# Distributed Preconditioned Accelerated Gradient Method

Nikolay Popov

*Optimization Class Project. MIPT*

## Introduction

The setting of distributed empirical risk minimization where multiple machines compute the gradients in parallel and a centralized server updates the model parameters is considered in this project. In order to reduce the number of communications required to reach a given accuracy, a preconditioned accelerated gradient method where the preconditioning is done by solving a local optimization problem over a subsampled dataset at the server is proposed. The convergence rate of the method depends on the square root of the relative condition number between the global and local loss functions. Experiments on real-world datasets illustrate the benefits of acceleration.

## Problem statement

Initially, SPAG was chosen as a method from [1] for solving Yahoo click prediction problem. Consider a supervised machine learning setting with the goal to predict a target variable $y \in \{-1, 1\}$ from a feature vector $a \in R^d$ using a model represented by real valued parameters $x \in R^d$. So we should minimize average loss on dataset:

$F(x) = \frac{1}{N} \sum_{i=1}^{N} \{ \; \log[1 + \exp(-y_i \langle x, a_i \rangle)] + 0.5 \cdot \lambda \cdot \|x\|_2^2 \; \} \to \underset{x \in R^d}{min} \quad (1),$

where x stands for parameter vector, $y_i -$ label and $a_i -$ feature vector. Calculation of the gradient of $F(x)$ is distributed among m macnihes in a cluster. One machine is called server and it hosts an uniform approximation $f(x)$ of $F(x) : (n << N) : \quad f(x) = \frac{1}{n} \sum_{i=1}^{n} \{\log[1+\exp(-y_i \langle x, a_i \rangle)] + 0.5 \cdot \lambda \cdot \|x\|_2^2\}.$

Here $\phi(x) = f(x) + 0.5\mu\|x\|_2^2; \quad D_\phi(x, y) = \phi(x) - \phi(y) - \nabla\phi(y)^T(x - y)$.

The proximal gradient method with the Bregman divergence of $\phi$ is used for minimization of $F(x) : x_{t+1} = \text{argmin}_{\, x \in R^d} \{\nabla F(x_t)^T x + \frac{1}{\eta_t} D_\phi(x, x_t) \} \quad (2).$

## Algorithm SPAG($L = L_{F/\phi}, \sigma = \sigma_{F/\phi}, x_0$)

```
$v_0 = x_0; A_0 = 0; B_0 = 1; G_{-1} = 1;$
$for\ t = 0, 1, 2, ...do$
.    $G_t = max 1, 0.5 G_{t-1}/2$
.    $repeat$
.        $G_t = 2G_t$
.        $Find\ a_{t+1} : a_{t+1}^2 L G_t = A_{t+1} B_{t+1} : A_{t+1} = A_t + a_{t+1};$
.        $B_{t+1} = B_t + a_{t+1}\sigma$
.        $\alpha_t = \frac{a_{t+1}}{A_{t+1}}; \beta = \frac{b_{t+1}}{B_{t+1}}\sigma; \eta_t = \frac{a_{t+1}}{B_{t+1}};$
.        $y_t = \frac{1}{1 - \alpha_t \beta_t}((1 - \alpha_t)x_t + \alpha_t(1 - \beta_t)v_t)$
.        $Compute\ \nabla F(y_t)$
.        $v_{t+1} = \text{argmin}_x V_t(x)$
.    $until\ Inequality\ (1)\ is\ satisfied$
$end\ for$
```

Here $L_{F/\phi}$ and $\sigma_{F/\phi} : \quad \sigma_{F/\phi} \nabla^2\phi(x) \leq \nabla^2 F(x) \leq L_{F/\phi} \nabla^2 \phi(x);$

$V_t(x) = \eta_t \nabla F(y_T)^T x + (1 - \beta_t) D_\phi(x, v_t) + \beta_t D_\phi(x, y_t);$

$(1) \to D_\phi(x_{t+1}, y_t) \leq \alpha_t^2 G_t[(1 - \beta_t) D_\phi(v_{t+1}, v_t) + \beta_t D_\phi(v_{t+1}, y_t)]$

## SPAG initialization and convergence

It is assumed that $\sigma_F \approx \lambda$ and that with high probability

$$\forall x \; \hookrightarrow \|\nabla^2 f(x) - \nabla^2 F(x)\|_2 \leq \mu,$$

which implies Lemma 3 from [2]:

$$\frac{\sigma_F}{\sigma_F + 2\mu} \nabla^2 \phi(x) \leq \nabla^2 F(x) \leq \nabla^2 \phi(x)$$

SPAG is initialized with $L_{F/\phi} = 1$, $\sigma_{F/\phi} = \frac{\sigma_F}{\sigma_F + 2\mu}$ and $x_0 = \text{argmin}_x f(x)$ $(3)$.

According to [3] and [4] with $\eta_t = \frac{1}{L_{F/\phi}}$ in $(2)$ the sequence $\{x_t\}$ satisfies

$$F(x) - F(x_*) \leq \left(1 - \frac{1}{\kappa_{F/\phi}}\right)^t L_{F/\phi} D_\phi(x_*, x_0); \qquad \kappa_{F/\phi} = \frac{L_{F/\phi}}{\sigma_{F/\phi}}.$$

## Local problems

Both local (i.e. calculated using only a little subsampled part of dataset) minimization problems in $(3)$ and in the body of repeat cycle in SPAG: $v_{t+1} = \text{argmin}_x V_t(x)$ are solved using implementation of L-BFGS optimization method from scipy python module with halt criterions respectively maxiter $= 5$ and pgtol $< 10^{-5}$, because it is crucial for SPAG to have an accurate solution of subproblems at each iteration.

## Dataset used

RCV dataset from website LibSVM was used in this project. The number of objects in it is 20242 and each of them is represented by a vector of 47236 features. This dataset is similar to the dataset in Yahoo problem, because its features are sparse enough as in Yahoo problem (e.g. one hot encoded features of ads and users).

## Tuning parameters

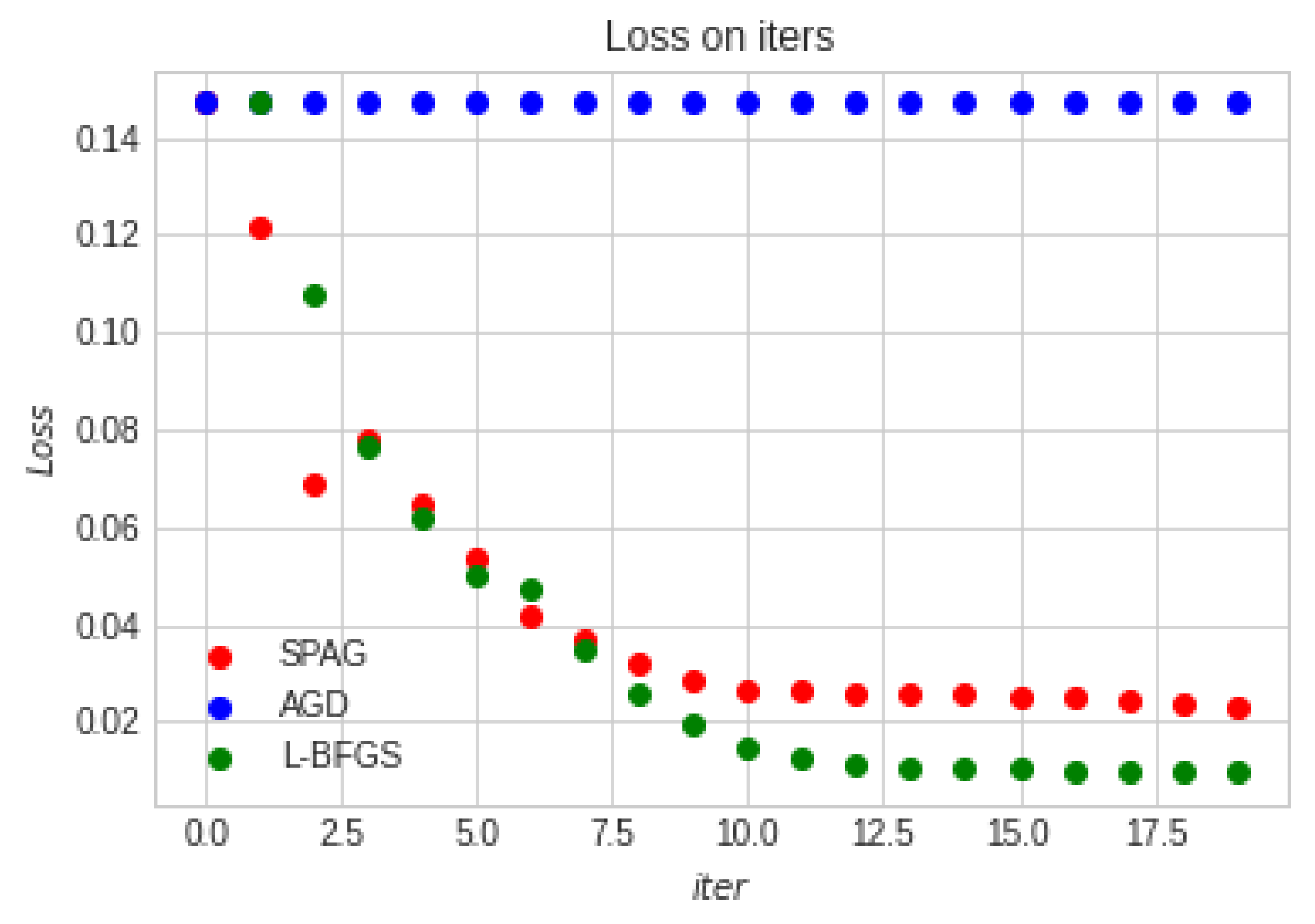
Although there are some analitical bounds on $\mu$ in [1], all hyper-parameters, such as $\lambda, \mu$ and number of objects sampled for server $n$, are found by the folowing grid search: $\lambda \in \{1e-4, 1e-5, 1e-6, 1e-7, 1e-8\}$, $\mu \in \{1e-4, 1e-5, 1e-6\}$, $n \in total\ number\ of\ objects\ divided\ by\ \{35, 30, 25, 20, 15, 10\}$. The best parameters for RCV dataset are $\lambda = 1e-7, \mu = 1e-5$ and $n = 2024$. This values are used for comparison of SPAG with other algorithms on RCV.

## The code

One can find my implementation of SPAG algorithm (Python 3.6) as well as all stuff needed to plot graphs and work with data in this Colab Notebook.

## Results

On this dataset, 3 methods are used. SPAG is implemented with tuned parameters. Nesterov accelerated gradient method (AGD) and L-BFGS are implemented with their default parameters.



Comparing the time of non-parallelized work of SPAG and L-BFGS on the same data with pgtol halt criterion did not reveal any useful information (time periods are aproximately in relation from 5:1 to 7:1) and that is because of usage L-BFGS as solver for local subproblems in SPAG.

## Conclusion

I have introduced SPAG, an accelerated algorithm that performs statistical preconditioning for large-scale optimization. Although the motivation is for distributed empirical risk minimization, SPAG applies to much more general settings. Acceleration allows SPAG to efficiently leverage rough preconditioning with limited number of local samples. Characterizing the effects of inaccurate inner solutions in the preconditioning setting would be an interesting extension of this work.

## References

[1] Bach Francis and Bubeck Sebastien. Statistically preconditioned accelerated gradient method for distributed optimization. *ArXiv*, page 34, 2020.

[2] Zhang Yuchen and Xiao Lin. Communication-efficient distributed optimization of self-concordant empirical loss in large-scale and distributed optimization. *ArXiv*, 2018.

[3] Shamir Ohad, Srebro Nati, and Tong Zhang. Communication-efficient distributed optimization using an approximate newton-type method. page 10001008, 2014.

[4] Lu Haihao, Freund Robert, and Nesterov Yurii. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, page 333354, 2018.