Non-asymptotic Analysis of Stochastic Methods for Non-Smooth Non-Convex Regularized Problems

Yi Xu[1], Rong Jin[2], Tianbao Yang[1]

1. Computer Science Department, The University of Iowa, Iowa City, IA, USA
2. Machine Intelligence Technology, Alibaba Group, Bellevue, WA, USA

Non-convex Non-smooth Optimization Problem

Stochastic non-convex non-smooth regularized optimization problems:

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) := \underbrace{\mathrm{E}_\xi[f(\mathbf{x};\xi)]}_{f(\mathbf{x})} + r(\mathbf{x}), \tag{1}$$

where ξ is a random variable, $f(\mathbf{x}): \mathbb{R}^d \to \mathbb{R}$ is smooth non-convex , and $r(\mathbf{x}): \mathbb{R}^d \to \mathbb{R}$ is proper non-smooth non-convex lower-semicontinuous. A special case of problem (1) in machine learning is of the following finite-sum form:

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) := \underbrace{\frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x})}_{f(\mathbf{x})} + r(\mathbf{x}), \tag{2}$$

where n is the number of data samples.

- Examples of smooth non-convex losses
  - non-linear square loss for classification
  - truncated square loss for regression
  - cross-entropy loss for learning a neural network with a smooth activation function
- Examples of non-smooth non-convex regualerizers
  - ℓ_p $(0 \le p < 1)$ norm
  - smoothly clipped absolute deviation (SCAD)
  - log-sum penalty (LSP)
  - minimax concave penalty (MCP)
  - an indicator function of a non-convex constraint as well (e.g., $\|\mathbf{x}\|_0 \le k$)

Main Contributions

- Establish **the first convergence rate** of standard mini-batch SPG (MB-SPG) for solving (1) in terms of finding an approximate stationary point, which is the same as its counterpart for solving a non-convex minimization problem with a convex regularizer [1].
- Analyze improved variants of mini-batch SPG that use a recursive stochastic gradient estimator (SARAH [2,3] or SPIDER [4,5]) referred to as SPGR, and achieve **the new state of the art** convergence results for both online setting and the finite-sum setting.
- Propose **more practical** variants of MB-SPG and SPGR by using dynamic mini-batch size instead of fixed mini-batch size to remove the requirement on the target accuracy level of solution for running the algorithms.

Summary of results for finding an ϵ-stationary point

Problem	Algorithm	Sample complexity	$r(\mathbf{x})$
Online setting (1)	MBSGA [6]	$O(\epsilon^{-5})$	PM, LC
Online setting (1)	SSDC-SPG [7]	$O(\epsilon^{-5})$	PM, LC
Online setting (1)	SSDC-SPG [7]	$O(\epsilon^{-6})$	PM, FV
Online setting (1)	**MB-SPG (this work)**	$O(\epsilon^{-4})$	PM
Online setting (1)	**SPGR (this work)**	$O(\epsilon^{-3})$	PM
Finite-sum setting (2)	VRSGA [6]	$O(n^{2/3}\epsilon^{-3})$	PM, LC
Finite-sum setting (2)	SSDC-SVRG [7]	$\widetilde{O}(n\epsilon^{-3})$	PM, LC
Finite-sum setting (2)	SSDC-SVRG [7]	$\widetilde{O}(n\epsilon^{-4})$	PM, FV
Finite-sum setting (2)	**SPGR (this work)**	$O(n^{1/2}\epsilon^{-2}+n)$	PM

- LC: Lipchitz continuous function; FV: finite-valued over $\mathbb{R}^d$; PM: the proximal mapping exists and can be obtained efficiently.
- $\widetilde{O}(\cdot)$ suppresses a logarithmic factor in terms of ϵ^{-1}

Preliminaries

- $\|\mathbf{x}\|$: Euclidean norm of a vector $\mathbf{x}\in\mathbb{R}^d$
- $\mathcal{S} = \{\xi_1,\ldots,\xi_m\}$: a set of random variables; $|\mathcal{S}|$: the number of elements in set $\mathcal{S}$; $f_{\mathcal{S}}(\mathbf{x}) = \frac{1}{|\mathcal{S}|}\sum_{\xi_i\in\mathcal{S}} f(\mathbf{x};\xi_i)$
- $\mathrm{dist}(\mathbf{x},\mathcal{S})$: distance between vector $\mathbf{x}$ and set $\mathcal{S}$
- $\hat{\partial} h(\mathbf{x})$: Fréchet subgradient

$$\hat{\partial} h(\bar{\mathbf{x}}) = \left\{\mathbf{v}\in\mathbb{R}^d : \liminf_{\mathbf{x}\to\bar{\mathbf{x}}} \frac{h(\mathbf{x}) - h(\bar{\mathbf{x}}) - \mathbf{v}^\top(\mathbf{x}-\bar{\mathbf{x}})}{\|\mathbf{x}-\bar{\mathbf{x}}\|} \ge 0\right\}$$

- $\partial h(\mathbf{x})$: limiting subgradient

$$\partial h(\bar{\mathbf{x}}) = \{\mathbf{v}\in\mathbb{R}^d : \exists \mathbf{x}_k \xrightarrow{h} \bar{\mathbf{x}}, v_k \in \hat{\partial} h(\mathbf{x}_k), \mathbf{v}_k \to \mathbf{v}\}$$

- Goal: finding an ϵ-**stationary point** of problem (1), i.e., to find a solution $\mathbf{x}$ such that $\mathrm{dist}(0, \hat{\partial} F(\mathbf{x})) = \mathrm{dist}(0, \nabla f(\mathbf{x}) + \hat{\partial} r(\mathbf{x})) \le \epsilon$.

- Assumptions:
  - (i) $\mathrm{E}_\xi[\nabla f(\mathbf{x};\xi)] = \nabla f(\mathbf{x})$, and there exists a constant $\sigma > 0$, s.t. $\mathrm{E}_\xi[\|\nabla f(\mathbf{x};\xi) - \nabla f(\mathbf{x})\|^2] \le \sigma^2$.
  - (ii) Given $\mathbf{x}_0$, there exists $\Delta < \infty$ s.t. $F(\mathbf{x}_0) - F(\mathbf{x}_*) \le \Delta$, where $\mathbf{x}_*$ denotes the global minimum of (1).
  - (iii) $f(\mathbf{x})$ is smooth with a L-Lipchitz continuous gradient, i.e., it is differentiable and there exists a constant $L > 0$ s.t. $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \le L\|\mathbf{x}-\mathbf{y}\|, \forall \mathbf{x},\mathbf{y}$.
  - (iv) $r(\mathbf{x})$ is simple enough such that its proximal mapping exists and can be obtained efficiently:

$$\mathrm{prox}_{\eta r}[\mathbf{x}] = \arg\min_{\mathbf{y}\in\mathbb{R}^d} \frac{1}{2\eta}\|\mathbf{y}-\mathbf{x}\|^2 + r(\mathbf{y}).$$

Warm-up: Proximal Gradient Descent (PGD) Method

The deterministic PGD method (a.k.a. forward-backward splitting, FBS) updates the solutions for $t = 0,\ldots,T-1$ iteratively given $\mathbf{x}_0$ with a step size η:

$$\mathbf{x}_{t+1} \in \mathrm{prox}_{\eta r}[\mathbf{x}_t - \eta\nabla f(\mathbf{x}_t)] = \arg\min_{\mathbf{x}\in\mathbb{R}^d}\left\{r(\mathbf{x}) + \langle\nabla f(\mathbf{x}_t), \mathbf{x}-\mathbf{x}_t\rangle + \frac{1}{2\eta}\|\mathbf{x}-\mathbf{x}_t\|^2\right\}. \tag{3}$$

Theorem 1. *Run (3) with $\eta = \frac{c}{L}$ $(0 < c < 1)$ and $T = \frac{4(\eta^2L^2+1)}{\eta(1-\eta L)\epsilon^2}\Delta = O(1/\epsilon^2)$, with R being uniformly sampled from $\{1,\ldots,T\}$, we have $\mathrm{E}[dist(0,\hat{\partial}F(\mathbf{x}_R))] \le \epsilon$.*

Proof Sketch. For the update (3), we can only leverage its optimality condition (e.g., by Exercise 8.8 and Theorem 10.1 of [8]):

$$-\nabla f(\mathbf{x}_t) - \frac{1}{\eta}(\mathbf{x}_{t+1}-\mathbf{x}_t) \in \hat{\partial} r(\mathbf{x}_{t+1}),$$

$$r(\mathbf{x}_{t+1}) + \langle\nabla f(\mathbf{x}_t), \mathbf{x}_{t+1}-\mathbf{x}_t\rangle + \frac{1}{2\eta}\|\mathbf{x}_{t+1}-\mathbf{x}_t\|^2 \le r(\mathbf{x}_t),$$

where the first implies that $\nabla f(\mathbf{x}_{t+1}) - \nabla f(\mathbf{x}_t) - \frac{1}{\eta}(\mathbf{x}_{t+1}-\mathbf{x}_t) \in \hat{\partial} F(\mathbf{x}_{t+1})$. Combining the second inequality with the smoothness of $f(\mathbf{x})$, i.e., $f(\mathbf{x}_{t+1}) \le f(\mathbf{x}_t) + \langle\nabla f(\mathbf{x}_t), \mathbf{x}_{t+1}-\mathbf{x}_t\rangle + \frac{L}{2}\|\mathbf{x}_{t+1}-\mathbf{x}_t\|^2$, we get

$$\frac{1}{2}(1/\eta - L)\|\mathbf{x}_{t+1}-\mathbf{x}_t\|^2 \le F(\mathbf{x}_t) - F(\mathbf{x}_{t+1}). \tag{4}$$

By telescoping the above inequality and connecting $\hat{\partial}F(\mathbf{x}_{t+1})$ with $\|\mathbf{x}_{t+1}-\mathbf{x}_t\|$ we can finish the proof.

Mini-batch Stochastic Proximal Gradient (MB-SPG) Methods

Algorithm 1 Mini-Batch Stochastic Proximal Gradient: MB-SPG

1. **Initialize**: $\mathbf{x}_0 \in \mathbb{R}^d$, $\eta = \frac{c}{L}$ with $0 < c < \frac{1}{2}$.
2. **for** $t = 0,1,\ldots,T-1$ **do**
3. Draw samples $\mathcal{S}_t = \{\xi_1,\ldots,\xi_{m_t}\}$, let $\mathbf{g}_t = \frac{1}{m_t}\sum_{i=1}^{m_t}\nabla f(\mathbf{x}_t;\xi_i)$
4. $\mathbf{x}_{t+1} \in \mathrm{prox}_{\eta r}[\mathbf{x}_t - \eta\mathbf{g}_t]$
5. **end for**
6. **Output**: $\mathbf{x}_R$, where R is uniformly sampled from $\{1,\ldots,T\}$.

Theorem 2. *Run Algorithm 1 with $\eta = \frac{c}{L}$ $(0 < c < \frac{1}{2})$, then*

$$\mathrm{E}[dist(0,\hat{\partial}F(\mathbf{x}_R))^2] \le \frac{c_1}{T}\sum_{t=0}^{T-1}\mathrm{E}[\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|^2] + \frac{c_2\Delta}{\eta T},$$

where $c_1 = \frac{2c(1-2c)+2}{c(1-2c)}$ and $c_2 = \frac{6-4c}{1-2c}$ are two positive constants. In particular,

(a) *(fixed mini-batch size) by setting $T = 2c_2\Delta/(\eta\epsilon^2)$ and $m_t = 2c_1\sigma^2/\epsilon^2$ for $t = 0,\ldots,T-1$, then $\mathrm{E}[dist(0,\hat{\partial}F(\mathbf{x}_R))^2] \le \epsilon^2$. The total complexity is $O(1/\epsilon^4)$.*

(b) *(increasing mini-batch sizes) by setting $T = \widetilde{O}(1/\epsilon^2)$ and mini-batch sizes $m_t = b(t+1)$ for $t = 0,\ldots,T-1$, where $b > 0$ is a constant, then $\mathrm{E}[dist(0,\hat{\partial}F(\mathbf{x}_R))^2] \le \epsilon^2$. The total complexity is $\widetilde{O}(1/\epsilon^4)$.*

Remark: Although using increasing mini-batch sizes has an additional logarithmic factor in the complexity than that using a fixed mini-batch size, it would be more practical and user-friendly because it does not require knowing the target accuracy ϵ to run the algorithm .

SPG Methods with Recursive Stochastic Gradient Estimator (SPGR)

Algorithm 2 Stochastic Proximal Gradient using SPIDER/SARAH: SPGA($\mathbf{x}_0$, T, q, L, c)

1. **Input**: $\mathbf{x}_0 \in \mathbb{R}^d$, the number of iterations T, $\eta = \frac{c}{L}$ with $0 < c < \frac{1}{6}$.
2. **for** $t = 0,1,\ldots,T-1$ **do**
3. **if** $\mathrm{mod}(t,q) == 0$ **then**
4. Draw samples $\mathcal{S}_1$, let $\mathbf{g}_t = \nabla f_{\mathcal{S}_1}(\mathbf{x}_t)$ // For finite-sum setting, $|\mathcal{S}_1| = n$
5. **else**
6. Draw samples $\mathcal{S}_2$, let $\mathbf{g}_t = \nabla f_{\mathcal{S}_2}(\mathbf{x}_t) - \nabla f_{\mathcal{S}_2}(\mathbf{x}_{t-1}) + \mathbf{g}_{t-1}$
7. **end if**
8. $\mathbf{x}_{t+1} \in \mathrm{prox}_{\eta r}[\mathbf{x}_t - \eta\mathbf{g}_t]$
9. **end for**
10. **Output**: $\mathbf{x}_R$, where R is uniformly sampled from $\{1,\ldots,T\}$.

Theorem 3. *Under an additional assumption that $f(\mathbf{x};\xi)$ is L-smooth, run Algorithm 2 with $\eta = \frac{c}{L}$ $(0 < c < \frac{1}{3})$ and $q = |\mathcal{S}_2|$, then*

$$\mathrm{E}[dist(0,\hat{\partial}F(\mathbf{x}_R))^2] \le \frac{2\theta\Delta + \gamma\eta\Delta}{\eta\theta T} + \frac{(\gamma + 4\theta L)\sigma^2}{2\theta L|\mathcal{S}_1|}$$

*for **online setting** and*

$$\mathrm{E}[dist(0,\hat{\partial}F(\mathbf{x}_R))^2] \le \frac{2\theta\Delta + \gamma\eta\Delta}{\eta\theta T}$$

*for **finite-sum setting**, where $\gamma = 4L^2 + \frac{1}{\eta^2} + \frac{2L}{\eta}$ and $\theta = \frac{1-3\eta L}{2\eta}$ are two positive constants. In order to have $\mathrm{E}[dist(0,\hat{\partial}F(\mathbf{x}_R))] \le \epsilon$ we can set:*

(a) *(Online setting) $q = |\mathcal{S}_2| = \sqrt{|\mathcal{S}_1|}$, $|\mathcal{S}_1| = \frac{(\gamma+4\theta L)\sigma^2}{\theta L\epsilon^2}$, and $T = \frac{2(2\theta+\gamma\eta)\Delta}{\eta\theta\epsilon^2}$. The total complexity is $O(\epsilon^{-3})$.*

(b) *(Finite-sum setting) $q = |\mathcal{S}_2| = \sqrt{n}$, $|\mathcal{S}_1| = n$, and $T = \frac{(2\theta+\gamma\eta)\Delta}{\eta\theta\epsilon^2}$. The total complexity is $O(\sqrt{n}\epsilon^{-2} + n)$.*

Remark: We also proposed a increasing mini-batch sizes version of SPGR.

Numerical Experiments

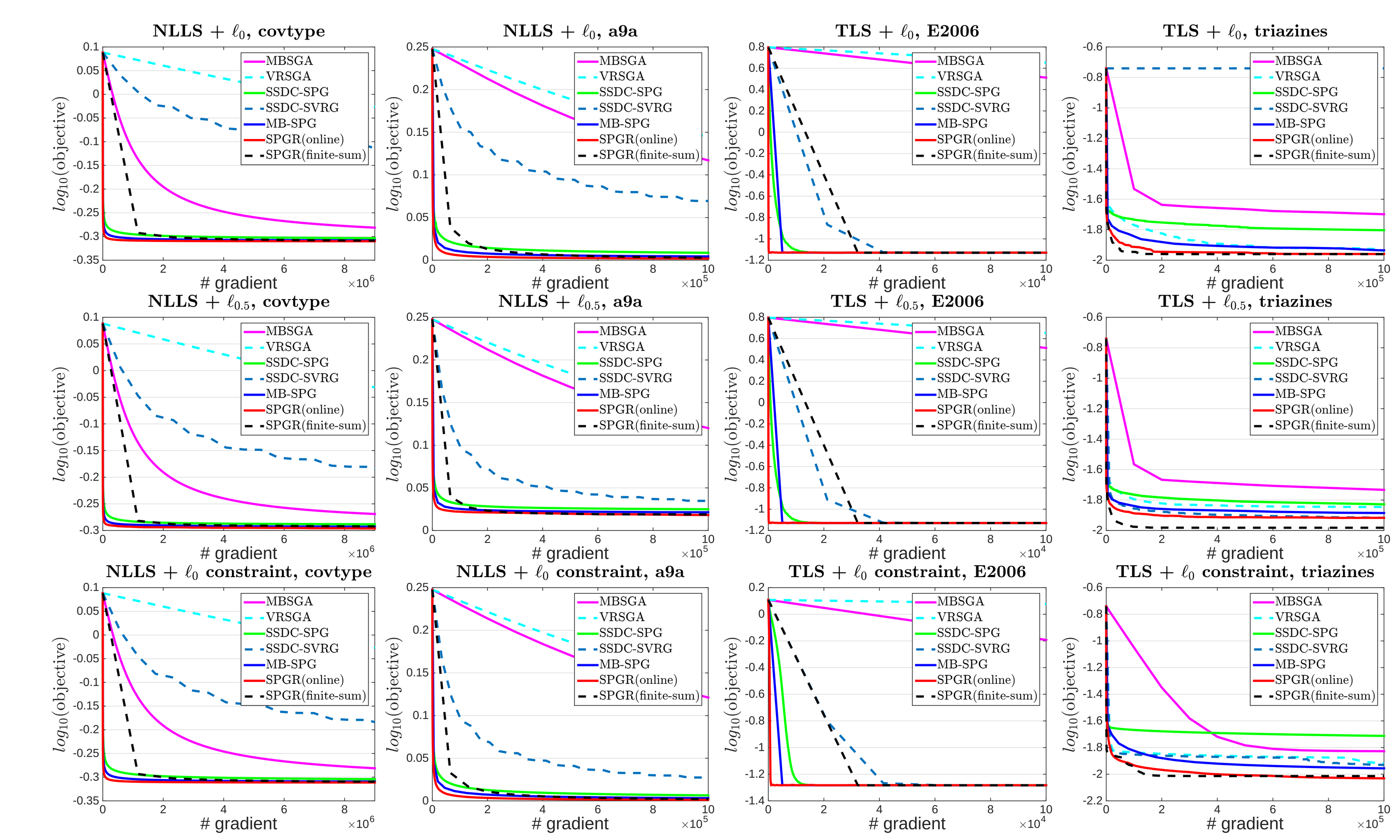

Figure: Comparisons of different algorithms for regularized loss minimization.

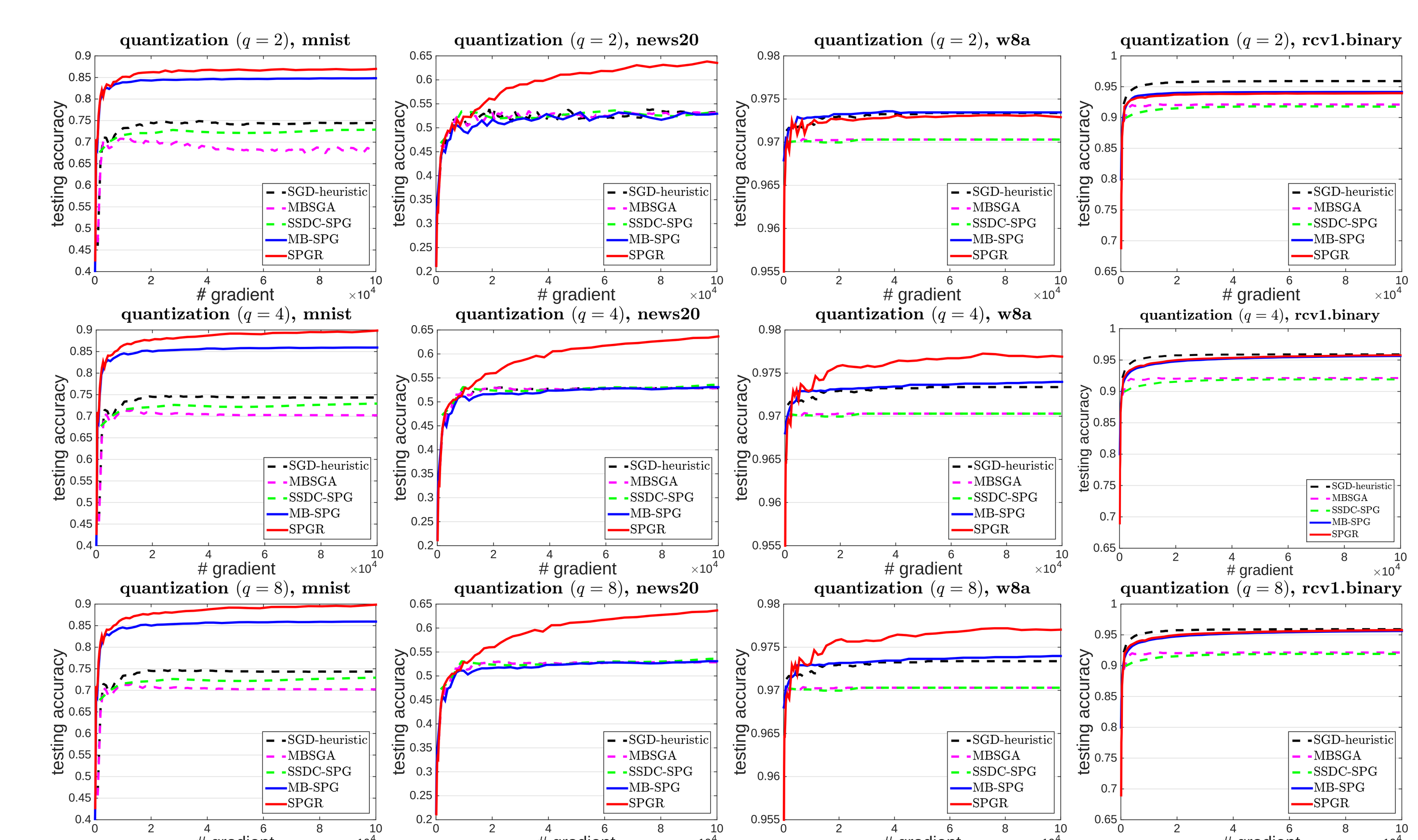

Figure: Comparisons of different algorithms for learning with quantization.

References

1. S. Ghadimi, G. Lan, and H. Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. Mathematical Programming, 155(1-2):267-305, 2016.
2. L. M Nguyen, J. Liu, K. Scheinberg, and M. Takác. SARAH: A novel method for machine learning problems using stochastic recursive gradient. ICML, pp. 2613?2621, 2017
3. L. M Nguyen, J. Liu, K. Scheinberg, and M. Takác. Stochastic recursive gradient algorithm for nonconvex optimization. arXiv preprint arXiv:1705.07261, 2017
4. C. Fang, C. J. Li, Z. Lin, and T. Zhang. Spider: Near-optimal non- convex optimization via stochastic path-integrated differential estimator. NeurIPS, pp. 687?697, 2018.
5. Z. Wang, K. Ji, Y. Zhou, Y. Liang, and V. Tarokh. SpiderBoost: A class of faster variance-reduced algorithms for nonconvex optimization. arXiv preprint arXiv:1810.10690, 2018.
6. M. R Metel and A. Takeda. Stochastic gradient methods for non-smooth non-convex regularized optimization. ICML, pp. 4537-4545, 2019.
7. Y. Xu, Q. Qi, Q. Lin, R. Jin, and T. Yang. Stochastic optimization for dc func- tions and non-smooth non-convex regularizers with non-asymptotic convergence. ICML, pp. 6942-6951, 2019.
8. R. T. Rockafellar and R. J.-B. Wets. Variational Analysis. Springer Verlag, Heidel-berg, Berlin, New York, 1998.