Stochastic Tensor Method for Convex Optimization

Artem Agafonov, Petr Ostroukhov, Eugene Lagutin, Daniil Selikhanovych
Advisor: Alexander Gasnikov
Co-Advisor: Dmitry Kamzolov

22 августа 2020 г.
Problem Statement

\[ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \rightarrow \min_{x \in \mathbb{E}}, \]

functions \( f_i, \nabla f_i, \nabla^2 f_i, \nabla^3 f_i \) are Lipschitz continuous for all \( i, \ p \in \{0,1,2,3\}, \ x, y \in \mathbb{E} : \)

\[ \| \nabla^p f_i(x) - \nabla^p f_i(y) \| \leq L_p \| x - y \| \]
Some Definitions

Power prox function:

\[ d_p(x) = \frac{1}{p} \|x\|^p \]

Taylor approximation of function \( f \):

\[
\Phi(s) \overset{\text{def}}{=} f(x) + \sum_{i=1}^{p} \frac{1}{i!} D^i f(x)[s]^i, \quad y \in E.
\]

Nesterov’s model:

\[
\Omega(s) = \Phi(s) + \frac{M}{(p - 1)!} d_{p+1}(s)
\]
Some Definitions

We use the following model:

\[ m_k(s) = \phi_k(s) + \frac{\sigma_k}{2} d_4(s), \]

\[ \phi_k(s) = f(x_k) + g_k^\top s + \frac{1}{2} s^\top B_k s + \frac{1}{6} T_k [s]^3, \]

where \( g_k, B_k, T_k \) are approximate derivatives \( \nabla f(x_k), \nabla^2 f(x_k), \nabla^3 f(x_k) \) through sampling.
Sampling Conditions

For a given $\varepsilon$ accuracy, one can choose the size of the sample sets $S^g$, $S^b$, $S^t$ for sufficiently small $\kappa_g$, $\kappa_b$, $\kappa_t > 0$ such that $\forall s \in \mathbb{R}^d$

$$\|g_k - \nabla f(x_k)\| \leq \kappa_g \varepsilon,$$

$$\|(B_k - \nabla^2 f(x_k))s\| \leq \kappa_b \varepsilon^{2/3} \|s\|,$$

$$\|T_k[s]^2 - \nabla^3 f(x_k)[s]^2\| \leq \kappa_t \varepsilon^{1/3} \|s\|^2.$$
Nesterov’s step

\[ s_{t+1} = \arg \min_{s \in \mathbb{E}} \Omega(s) \]
\[ x_{t+1} = x_t + s_{t+1} \]

Nesterov’s model \( \Omega(s) \) satisfies two main conditions:

- model majorizes the function \( f \):
  \[ f(x + s) \leq \Omega(s), \quad x, s \in \mathbb{E}, \]
- model is convex.

We want to obtain inexact model, that satisfies these conditions.
Inexact Model

\[
\omega_k(s) = \phi_k(s) + \frac{\sigma}{2} d_4(s) +
\]
\[
+ \kappa_g \varepsilon d_1(s) + \kappa_b \varepsilon^{2/3} d_2(s) + \kappa_t \varepsilon^{1/3} d_3(s)
\]

**Theorem 1.**
Model \(\omega_k(s)\) majorizes the function \(f\):

\[
f(x + s) \leq \omega_k(s).
\]

**Theorem 2.**
Model \(\omega_k(s)\) is convex for all \(s \in \mathbb{E}\).
Inexact Model

We can smooth $\omega_k(s)$ using the following inequality

$$
\|x\| \leq \frac{\|x\|^2}{2\alpha} + \frac{\alpha}{2}.
$$

Smooth model is

$$
\zeta(s) = \phi(s) + \left( \frac{\sigma}{2} + \frac{2\kappa_t}{3} \right) d_4(s) +
\left( \frac{\kappa_g \varepsilon^{\frac{2}{3}}}{2} + \kappa_b \varepsilon^{\frac{2}{3}} + \frac{\kappa_t \varepsilon^{\frac{2}{3}}}{2} \right) d_2(s) + \frac{\kappa_g \varepsilon^{\frac{4}{3}}}{2}
$$

Theorems 1, 2 are true for $\zeta(s)$, because $\zeta(s)$ majorizes $\omega(s)$. 
Algorithm

Non-smooth step:
\[ s_{t+1} = \arg \min_{s \in E} \omega_k(s) \]
\[ x_{t+1} = x_t + s_{t+1} \]

Smooth step:
\[ s_{t+1} = \arg \min_{s \in E} \zeta_k(s) \]
\[ x_{t+1} = x_t + s_{t+1} \]
Rate of Convergence

**Theorem 3.** [Non-smooth step]
After $t + 1$ iterations function $f$ will satisfy:

$$f(x_{t+1}) - f(x_*) \leq 2\kappa_g \varepsilon D + \kappa_b \varepsilon^{2/3} D^2 \frac{4}{t + 2} +$$

$$+ 3\kappa_t \varepsilon^{1/3} \frac{D^3}{(t + 3)^2} + (L_3 + 3\sigma) \frac{32D^4}{3(t + 4)^3}.$$

**Theorem 4.** [Smooth step]
After $t + 1$ iterations function $f$ will satisfy:

$$f(x_{t+1}) - f(x_*) \leq \frac{t}{2} \kappa_g \varepsilon^4 + \left( \frac{\sigma}{2} + \frac{2\kappa_t}{3} \right) \frac{32D^4}{3(t + 4)^3} +$$

$$\left( \frac{\kappa_g \varepsilon^3}{2} + \kappa_b \varepsilon^3 + \frac{\kappa_t \varepsilon^3}{2} \right) \frac{4D^2}{t + 2}.$$
On each step of our smooth algorithm we need to solve the following problem:

\[
\zeta_k(s) = \phi_k(s) + \left( \frac{\sigma}{2} + \frac{2\kappa_t}{3} \right) d_4(s) + \\
\left( \frac{\kappa_g \varepsilon^2}{2} + \kappa_b \varepsilon^3 + \frac{\kappa_t \varepsilon^3}{2} \right) d_2(s) + \frac{\kappa_g \varepsilon^4}{2} \rightarrow \min_{s \in \mathbb{E}}.
\]
Solution of the Auxiliary Problem

Lemma 1.
Function $\zeta_k(s)$ satisfies the strong relative convexity and relative smoothness conditions

$$\nabla^2 \rho_x(s) \preceq \nabla^2 \zeta(s) \preceq \left(\frac{\tau + 2}{\tau - 2}\right) \nabla^2 \rho_x(s).$$

with

$$\rho_x = \frac{1}{2} \left(1 - \frac{2}{\tau}\right) \langle B h, h \rangle + \frac{\sigma - L^3 \tau}{2} d_4(s) + \left(1 - \frac{2}{\tau}\right) C_2 d_2(s) + \left(1 - \frac{2}{\tau}\right) \frac{2\kappa_t}{3} d_4(s).$$

This condition allows us to solve the auxiliary problem very efficiently.
Solution of the Auxiliary Problem

We solve the auxiliary problem with the following algorithm:

\[
    h_{k+1} = \arg \min_{h \in \mathcal{E}} \left\{ \langle \nabla \zeta(h_k), h - h_k \rangle + \kappa(\tau) \beta_{\rho_x}(h_k, h) \right\},
\]

where \( \beta_{\rho_x}(u, v) \) is the Bregman divergence of function \( \rho_x(\cdot) \):

\[
    \beta_{\rho_x}(u, v) = \rho_x(v) - \rho_x(u) - \langle \nabla \rho_x(u), v - u \rangle.
\]

This method has linear rate of convergence.
Stochastic Tensor Method
Comparison to other methods

Our algorithm requires \( \frac{1}{\varepsilon^2} \) computations of gradient, \( \frac{1}{4\varepsilon^3} \) computations of hessian and none computations of the third derivative.

Rate of convergence of our method is \( O \left( \frac{1}{\varepsilon^{\frac{1}{3}}} \right) \).

Total complexity is

\[
O \left( \frac{1}{\varepsilon^{\frac{7}{3}}} \right) \quad \text{in gradients,} \quad O \left( \frac{1}{\varepsilon^{\frac{5}{3}}} \right) \quad \text{in hessians.}
\]
Stochastic Average Approximation
Comparison to other methods

One can choose \( m = \tilde{O} \left( \frac{M^2 R^2}{\varepsilon^2} \right) \) and minimize the following function:

\[
F(x) \overset{\text{def}}{=} \frac{1}{m} \sum_{k=1}^{m} f(x, \xi_k) + \frac{\mu}{2} \|x\|^2,
\]

where \( \mu \sim \frac{\varepsilon}{R^2} \).

Assume that \( F(x) \) is smooth. We can solve this problem with Hyperfast algorithm with complexity:

\[
O \left( \frac{1}{\mu^{1/5}} \times \frac{1}{\varepsilon^2} \right) = O \left( \frac{1}{\varepsilon^{11/5}} \right).
\]

However, this approach requires \( O(\frac{1}{\varepsilon^2}) \) computations of both gradient and hessian.
Consider a variance-reduced accelerated gradient method Varag.

The complexity of Varag algorithm in the case, when \( m \sim \frac{1}{\varepsilon^2} \) is

\[
O \left( m \ln m + \sqrt{m \frac{L}{\mu} \ln \frac{1}{\varepsilon}} \right) \sim \tilde{O} \left( \frac{1}{\varepsilon^2} \right).
\]
The FGM requires the batch size $r$ to be $r \sim \frac{1}{\varepsilon^{3/2}}$.

On each step we need to compute $r$ gradients. However, instead of taking constant $r$, one can take increasing batch size

$$r_k \sim \frac{k}{\varepsilon}.$$ 

At large iterations

$$r_k \sim r \sim \frac{1}{\varepsilon^{3/2}}.$$ 

Number of iterations of the FGM $\sim \frac{1}{\varepsilon^{1/2}}$. 
Comparison to other methods

<table>
<thead>
<tr>
<th></th>
<th>FGM</th>
<th>SAA</th>
<th>STM</th>
<th>OSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>parallelism $\nabla f$</td>
<td>$\varepsilon^{-\frac{3}{2}}$</td>
<td>$\varepsilon^{-2}$</td>
<td>$\varepsilon^{-2}$</td>
<td>$\varepsilon^{-2}$</td>
</tr>
<tr>
<td>parallelism $\nabla^2 f$</td>
<td>0</td>
<td>$\varepsilon^{-2}$</td>
<td>$\varepsilon^{-\frac{4}{3}}$</td>
<td>$\varepsilon^{-\frac{4}{3}}$</td>
</tr>
<tr>
<td>iterations</td>
<td>$\varepsilon^{-\frac{1}{2}}$</td>
<td>$\varepsilon^{-\frac{1}{5}}$</td>
<td>$\varepsilon^{-\frac{1}{3}}$</td>
<td>$\varepsilon^{-\frac{1}{5}}$</td>
</tr>
<tr>
<td>total complexity $\nabla f$</td>
<td>$\varepsilon^{-2}$</td>
<td>$\varepsilon^{-\frac{11}{5}}$</td>
<td>$\varepsilon^{-\frac{7}{3}}$</td>
<td>$\varepsilon^{-\frac{11}{5}}$</td>
</tr>
<tr>
<td>total complexity $\nabla^2 f$</td>
<td>0</td>
<td>$\varepsilon^{-\frac{11}{5}}$</td>
<td>$\varepsilon^{-\frac{5}{3}}$</td>
<td>$\varepsilon^{-\frac{23}{15}}$</td>
</tr>
</tbody>
</table>
## STM vs FGM+batch

### Comparison to other methods

<table>
<thead>
<tr>
<th></th>
<th>FGM + batch</th>
<th>OSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>parallelism</td>
<td>$r \nabla \sim \varepsilon^{-\frac{3}{2}}$</td>
<td>$\tilde{r} \nabla \sim \varepsilon^{-2}$</td>
</tr>
<tr>
<td></td>
<td>$\tilde{r}^2 \nabla \sim \varepsilon^{-\frac{4}{3}}$</td>
<td></td>
</tr>
<tr>
<td>iteration cost</td>
<td>$d + r \nabla T \nabla$</td>
<td>$d^3 + \tilde{r} \nabla T \nabla +$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$+ d \tilde{r} \nabla^2 T \nabla$</td>
</tr>
<tr>
<td>iterations</td>
<td>$\varepsilon^{-\frac{1}{2}}$</td>
<td>$\varepsilon^{-\frac{1}{3}}$</td>
</tr>
</tbody>
</table>

One can see, that our algorithm is faster, if $d \leq \frac{1}{\varepsilon^3}$
# OSTM vs FGM + batch

Comparison to other methods

<table>
<thead>
<tr>
<th></th>
<th>FGM + batch</th>
<th>OSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>parallelism</td>
<td>$\tilde{r}_\nabla \sim \varepsilon^{-\frac{3}{2}}$</td>
<td>$\tilde{r}_\nabla \sim \varepsilon^{-2}$</td>
</tr>
<tr>
<td>iteration cost</td>
<td>$d + r_\nabla T_\nabla$</td>
<td>$d^3 + \tilde{r}<em>\nabla T</em>\nabla +$</td>
</tr>
<tr>
<td>iterations</td>
<td>$\varepsilon^{-\frac{1}{2}}$</td>
<td>$\varepsilon^{-\frac{1}{5}}$</td>
</tr>
</tbody>
</table>

One can see, that our algorithm is faster, if $d \leq \frac{1}{7} \varepsilon^{\frac{1}{15}}$
Gradient computation time

Gradient time calculation for num parameters

Gradient time calculation for batch size