Loss function optimization in the click prediction models

Sergey Egorov

Optimization Class Project. MIPT, 26 April 2021

Introduction

The problem of optimization of the logistic loss function with quadratic regularization, which occurs in the problem of forecasting user clicks, is solved. A special feature of the problem is that these two types of features are dense and sparse, and each group has its own regularization coefficient. Due to the large dimension of the problem, modifications of stochastic gradient descent are considered.

Problem

$$F(\mathbf{w}) + \psi(\mathbf{w}) = \frac{1}{m}\sum_{k=1}^{m} f_k(\mathbf{w}, (\mathbf{x}_k, y_k)) + \psi(\mathbf{w}) \to \min_{\mathbf{w}\in\mathbf{R}^n},$$

$$f_k(\mathbf{w}, (\mathbf{x}_k, y_k)) = \ln(1+\exp(-y_k\mathbf{w}^\top\mathbf{x}_k)), \quad \psi(\mathbf{w}) = \lambda_1\sum_{i=1}^{n_1} w_i^2 + \lambda_2\sum_{i=n_1+1}^{n_1+n_2} w_i^2,$$

where $\mathbf{y} \in \{1, -1\}^m$, $\mathbf{x}_k, \mathbf{w} \in \mathbf{R}^n$, $\forall 1 \le i \le n_1 : x_{ki} \ne 0$ for almost all $k \le m$, and $\forall n_1 + 1 \le i \le n_1 + n_2 : x_{ki} = 0$ for many $k \le m$.

It is proposed to compare the performance of several stochastic gradient algorithms.

Adaptive Stochastic Accelerated Gradient

This approach consists of implementing adaptive stochastic accelerated gradient descent with the selection of the Butch size at each iteration [1]. The correct Butch size allows you to control the variance, which leads to faster convergence. Here $f(x) = F(x) + \psi(x)$

```
Require: Number of iterations $N$, $D_0$ accuracy $\varepsilon$, $A_0 = 0$,
    initial guess $L_0$, $y^0 = u^0 = x^0$.
for $k = 0, \dots, N-1$ do
    $L_{k+1} := \frac{L_k}{4}$.
    repeat
        $L_{k+1} := 2L_{k+1}$.
        $\alpha_{k+1} = (1 + \sqrt{1 + 4A_kL_{k+1}})/(2L_{k+1})$,
        $A_{k+1} = A_k + \alpha_{k+1}$.
        $r_{k+1} = \max\left\{\frac{\alpha_{k+1}D_0}{\varepsilon}, 1\right\}$
        $y^{k+1} = (\alpha_{k+1}u^k + A_kx^k)/A_{k+1}$.
        $u^{k+1} = u^k - \alpha_{k+1}\nabla^{r_{k+1}} f(y^{k+1}, \{\xi_l^{k+1}\}_{l=1}^{r_{k+1}})$.
        $x^{k+1} = (\alpha_{k+1}u^{k+1} + A_kx^k)/A_{k+1}$.
    until
        $f(x^{k+1}) \le f(y^{k+1})$
        $+ \langle\nabla^{r_{k+1}} f(y^{k+1}, \{\xi_l^{k+1}\}_{l=1}^{r_{k+1}}), x^{k+1} - y^{k+1}\rangle$
        $+ L_{k+1}\|x^{k+1} - y^{k+1}\|_2^2 + \frac{\alpha_{k+1}}{2A_{k+1}}\varepsilon$.
end for
```

Statistically Preconditioned Accelerated Gradient Method

The problem will also be solved using SPAG [2]. Each iteration of the algorithm is a step of a non-stochastic gradient in the metric natural for the function $F(x)$. In fact, this is a kind of mirror descent, where the approximation of the natural metric is used. The entire stochasticity of the method consists in approximation $f(x) = \frac{1}{s}\sum\limits_{k=1}^{s} f_k(\mathbf{w}, (\mathbf{x}_k, y_k))$ of the original function $F(x)$.

```
Require: Number of iteration $N$, $\mu$ and $L_{F/\phi}$ as $\nabla^2F(x) \prec L_{F/\phi}(\nabla^2 f(x) + \mu)$
$v_0 = x_0, \quad A_0 = 0, \quad B_0 = 1, \quad G_{-1} = 1$
for $t = 0, 1, 2, \dots N$ do
    $G_t = \max\{1, G_{t-1}/2\}/2$
    repeat
        $G_t \leftarrow 2G_t$
        2. Find $a_{t+1}$ such that $a_{t+1}^2 L_{F/\phi}G_t = A_{t+1}B_{t+1}$
           where $A_{t+1} = A_t + a_{t+1}, \quad B_{t+1} = B_t + a_{t+1}\sigma_{F/\phi}$
        3. $\alpha_t = \frac{a_{t+1}}{A_{t+1}}, \quad \beta_t = \frac{a_{t+1}}{B_{t+1}}\sigma_{F/\phi}, \quad \eta_t = \frac{a_{t+1}}{B_{t+1}}$
        4. $y_t = \frac{1}{1-\alpha_t\beta_t}((1-\alpha_t)x_t + \alpha_t(1-\beta_t)v_t)$
        5. Compute $\nabla F(y_t)$
        6. $v_{t+1} =_x \left[\eta_t\left(\nabla F(y_t)^\top x + \psi(x)\right) + (1-\beta_t)D_\phi(x, v_t) + \beta_tD_\phi(x, y_t)\right]$
           where $D_\phi(x, y) = \phi(x) - \phi(y) - \nabla\phi(y)^\top(x-y), \quad \phi(x) = f(x) + \frac{\mu}{2}||x||_2^2$,
           $f(x)$ is an approximation of $F(x)$
        7. $x_{t+1} = (1-\alpha_t)x_t + \alpha_tv_{t+1}$
    until $D_\phi(x_{t+1}, y_t) \le \alpha_t^2G_t((1-\beta_t)D_\phi(v_{t+1}, v_t) + \beta_tD_\phi(v_{t+1}, y_t))$
end for
```

In this case, a gradient descent with adaptive butch selection is used to solve the subproblem.

Fast proximal gradient method (FISTA)

As the basic method for solving the problem, we will use FISTA [3]. Its main step is as follows:

$y_{k+1} = x_k + \frac{k}{k+3}(x_k - x_{k+1})$
$x_{k+1} = prox_{\alpha_k\psi}(y_{k+1} - \alpha_k\nabla f(y_{k+1}))$,
where the step is selected adaptively according to the condition:
$f(x_{k+1}) \le f(y_{k+1}) + \nabla f(y_{k+1})^\top(x_{k+1} - y_{k+1}) + \frac{1}{2\alpha_k}||x_{k+1} - y_{k+1}||_2^2$

Convergence

Here are the main estimates of the convergence rate. Number of Oracle calls:

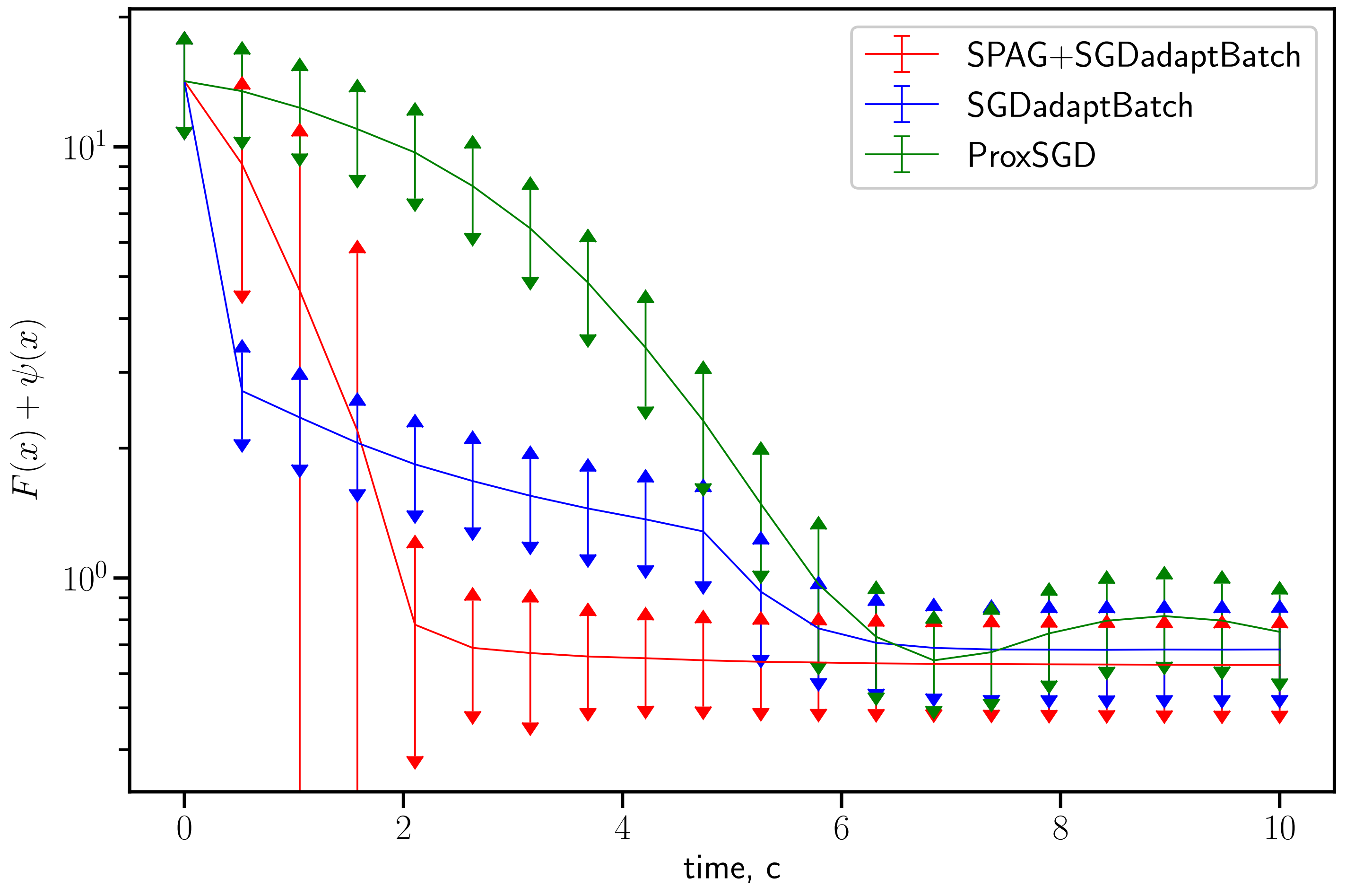
- FISTA: $T = batch\ size \cdot \widetilde{O}\left(\frac{\epsilon_0^2}{\epsilon^2} + \frac{\sigma^4}{\epsilon^2}\right)$,
- ASGD: $T = m \cdot \widetilde{O}\left(\frac{\sigma^2R^2}{\epsilon^2}\right)$,
- SPAG: the number of solutions to the subproblem is $1 + \widetilde{O}\left(\frac{R^4}{\sqrt{s\mu}}\right)$, so in this case $T = \widetilde{O}\left(\frac{R^4mn}{\sqrt{s\mu}} + \frac{\sqrt{s}\sigma^2R^6}{\epsilon^2\mu}\right)$

Numerical example

Consider a numerical example with $m = 327,062$, $n = 37,894$ and X with $26,951,907$ non-zero elements. The entries X and y are taken from the matrix collection. Here $\lambda_1 = 10^{-3}$ and $\lambda_2 = 10^{-5}$.

Results

On this numerical example, the SPAG method converges quickly.



	mean time	std time	$\min f(x)$
FISTA	6.72	0.69	0.700
ASGD	7.81	1.01	0.753
SPAG	4.75(3.84)	2.69(2.09)	0.695

Conclusion

The SPAG method showed a high rate of convergence, but its advantage became noticeable only at large matrix dimensions. In addition, the appearance of a subproblem leads to a doubling of the number of hyperparameters of the method and difficulties in configuring them. Moreover, new hyperparameters appear, such as the number of iterations of a submetod or its maximum discrepancy.

References

[1] Aleksandr Ogaltsov Darina Dvinskikh Pavel Dvurechensky Alexander Gasnikov Vladimir G. Spokoiny. Adaptive gradient descent for convex and non-convex stochastic optimization. 2020.

[2] Hadrien Hendrikx Lin Xiao S ́ebastien Bubeck Francis Bach Laurent Massouli ́e. Statistically preconditioned accelerated gradient method for distributed optimization. 2020.

[3] A. Salim and W. Hachem. On the performance of the stochastic fista. 2019.