002.02417.

Wang Y., Li J. Improved algorithms for convex-concave minimax optimization. arXiv preprint arXiv:2006.06359

Remember the main algorithm

Algorithm 3 Accelerated Methaalgorithm (AM) (AM(x_0,f,g,p,H,K))

```
1: Input: $p \in \mathbb{N},\ f : \mathbb{R}^d \to \mathbb{R},\ g : \mathbb{R}^d \to \mathbb{R},\ H > 0.$
2: $A_0 = 0, y_0 = x_0.$
3: for $k = 0$ to $k = K - 1$
4:     Define $\lambda_{k+1} > 0$ and $y_{k+1} \in \mathbb{R}^d$ from
           $\frac{1}{2} \le \lambda_{k+1} \frac{H\|y_{k+1} - \tilde{x}_k\|^{p-1}}{p!} \le \frac{p}{p+1}$ ,
       where
           $y_{k+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \left\{ f(\tilde{x}_k) + \sum_{k=1}^{p} \frac{1}{k!} D^k f(\tilde{x}_k)\,[y - \tilde{x}_k]^k + g(y) + \frac{H}{(p+1)!}\|y - \tilde{x}_k\|^{p+1} \right\}$   (AP)
           $a_{k+1} = \frac{\lambda_{k+1} + \sqrt{\lambda_{k+1}^2 + 4\lambda_{k+1}A_k}}{2}$ , $A_{k+1} = A_k + a_{k+1}$ ,
           $\tilde{x}_k = \frac{A_k}{A_{k+1}} y_k + \frac{a_{k+1}}{A_{k+1}} x_k.$
5:     $x_{k+1} := x_k - a_{k+1}\nabla f(y_{k+1}) - a_{k+1}\nabla g(y_{k+1}).$
6: end for
7: return $y_K$
```

Some concrete examples

Kernel SVM

$$\min_{x\in\mathbb{R}^d}\left\{F(x) := \frac{1}{m}\sum_{k=1}^{m}\max\{0, 1 - y_k\langle A_k, x\rangle\} + \frac{1}{2}\langle x, Cx\rangle\right\}$$

That can be approximated

$$\min_{x\in\mathbb{R}^d}\left\{F(x) := \frac{1}{m}\sum_{k=1}^{m} f_k(\langle A_k, x\rangle) + \frac{1}{2}\langle x, Cx\rangle\right\},$$

$$f_k(x) \simeq \varepsilon \ln\left(1 + \exp\left(\frac{1 - y_k\langle A_k, x\rangle}{\varepsilon}\right)\right)$$

Some concrete examples

New accelerated gradient sliding, $p = 1$

Idea: To use accelerated variance reduced methods for (AP)!

Let's consider the following function

$$\min_{x\in\mathbb{R}^d}\left\{F(x) := \frac{1}{m}\sum_{k=1}^{m} f_k(\langle A_k, x\rangle) + \frac{1}{2}\langle x, Cx\rangle\right\}$$

We assume that $|f_k''(y)| = O(1/\epsilon)$, matrix $A = [A_1, ..., A_m]^T$ has ms nonzero elements, $\max_{k=1,\dots,m}\|A_k\|_2^2 = O(s)$, where $1 \ll s \leq n$ and C is positive semidefinite matrix, with $\lambda_{\max}(C) \leq 1/(\varepsilon m)$.

Comment: The first term is better to optimize by accelerated variance reduced scheme, the second term - by FGM. Can we split the complexities? YES! This is the first time when we can provably splits complexities from different types of oracles!

Kernel SVM

Fast Gradient Method (Nesterov, 2018) requires

$$O\left(\sqrt{\frac{(s/\varepsilon + \lambda_{\max}(C))\, R^2}{\varepsilon}}\right)$$

iterations with the complexity of each iteration

$$O\left(ms + n^2\right).$$

For proposed in this paper approach we have

$$\tilde{O}\left(\sqrt{\frac{\lambda_{\max}(C) R^2}{\varepsilon}}\right)$$

iterations of FGM for the second term in target function with the complexity of each iteration

$$O(n^2)$$

and

$$\tilde{O}\left(\sqrt{\frac{(ms/\varepsilon)\, R^2}{\varepsilon}}\right)$$

iterations of variance reduction algorithm (Allen-Zhu, 2017) with the complexity of each

$$O(s).$$

Kernel SVM

New accelerated gradient sliding, $p = 1$

From the table one can conclude that since $\lambda_{\max}(C) \leq 1/(\varepsilon m) \ll s/\varepsilon$, then our approach has better theoretical complexity.

Algorithm	Complexity	Reference
FGM	$O\left(\frac{R}{\varepsilon}\sqrt{s}\left(ms + n^2\right)\right)$	(Nesterov, 2018)
Our approach	$\tilde{O}\left(\frac{R}{\varepsilon}\sqrt{ms} \cdot s\right) + \tilde{O}\left(\sqrt{\frac{\lambda_{\max}(C)R^2}{\varepsilon}} \cdot n^2\right)$	(Ivanova et al, 2020)

Ivanova, A., Gasnikov, A., Dvurechensky, P., Dvinskikh, D., Tyurin, A., Vorontsova, E., & Pasechnyuk, D. (2020). Oracle complexity separation in convex optimization. arXiv preprint arXiv:2002.02706.

https://github.com/DenisAltruist/Oracle-Complexity-Separation

Reminder. Click-prediction model

$$\min_{x \in \mathbb{R}^d} \left\{ F(x) := f(x) + \frac{\mu}{2}\|x\|_2^2 \right\}$$

$$f(x) := \frac{1}{m}\sum_{k=1}^{m} f_k(x) \qquad f_k(x) = \log\Big(1 + \exp\big(-y_k\langle a_k, x\rangle\big)\Big)$$

Let s be a maximal number of nonzero elements in a_k and $L_k = O\left(\|a_k\|^2\right) = O(s)$, $k = 1,\ldots,m$. The total complexity (arithmetic operations) of optimal first-order variance-reduced schemes will be (we consider $\mu \leq \varepsilon$)

$$O\left(sm + s\sqrt{m\frac{\max L_k R^2}{\varepsilon}}\right) = O\left(sm + s\sqrt{m\frac{sR^2}{\varepsilon}}\right) \quad \text{a.o.}$$

Is it possible to solve this problem faster? If ε is small enough the answer is YES. For that we should use accelerated tensor methods!

Some concrete examples

Hyper-fast Second-order method for click-prediction model

Kamzolov, D., & Gasnikov, A. (2020). Near-Optimal Hyperfast Second-Order Method for convex optimization and its Sliding. arXiv preprint arXiv:2002.09050.

Idea: For sum type problem Hessian calculation can be comparable with Hessian inversion. Try to use this by applying tensor method for sums!

$$\min_{x\in\mathbb{R}^d} \frac{1}{m}\sum_{k=1}^{m} f_k(x) + g(x)$$

$$f_k(x) = \log\left(1 + \exp(-y_k\langle a_k, x\rangle)\right)$$

Some concrete examples

Hyper-fast Second-order method for click-prediction model

$$\min_{x\in\mathbb{R}^d}\left\{F(x) := \frac{1}{m}\sum_{k=1}^{m}\log\Big(1+\exp\left(-y_k\langle a_k, x\rangle\right)\Big) + \frac{\mu}{2}\|x\|_2^2\right\}$$

Hessian calculation: $O\left(s^2 m\right)$ a.o.

Hessian inversion: $O\left(d^3\right)$ a.o.

Number of iterations: $\tilde{O}\left(\left(\frac{L_{3,F}R^4}{\varepsilon}\right)^{1/5}\right)$, $L_{3,F} = O\left(\max_{k=1,\dots,m}\|a_k\|^4\right) = O(s^2)$

Totally: $\tilde{O}\left(\left(s^2 m + d^3\right)\left(\frac{s^2R^4}{\varepsilon}\right)^{1/5}\right)$ a.o.

Some concrete examples

Accelerated variance reduced method for click-prediction model

$$O\left(sm + s\sqrt{m\frac{sR^2}{\varepsilon}}\right) = O\left(s\sqrt{m\frac{sR^2}{\varepsilon}}\right) \quad \text{a.o.}$$

Hyper-fast Second-order method for click-prediction model

$$\tilde{O}\left((s^2m + d^3)\left(\frac{s^2R^4}{\varepsilon}\right)^{1/5}\right) \quad \text{a.o.}$$

Assume $d^3 = \tilde{O}(s^2m)$, $R^2 = \tilde{O}(d) = \tilde{O}\left(s^{2/3}m^{1/3}\right)$.

Then Hyper-fast method is better if: $\frac{s^{8/3}m^{17/15}}{\varepsilon^{1/5}} = \tilde{O}\left(\frac{s^{11/6}m^{2/3}}{\varepsilon^{1/2}}\right)$.

Some concrete examples

Hyper-fast Second-order method for click-prediction model

$$\min_{x\in\mathbb{R}^d}\left\{F(x) := \frac{1}{m}\sum_{k=1}^{m}\log\left(1+\exp\left(-y_k\langle a_k, x\rangle\right)\right)+\frac{\mu}{2}\|x\|_2^2\right\}$$

So, if $s = \tilde{O}(1)$ and $\varepsilon = \tilde{O}\left(m^{-14/15}\right)$

hyper-fast scheme is the best one

Unfortunately, the requirement on accuracy ε is not very practical one.

In the subsequent lecture Dmitry Kamzolov will explain how to make the applicability of Hyper-fast Second-order scheme to sum-type problems broader by using statistical preconditioned SPAG (ICML2020 workshop).

Hendrikx, H., Xiao, L., Bubeck, S., Bach, F., & Massoulie, L. (2020). Statistically Preconditioned Accelerated Gradient Method for Distributed Optimization. arXiv preprint arXiv:2002.10726.

Some concrete examples

Accelerated methods for composite saddle point problems

$$\min_{x \in Q_x} \max_{y \in Q_y} f(x, y)$$

We assume that $f(x, y)$ is ~ 1-smooth (has ~ 1 Lipschitz gradient) and μ_x-strongly convex on x and μ_y-strongly concave on y.

Since we have μ_y-strongly concavity on y, $F(x) = \max_{y \in Q_y} f(x, y) = f(x, y(x))$ has $L_F \sim 1/\mu_y$ Lipschitz gradient (see arXive:1711.03669).

So we can reduce saddle point problem to μ_x-strongly convex and L_F-smooth problem $\min_{x \in Q_x} F(x)$ with $\nabla F(x) = \nabla_x f(x, y(x))$. The complexity of this problem is $\tilde{O}(\sqrt{L_F/\mu_x}) \sim 1/\sqrt{\mu_y \mu_x}$ calls of $\nabla_x f(x, y(x))$. But to obtain $y(x)$ with required accuracy we also required $\sim 1/\sqrt{\mu_y}$ $\nabla_y f(x, y)$ calculations.

So, totally, we required $\sim 1/\sqrt{\mu_y^2 \mu_x}$ $\nabla_y f(x, y)$ calculations

Some concrete examples

Catalyst acceleration for composite saddle point problems

$$\min_{x\in Q_x}\max_{y\in Q_y} f(x,y)$$

Idea (arXiv:2002.02417): Based on minimax von Neumann-Sion-Kakutani theorem let's rewrite saddle point problem as $\max_{y\in Q_y}\min_{x\in Q_x} f(x,y) = \max_{y\in Q_y} G(y)$ and apply Catalyst to the last problem with parameter $H \sim 1$.

The number of Catalyst iterations $\sim 1/\sqrt{\mu_y}$. To solve auxiliary problem at each iteration of Catalyst by using the the approach from the previous slide (with $\mu_y := H \sim 1$) we require $\sim 1/\sqrt{H^2\mu_x} \sim 1/\sqrt{\mu_x}$ $\nabla_y f(x,y)$ calculations and $\sim 1/\sqrt{H\mu_x} \sim 1/\sqrt{\mu_x}$ $\nabla_x f(x,y)$ calculations. So totally we required $\sim 1/\sqrt{\mu_y\mu_x}$ $\nabla_x f(x,y)$ and $\nabla_y f(x,y)$ calculations. This is a lower bound!

Some references on modern 1-order acceleration schemes

Dvinskikh D. et al. Accelerated and nonaccelerated stochastic gradient descent with inexact model // arXiv preprint arXiv:2004.04490. – 2020. Acceleration in model generality (composite etc. for smooth stochastic convex programming)

Joulani P. et al. A Simpler Approach to Accelerated Stochastic Optimization: Iterative Averaging Meets Optimism // ICML, 2020. Unified view on different acceleration schemes (Nesterov's acceleration, variance reduction acceleration) for smooth stochastic convex programming

Ene A., Nguyen H. L., Vladu A. Adaptive Gradient Methods for Constrained Convex Optimization // arXiv preprint arXiv:2007.08840. – 2020. Adaptive (a la AdaGrad) accelerated schemes for smooth convex (stochastic) programming problems