Step Size Learning Project

Alina Nurysheva [1] **Grégoire Ouerdane** [1] **Hussain Bukhari** [1]

Abstract

Online methods can accelerate gradient-based methods asymptotically. Here we show how hypergradient descent improves on the convergence of gradient descent. We propose a simple adaptive first-order gradient-based method. In particular, one realization of our framework achieves superlinear convergence on strongly convex quadratics using first-order information

Github repo: https://github.com/alina2002200/Step-Size-Learning.git
Presentation file: https://github.com/alina2002200/Step-Size-Learning.git

1. Introduction

1.1. GD Method and Minimization Problem

As described in (Orabona, 2023), the online learning framework is fundamental to many machine learning applications. Consider the problem of minimizing a smooth convex function $f : \mathbb{R}^d \to \mathbb{R}$. We are interested in the following optimization problem:

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

Assume the function f is L–smooth and convex:

A function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is L–smooth if f is differentiable on $\mathbb{R}^d$, and

$$\nabla f(x) - \nabla f(y) \leq Lx - y \tag{1}$$

for all $x, y \in \mathbb{R}^d$.

A function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is *convex* and attains the minimum at a (non-unique) $x_* \in \mathbb{R}^d$. We define $R\|x_0 - x_*\|$, where x^0 is a starting point of numerical methods.

[1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Alina Nurysheva <Alina.Nurysheva@skoltech.ru>.

Final Projects of the Machine Learning 2023 Course, Skoltech, Moscow, Russian Federation, 2023.

It is well-known (Bubeck et al., 2015; Nesterov, 2018) that the gradient descent (GD) method converges at a rate of $\left(\frac{LR}{T}\right)$, where T is the number of iterations. The GD method is given by the following update rule:

$$x_{k+1} = x_k - \gamma \nabla f(x_k), \tag{2}$$

where $\gamma > 0$ is the step size taken as $\gamma = \frac{1}{L}$, and $x_0 \in \mathbb{R}^d$ is a starting point.

1.2. Strongly convex case

Consider the following assumption:

A function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is μ–strongly convex if for all $x, y \in \mathbb{R}^d$,

$$f(y) \geq f(x) + \nabla f(x)y - x + \frac{\mu}{2}y - x^2. \tag{3}$$

1.3. Accelerated GD Method

At the same time, it is possible to improve these results to $\left(\sqrt{\frac{LR}{T}}\right)$ and $\left(LR^2 \exp(-\sqrt{\frac{\mu}{L}}T)\right)$. using the Nesterov's accelerated gradient method (Nesterov, 2018). We have not discussed it yet, so you should read (Nesterov, 2018) or google it. Ask me for good resources also.

1.4. Preconditioning Technique

The previous results are classical and well-known. However, it is possible to improve them further by using the preconditioning technique. Instead of (2), we consider the following update rule:

$$x_{k+1} = x_k - P_k \nabla f(x_k), \tag{4}$$

where $P_k \in \mathbb{R}^{d \times d}$ **is a matrix.** When $P_k = \gamma I$, we recover (2). It turns out that the choice of P_k can significantly improve the convergence rate of the classical version of GD (2). See a detailed discussion in (Gao et al., 2024).

Under the additional assumption that the function is μ–strongly convex, it is possible to consider (2) and show that it has the rate $\left(LR^2 \exp(-\frac{\mu}{L}T)\right)$.

1.5. The "Step Size Learning" Project

A recent work (Gao et al., 2024) proposed a new remarkably simple and efficient idea how to learn P_k during an

optimization process:

$$\begin{aligned} x_{k+1} &= x_k - P_k \nabla f(x_k), \\ P_{k+1} &= P_k - \eta \nabla g_{x_k}(P_k), \end{aligned} \tag{5}$$

where

$$g_{x_k}(P) = \frac{f(x_k - P\nabla f(x_k)) - f(x_*)}{f(x_k) - f(x_*)} \text{ or}$$

$$g_{x_k}(P) = \frac{f(x_k - P\nabla f(x_k)) - f(x_*)}{\nabla f(x_k)^2}.$$

- We reproduced a framework that accelerates gradient-based algorithms through online learning.
- One of the realizations achieved super-linear convergence on strongly convex quadratics.
- Implemented Accelerated GD to improve the method and checked, whether it works or not in quadratic case

2. Algorithms and Models

Here we briefly describe Algorithms represented in (Gao et al., 2024).

2.1. OSGM-R

The Online Scaled Gradient Method (OSGM) with ratio surrogate loss is designed to minimize the objective function through iterative updates. Below, we describe the algorithm and its key components.

We define the optimality measure $\phi(x) := f(x) - f(x^\star)$, the surrogate loss $\ell_x(P) := r_x(P)$, and use online gradient descent for the update rule.

Algorithm 1 OSGM-R

Input: Initial point x_1, scaling matrix $P_1 \in P$, stepsize $\eta > 0$
for $k = 1, 2, \ldots$ **do**
  $x_{k+1} = x_k - P_k \nabla f(x_k)$
  $P_{k+1} = \Pi_P \left[P_k - \eta \nabla r_{x_k}(P_k)\right]$
end for
Output: x_{best} with minimum objective value

The algorithm iteratively updates the solution x_k and scaling matrix P_k using online gradient descent and the ratio surrogate loss function $r_{x_k}(P_k)$. The projection operator Π_P ensures the scaling matrix stays within the feasible set P.

2.2. OSGM-G

The Online Scaled Gradient Method (OSGM) with gradient norm surrogate loss is an extension of the OSGM approach, utilizing the gradient norm surrogate to optimize the objective function.

We define the optimality measure $\phi(x) := \|\nabla f(x)\|$, the surrogate loss $\ell_x(P) := g_x(P)$, and use an online subgradient method for the update rule.

Algorithm 2 OSGM-G

Input: Initial point x_1, scaling matrix $P_1 \in P$, stepsize $\eta > 0$, nonempty oracle $M_{\|\nabla f(x)\|, P}$
for $k = 1, 2, \ldots$ **do**
  $x_{k+1} = M_{\|\nabla f(x_k)\|, P_k}(x_k)$
  $P_{k+1} = \Pi_P \left[P_k - \eta g'_{x_k}(P_k)\right]$
end for
Output: x_{best} with minimum objective value

The algorithm iteratively updates the solution x_k using the online subgradient method and adjusts the scaling matrix P_k by applying the gradient norm surrogate loss function $g_{x_k}(P_k)$. The projection operator Π_P ensures that the scaling matrix remains within the feasible set P. Additionally, the monotone oracle $M_{\|\nabla f(x)\|, P}$ is necessary for selecting the appropriate update for the solution.

2.3. OSGM-H Algorithm

The Online Scaled Gradient Method (OSGM) with hypergradient surrogate loss is another variant of OSGM that utilizes the hypergradient surrogate loss to optimize the objective function.

We define the optimality measure $\phi(x) := f(x) - f(x^\star)$ or $\|\nabla f(x)\|$, the surrogate loss $\ell_x(P) := h_x(P)$, and use online gradient descent for the update rule.

Algorithm 3 OSGM-H

Input: Initial point x_1, scaling matrix $P_1 \in P$, stepsize $\eta > 0$, nonempty oracle $M_{f(x)-f(x^\star), P}$
for $k = 1, 2, \ldots$ **do**
  Update $x_{k+1} = M_{f(x)-f(x^\star), P_k}(x_k)$
  Update the scaling matrix $P_{k+1} = \Pi_P \left[P_k - \eta \nabla h_{x_k}(P_k)\right]$
end for
Output: x_{best} with minimum objective value

The algorithm iteratively updates the solution x_k using the online gradient descent method, and updates the scaling matrix P_k using the hypergradient surrogate loss function $h_{x_k}(P_k)$. The projection operator Π_P ensures that the scaling matrix remains within the feasible set P. Additionally, the monotone oracle $M_{f(x)-f(x^\star), P}$ is required to guide the

solution updates.

3. Experiments

3.1. Experiment Setup

We perform experiments to evaluate the performance of online scaled gradient methods on strongly convex optimization problems, such as least squares and regularized logistic regression.

3.2. Synthetic Data

For the least squares problem, $f(x) = \frac{1}{2}\|Ax - b\|^2$, $A \in \mathbb{R}^{n\times n} = CDC^\top + \sigma I$, where C is element-wise generated from $0.01 \times N(0,1)$, D is a diagonal matrix with $U[0,1]^n$ diagonals, and $b \in \mathbb{R}^m$ is generated from $U[0,1]^n$.

3.3. Real Data

We use LIBSVM datasets(Chang & Lin, 2011) for the support vector machine (SVM) problem, $f(x) = \frac{1}{m}\sum_{i=1}^m f_i(x) + \frac{\lambda}{2}\|x\|^2$, where f_i is the squared hinge loss. We set $\lambda = \frac{5}{n}$.

We compare the following eight algorithms:

- (GD) Gradient descent with $\frac{1}{L}$-stepsize.
- (OptDiagGD) Gradient descent with the optimal diagonal preconditioner.
- (OSGM-R) Online scaled gradient method with ratio surrogate.
- (OSGM-G) Online scaled gradient method with gradient norm surrogate.
- (OSGM-H) Online scaled gradient method with hypergradient surrogate.
- (AdaGrad) Adaptive (sub)gradient method.
- (AGD) Accelerated gradient descent for general convex problems.
- (SAGD) Accelerated gradient descent for strongly convex problems.

OptDiagGD and OSGM-G are only tested on problems with fixed Hessian.

3.4. Algorithm Configurations

1. **Dataset Generation**: For synthetic data, $n = 100$ and $\sigma \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$.
2. **Initial Point**: Initial points x_1 are generated from $N(0, I_n)$ and scaled to unit length. Initial scaling matrix $P_0 = 0$.
3. **Maximum Iteration**: Maximum iterations set to $K = 10000$.
4. **Stopping Criterion**: Stop when $\|\nabla f(x_k)\| \leq 10^{-10}$.
5. **Stepsize Configuration**: (AdaGrad) uses the optimal stepsize from $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$.
6. **Monotone Oracle**: All OSGM methods use simple comparison as the monotone oracle.
7. **Online Learning Algorithm**: All OSGM methods use AdaGrad with the optimal stepsize.
8. **Scaling Matrix** P: $P = \mathbb{R}^{n\times n}$ for Section 8.3, and diagonal matrices for the other experiments.
9. **Knowledge of Optimal Value**: When the exact optimal value is unknown, OSGM-R uses the auxiliary surrogate $r_z(x)$. If $z > f(x_k)$, we heuristically adjust $z \leftarrow f(x_k) - \min\{5(z - f(x_k)), 1\}$ to update the lower bound. This strategy is not theoretically justified but works well in practice.

4. Results

...

References

Bubeck, S. et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Chang, C.-C. and Lin, C.-J. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):1–27, 2011. doi: 10.1145/1961189.1961199.

Gao, W., Chu, Y.-C., Ye, Y., and Udell, M. Gradient methods with online scaling. *arXiv preprint arXiv:2411.01803*, 2024.

Nesterov, Y. *Lectures on convex optimization*, volume 137. Springer, 2018.

Orabona, F. *A Modern Introduction to Online Learning*. Boston University, 2023. Available at: https://www.bu.edu/.

A. Team member's contributions

Alina Nurysheva (34% of work)

- Reviewing literate on the topic (3 papers)
- Coding the OptDiag, OSGM-R, SAGD methods in quadratic case
- Preparing the GitHub Repo
- Preparing the Section 1 of this report

Grégoire Ouerdane (33% of work)

- Reviewing literate on the topic (3 papers)
- Coding the OSGM-H, reproducing the results with real data
- Preparing the GitHub Repo
- Preparing the Section 2 of this report

Hussain Bukhari (33% of work)

- Reviewing literate on the topic (3 papers)
- Coding the OSGM-G, AdaGrad in quadratic case
- Preparing the GitHub Repo
- Preparing the Section 3 of this report

B. Reproducibility checklist

Answer the questions of following reproducibility checklist. If necessary, you may leave a comment.

1. A ready code was used in this project, e.g. for replication project the code from the corresponding paper was used.
   - ☐ Yes.
   - ☑ No.
   - ☐ Not applicable.

   Students' comment: None

2. A clear description of the mathematical setting, algorithm, and/or model is included in the report.
   - ☑ Yes.
   - ☐ No.
   - ☐ Not applicable.

   Students' comment: None

3. A link to a downloadable source code, with specification of all dependencies, including external libraries is included in the report.
   - ☑ Yes.
   - ☐ No.
   - ☐ Not applicable.

   Students' comment: None

4. A complete description of the data collection process, including sample size, is included in the report.
   - ☑ Yes.
   - ☐ No.
   - ☐ Not applicable.

   Students' comment: None

5. A link to a downloadable version of the dataset or simulation environment is included in the report.
   - ☑ Yes.
   - ☐ No.
   - ☐ Not applicable.

   Students' comment: None

6. An explanation of any data that were excluded, description of any pre-processing step are included in the report.
   - ☐ Yes.
   - ☑ No.
   - ☐ Not applicable.

   Students' comment: None

7. An explanation of how samples were allocated for training, validation and testing is included in the report.
   - ☐ Yes.
   - ☐ No.
   - ☑ Not applicable.

   Students' comment: None

8. The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results are included in the report.
   - ☑ Yes.
   - ☐ No.
   - ☐ Not applicable.

   Students' comment: None

9. The exact number of evaluation runs is included.
   - ☑ Yes.
   - ☐ No.
   - ☐ Not applicable.

   Students' comment: None

10. A description of how experiments have been conducted is included.
    - ☑ Yes.
    - ☐ No.
    - ☐ Not applicable.

    Students' comment: None

11. A clear definition of the specific measure or statistics used to report results is included in the report.
    - ☐ Yes.
    - ☐ No.
    - ☑ Not applicable.

    Students' comment: None

12. Clearly defined error bars are included in the report.
    - ☐ Yes.
    - ☐ No.
    - ☑ Not applicable.

    Students' comment: None

13. A description of the computing infrastructure used is included in the report.
    - ☑ Yes.
    - ☐ No.
    - ☐ Not applicable.

    Students' comment: None