Inexact Proximal Gradient Method with Optimal Support Identification

Yutong Dai[1] Daniel P. Robinson[1]

[1]Industrial and Systems Engineering, Lehigh University

INFORMS Annual Meeting 2022

October 16, 2022

Outline

Outline

Problem of Interest

Sparse optimization problem

$$\min_{x \in \mathbb{R}^n} \quad f(x) + r(x)$$

- f: loss function; L-smooth:
  - logistic regression; $f(x) = \frac{1}{N} \sum_{i=1}^{N} \log(1 + e^{-y_i x^T d_i})$
  - Huber loss function; $f(x) = \begin{cases} \frac{1}{2\mu} \|x\|^2, & \|x\| \le \mu \\ \|x\| - \frac{\mu}{2}, & \|x\| > \mu \end{cases}$
  - `tanh` activation function; $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.
- r: group sparsity inducing regularizer; convex and nonsmooth:
  - group ℓ_1: $r(x) = \sum_{i \in n_{\mathcal{G}}} \lambda_i \|[x]_{g_i}\|_2 \quad \left(\lambda_i > 0 \text{ for all } i \in n_{\mathcal{G}} \text{ and } \bigcup_{i \in n_{\mathcal{G}}} g_i = [n]\right)$
  - Example: for $x \in \mathbb{R}^3$

non-overlapping $\quad g_1 = \{1, 2\}$ and $g_2 = \{3\} : r(x) = \lambda_1 \left\| \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \right\| + \lambda_2 \|x_3\|$.

overlapping $\quad g_1 = \{1, 2\}$ and $g_2 = \{2, 3\} : r(x) = \lambda_1 \left\| \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \right\| + \lambda_2 \left\| \begin{pmatrix} x_2 \\ x_3 \end{pmatrix} \right\|$.

- problems arise in signal processing and machine learning applications
  - jointly select genes that regulate hormone levels
- sparsity in group structure imposes more optimization challenges

Brief Literature Review

- **First Order Methods**
  - (Accelerated) Proximal Gradient Method: ISTA/FISTA [Donoho, 1995, Beck and Teboulle, 2009]

$$x_{k+1} \leftarrow \arg\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2\alpha_k} \|x - \big(x_k - \alpha_k \nabla f(x_k)\big)\|_2^2 + r(x) \right\}$$

- **Second Order Methods**
  - Proximal Newton Method [Lee et al., 2014]

$$x_{k+1} \leftarrow \arg\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2\alpha_k} \|x - \big(x_k - \alpha_k H_k^{-1} \nabla f(x_k)\big)\|_{H_k}^2 + r(x) \right\}$$

- **Other Methods**
  - Stochastic Settings: SAGA[Defazio et al., 2014] and ProxSVRG[Xiao and Zhang, 2014]

$$x_{k+1} \leftarrow \arg\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2\alpha_k} \|x - \big(x_k - \alpha_k d_k\big)\|_2^2 + r(x) \right\}$$

  with d_k being some form of stochastic gradient estimator for $\nabla f(x_k)$.

Challenges

Algorithm Proximal Gradient Method - Skeleton

Initialization: pick $x_0 \in \text{int}(\text{dom}(f))$.
while not converged **do**
  ...
  Choose some $\alpha_k > 0$;
  Compute $x_{k+\frac{1}{2}} = \arg\min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2\alpha_k} \|x - \big(x_k - \alpha_k \nabla f(x_k)\big)\|_2^2 + r(x) \right\}$;
  ...
end while

However, all aforementioned methods require the exact solve of the sub-problem. What if we cannot solve the sub-problem exactly?

1: This is too hard and I give up.

2: Solve the sub-problem as accurate as possible and hope for the good.

3: OR ...

Outline

Inexact Proximal Gradient

Algorithm Inexact Proximal Gradient Method - Skeleton

Initialization: pick $x_0 \in \text{int}(\text{dom}(f))$.
while not converged **do**
    ...
    Choose some $\alpha_k > 0$;
    Compute $\hat{x}_{k+1} \approx \arg\min_{x \in \mathbb{R}^n} \left\{ \phi_p(x) := \frac{1}{2\alpha_k} \| x - \left(x_k - \alpha_k \nabla f(x_k) \right) \|_2^2 + r(x) \right\}$;
    ...
    Get the next iterate x_{k+1};
end while

Define ϵ_k accurate solution $\hat{x}_{k+1}$ as $\phi_p(\hat{x}_{k+1}) - \phi_p^* \leq \epsilon_k$ for any $k \geq 1$

- `Option_1`: $\epsilon_k = \gamma_1 \|\hat{x}_{k+1} - x_k\|^2$ (ours)
- `Option_2`: $\epsilon_k = \gamma_2 \left(\phi(x_k) - \phi_p^* \right)$ ([Lee and Wright, 2019])
- `Option_3`: $\epsilon_k = \mathcal{O}\left(1/k^{\delta} \right)$ with $\delta > 2$ ([Schmidt et al., 2011])

Wait.... how could it be practical as one needs to know ϕ_p^*!

Inexact Proximal Gradient: Test the termination conditions

$$\min_{x\in\mathbb{R}^n} \phi_p(x) := \frac{1}{2\alpha_k}\|x - \big(x_k - \alpha_k \nabla f(x_k)\big)\|_2^2 + r(x)$$

$$\max_{y\in\mathcal{Y}} \phi_d(y) \text{ for some } \mathcal{Y} \text{ and } \phi_d$$

Design a primal-dual type sub-problem solver. For the any primal-dual solution pair $(\hat{x}_{k+1}, \hat{y}_{k+1})$, and define the duality gap

$$\text{Gap}_k := \phi_p(\hat{x}_{k+1}) - \phi_d(\hat{y}_{k+1})$$

Since $\phi_p(\hat{x}_{k+1}) - \phi_k^* \le \text{Gap}_k$, then

- $\text{Gap}_k \le \gamma_1 \|\hat{x}_{k+1} - x_k\|^2$ implies `Option_1`
- $\text{Gap}_k \le \gamma_2(\phi_p(x_k) - \phi_d(\hat{y}_{k+1}))$ implies `Option_2`
- $\text{Gap}_k \le \mathcal{O}\left(1/k^\delta\right)$ implies `Option_3`

For `Option_2`, [Lee and Wright, 2019] points out that for any solver (e.g. `SpaRSA` [Wright et al., 2009]) that has ρ-linear rate convergence for solving the sub-problem, suffice it to run $\mathcal{O}(\gamma_2/\log\rho)$ number of iterations and then just terminate.

Global convergence

Assumptions:

- f is a C^1 function with ∇f Lipschitz continuous; proper, and closed;
- r are convex, proper, and closed
- $f + r$ is bounded below

Theorem 1 (worst-case complexity, informal)

For $\epsilon \in (0, \infty)$, the maximum number of iterations required before x_k becomes the ϵ-approximate stationary point is $\mathcal{O}\left(\epsilon^{-2}\right)$.

Remark: This is the same complexity as if the sub-problem is solved exactly.

Outline

Preliminaries

support identification

The support of a point $x \in \mathbb{R}^n$ is defined as

$$\mathcal{S}(x) = \{i \in \{1, \ldots, n_{\mathcal{G}}\} \mid [x]_{g_i} \neq 0\}.$$

We say that support identification happens at point $x \in \mathbb{R}^n$ for a solution $x^* \in \mathbb{R}^n$ to the problem if $\mathcal{S}(x) = \mathcal{S}(x^*)$.

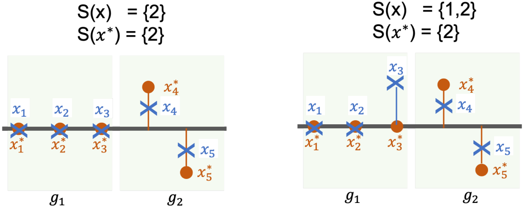


Figure: Support identification. The solution $x^* \in \mathbb{R}^5$ with group structures $g_1 = \{1, 2, 3\}$ and $g_2 = \{4, 5\}$. Support identification happens at the x for the left figure while not for the right one.

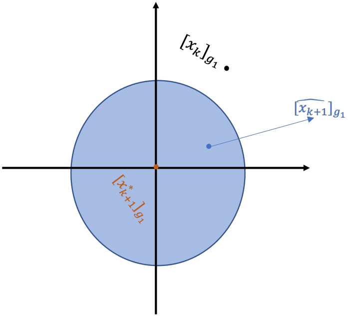
Why we care about the support identificaiton?

- Terminate the algorithm before finding the solution (variable selection problems)
- Design high-order methods (subspace acceleration)

Challenge

Assume $\hat{x}_{k+1}$ and x^*_{k+1} are the inexact solution and exact solution to the sub-problem and $\mathcal{S}(x^*_{k+1}) = \mathcal{S}(x^*)$.

Regardless of how accurately the sub-problem is being approximately solved, it is not guaranteed that $\mathcal{S}(\hat{x}_{k+1}) = \mathcal{S}(x^*_{k+1})$!



A sub-solver that exploits the geometric property of the $r(x)$ is required.

Case Study: Overlapping-Group ℓ_1 regularizer

- Formulation:

$$r(x) = \sum_{i \in n_{\mathcal{G}}} \lambda_i \|[x]_{g_i}\|_2 \text{ with } \lambda_i > 0 \text{ for all } i \in n_{\mathcal{G}} \text{ and } \bigcup_{i \in n_{\mathcal{G}}} g_i = [n] \tag{1}$$

where $[x]_{g_i}$ is a sub-vector of x whose coordinates are in the group g_i.

- Example:

$$g_1 = \{1, 2, 3\}, g_2 = \{3, 4, 5\}, g_3 = \{1, 3, 5\}.$$

Primal-Dual Problem Pair for the proximal subproblem

To avoid the cluttered notations, we introduce $u_k := x_k - \alpha_k \nabla f(x_k)$, then the sub-problem and its dual problem can be written as

$$\min_{x\in\mathbb{R}^n} \left\{ \phi_p(x; x_k, \alpha_k) := \frac{1}{2\alpha_k}\|x - u_k\|^2 + \sum_{i=1}^{n_\mathcal{G}} [\lambda]_i \|[x]_{g_i}\| \right\}$$

$$\downarrow$$

$$\begin{cases} \min_{x,z} \ \frac{1}{2\alpha_k}\|x - u_k\|^2 + \lambda^T z \\ \text{s.t.} \quad \begin{bmatrix} [x]_{g_i} \\ [z]_i \end{bmatrix} \in \mathcal{K}_i := \left\{ \begin{bmatrix} v \\ \theta \end{bmatrix} \mid v \in \mathbb{R}^{|g_i|}, \theta \in \mathbb{R}, \text{ and } \|v\| \le \theta \right\} \text{ for all } i \in [n_\mathcal{G}] \end{cases}$$

Primal-Dual Problem Pair for the Proximal Subproblem: Cont'

Primal-Dual Problem Pair

$$\min_{x\in\mathbb{R}^n} \left\{ \phi_p(x; x_k, \alpha_k) := \frac{1}{2\alpha_k}\|x-u_k\|^2 + \sum_{i=1}^{n_{\mathcal{G}}} [\lambda]_i \|[x]_{g_i}\| \right\} \tag{2}$$

$$\max_{\hat{y}\in\mathcal{F}_d} \left\{ \phi_d(\hat{y}; x_k, \alpha_k) := -\frac{\alpha_k}{2}\|A\hat{y}\|^2 - u_k^T A\hat{y} \right\}, \tag{3}$$

1. $\mathcal{M}$ is a set value mapping that relates $[x]_{g_i}$ to $[\hat{y}]_{\mathcal{M}(i)}$;
2. $\mathcal{F}_d := \{\hat{y} \in \mathbb{R}^{\sum_{i=1}^{n_{\mathcal{G}}} |g_i|} \mid \|[\hat{y}]_{\mathcal{M}(i)}\| \le [\lambda]_i \text{ for each } i \in [n_{\mathcal{G}}]\}$
3. A is a sparse, full column-rank, and flat matrix.

Primal-Dual Problem Pair for the Proximal Subproblem: Cont'

Primal-Dual Problem Pair

$$\min_{x\in\mathbb{R}^n} \left\{ \phi_p(x; x_k, \alpha_k) := \frac{1}{2\alpha_k}\|x - u_k\|^2 + \sum_{i=1}^{n_{\mathcal{G}}} [\lambda]_i \|[x]_{g_i}\| \right\}$$

$$\max_{\hat{y}\in\mathcal{F}_d} \left\{ \phi_d(\hat{y}; x_k, \alpha_k) := -\frac{\alpha_k}{2}\|A\hat{y}\|^2 - u_k^T A\hat{y} \right\},$$

Example 2

Consider the group structure for problem (1) given by

$$g_1 = \{1,2,3\}, \quad g_2 = \{2,3,4\}, \quad \text{and} \quad g_3 = \{1,3,5\}.$$

$$A\hat{y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \hdashline y_4 \\ y_5 \\ y_6 \\ \hdashline y_7 \\ y_8 \\ y_9 \end{bmatrix}$$

Solve the Primal-Dual Subproblem

Primal-Dual Problem Pair

$$x_k^* = \arg\min_{x \in \mathbb{R}^n} \left\{ \phi_p(x; x_k, \alpha_k) := \frac{1}{2\alpha_k} \|x - u_k\|^2 + \sum_{i=1}^{n_{\mathcal{G}}} [\lambda]_i \|[x]_{g_i}\| \right\} \quad (4)$$

$$\hat{\mathcal{Y}}(x_k, \alpha_k) = \operatorname{Arg}\max_{\hat{y} \in \mathcal{F}_d} \left\{ \phi_d(\hat{y}; x_k, \alpha_k) := -\frac{\alpha_k}{2} \|A\hat{y}\|^2 - u_k^T A\hat{y} \right\}, \quad (5)$$

Lemma 3 (linking equation)

The unique solution x_k^* *satisfies* $x_k^* = u_k + \alpha_k A\hat{y}_k^*$ *for all* $\hat{y}_k^* \in \hat{\mathcal{Y}}(x_k, \alpha_k)$.

Lemma 4

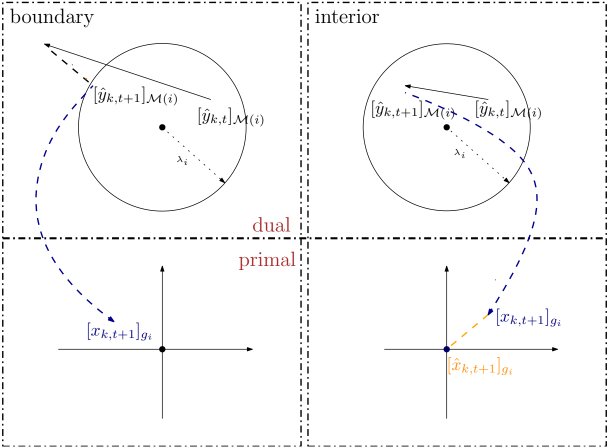
Let $i \in [n_{\mathcal{G}}]$. *If there exists* $\hat{y}_k^* \in \hat{\mathcal{Y}}(x_k, \alpha_k)$ *satisfying* $\left\|[\hat{y}_k^*]_{\mathcal{M}(i)}\right\| < [\lambda]_i$, *then* $[x_k^*]_{g_i} = 0$.

Primal-Dual Subproblem Solver: graphical demo

Enhanced Projected Gradient Dual Gradient Ascent

Given the tth iterate $\hat{y}_{k,t}$:

1. Get $\hat{y}_{k,t+1} \leftarrow \texttt{PGA}(\hat{y}_{k,t+1})$.
2. Construct a trail primal iterate $x_{k,t+1} \leftarrow u_k + \alpha_k A\hat{y}_{k,t+1}$.
3. Project $[x_{k,t+1}]_{g_i}$ to 0 based on if $\|[\hat{y}_{k,t+1}]_{g_i}\| < \lambda_i - \epsilon_{k-1}$ for all $i \in [n_{\mathcal{G}}]$.



Support Identification Complexity

Assumption 3.1

- (non-degeneracy) The quantity

$$\delta_{\text{nd}} := \begin{cases} \min_{\hat{y}\in\hat{\mathcal{Y}}(x^*,\alpha^*), i\notin\mathcal{S}(x^*)} \left([\lambda]_i - \|[\hat{y}]_{\mathcal{M}(i)}\|\right) & \text{if } \mathcal{S}(x^*) \subsetneq [n_{\mathcal{G}}], \\ 1 & \text{if } \mathcal{S}(x^*) = [n_{\mathcal{G}}], \end{cases}$$

  satisfies $\delta_{\text{nd}} > 0$. It follows that $\delta^* := \min\{1, \delta_{\text{nd}}\} \in (0, 1]$.
- f is μ_f strongly convex and L_g smooth.

Define $\Theta := \begin{cases} \min\{1, \min_{i\in\mathcal{S}(x^*)} \|[x^*]_{g_i}\|\} & \text{if } \mathcal{S}(x^*) \neq \emptyset, \\ 1 & \text{otherwise.} \end{cases}$ $\theta := (1 - \mu_f/L_g) \in [\eta, 1)$

Theorem 5 (Support identification complexity)

For some $\omega \in (0,1)$, the sequence $\{\epsilon_k\}$ satisfies $\epsilon_{k+1} \le \omega^2\epsilon_k$. Then, under the Assumption 3.1 $\mathcal{S}(x_{k+1}) = \mathcal{S}(x^*)$ for all $k \ge K$ with

$$K := \begin{cases} \max\left(\mathcal{O}\left(\frac{\log\Theta}{\log\theta}\right), \mathcal{O}\left(\frac{\log\delta^*}{\log\left(\max\{\omega^{\rho_{\min}}, \theta^{\rho^*}, \omega^{2\iota}\}\right)}\right)\right) & \textit{if } \omega < \theta, \\ \max\left(\mathcal{O}\left(\frac{\log\Theta}{\log\omega}\right), \mathcal{O}\left(\frac{\log\delta^*}{\log\left(\max\{\omega^{\min\{\rho_{\min}, \rho^*\}}, \omega^{2\iota}\}\right)}\right)\right) & \textit{if } \omega > \theta, \\ \max\left(\mathcal{O}(C_\Theta), \mathcal{O}(C_{\delta^*})\right) & \textit{if } \omega = \theta. \end{cases}$$

Outline

Setup

- Logistic loss with overlapping group ℓ_1 regularizer

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} \log\left(1 + e^{-y_i x^T d_i}\right).$$

- 132 problem instances created based 11 datasets from LIBSVM
- Compare `Option_1`, `Option_2`, and `Option_3` in terms of the solution sparsity, solution quality, running time
  - `Option_1`: $\epsilon_k = \gamma_1 \|\hat{x}_{k+1} - x_k\|^2$
  - `Option_2`: $\epsilon_k = \gamma_2 \left(\phi(x_k) - \phi_p^*\right)$
  - `Option_3`: $\epsilon_k = \mathcal{O}\left(1/k^\delta\right)$

Sensitivity to parameters

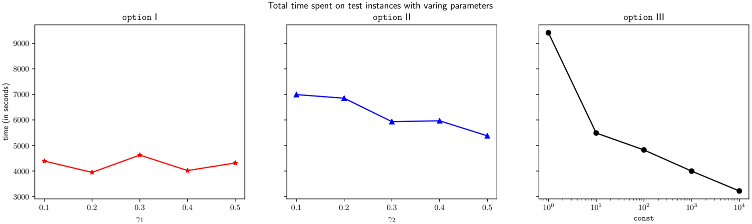


Figure: Compare the performance in CPU time for three options with different algorithm parameters. γ_1 for `Option_1` and γ_2 for `Option_2` are both selected from $\{0.1,0.2,0.3,0.4,0.5\}$ and `const` for `Option_3` is selected from $\{10^i\}_{i=0}^4$.

Time comparison

	approximate solution found	maximum iteration limit	maximum time limit	numerical difficulties
Option_1	108	16	7	1
Option_2	107	15	8	2
Option_3	107	16	9	0

Table: Termination status summary for the three algorithm variants Option_1, Option_2, and Option_3 on the 132 test instances with our subproblem solver.

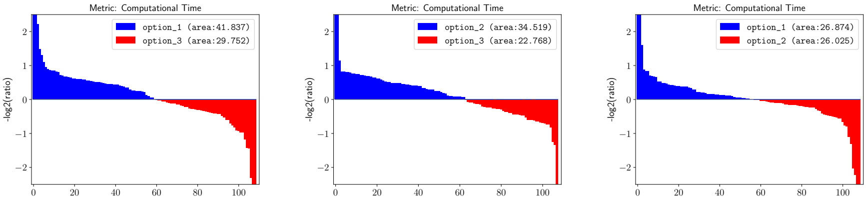


Figure: A performance profile for CPU time (seconds). In each plot, we exclude problem instances for which both algorithms fail.

$$\text{height of the bar} = -\ln\left(\frac{\text{metric of one algorithm}}{\text{metric of another algorithm}}\right)$$

Enhanced Projected Gradient Ascent v.s. Vanilla Projected Gradient Ascent

		IPG+EPGA			IPG+PGA		
data set	Λ	#z	#nz	F	#z	#nz	F
a9a	0.013458	12	2	0.508337	0	14	0.508337
colon-cancer	0.017751	213	10	0.336270	1	222	0.336270
duke breast-cancer	0.016198	779	13	0.246910	2	790	0.246910
gisette	0.012003	536	20	0.402671	2	554	0.402671
leukemia	0.020514	781	11	0.258627	0	792	0.258627
madelon	0.000402	19	37	0.666079	0	56	0.666112
mushrooms	0.009528	10	3	0.316138	0	13	0.316138
w8a	0.006687	24	10	0.429029	0	34	0.429029

Table: The test results for IPG using EPGA or PGA algorithm as the subproblem solvers. Columns "#z", "#nz", and "F" give the number of zero groups, the number of non-zero groups, and the final objective value, respectively.

Summary

- Discussed two **adaptive** and **implementable** termination conditions for the inexact proximal gradient method (IPG) and provided unified convergence analysis.
- Crafted a specialized proximal subproblem solver to enable the support identification property of the IPG method when using the overlapping group ℓ_1 regularizer.
- Derived the support identification complexity for IPG method when using the overlapping group ℓ_1 regularizer.

Q&A

Thank you and Questions?

Contact: yud319@lehigh.edu

References I

[Beck and Teboulle, 2009] Beck, A. and Teboulle, M. (2009).
A fast iterative shrinkage-thresholding algorithm for linear inverse problems.
SIAM Journal on Imaging Sciences, 2(1):183–202.

[Defazio et al., 2014] Defazio, A., Bach, F., and Lacoste-Julien, S. (2014).
Saga: a fast incremental gradient method with support for non-strongly convex composite objectives.
In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 1*, pages 1646–1654.

[Donoho, 1995] Donoho, D. (1995).
Denoising by soft-thresholding.
Trans. Inform. Theory, 41:613–627.

[Lee and Wright, 2019] Lee, C.-p. and Wright, S. J. (2019).
Inexact successive quadratic approximation for regularized optimization.
Computational Optimization and Applications, 72(3):641–674.

[Lee et al., 2014] Lee, J. D., Sun, Y., and Saunders, M. A. (2014).
Proximal newton-type methods for minimizing composite functions.
SIAM Journal on Optimization, 24(3):1420–1443.

References II

[Schmidt et al., 2011] Schmidt, M., Roux, N. L., and Bach, F. (2011).
Convergence rates of inexact proximal-gradient methods for convex optimization.

[Wright et al., 2009] Wright, S. J., Nowak, R. D., and Figueiredo, M. A. (2009).
Sparse reconstruction by separable approximation.
IEEE Transactions on Signal Processing, 57(7):2479–2493.

[Xiao and Zhang, 2014] Xiao, L. and Zhang, T. (2014).
A proximal stochastic gradient method with progressive variance reduction.
SIAM Journal on Optimization, 24(4):2057–2075.