Stochastic Variance Reduced Primal Dual Algorithms for Empirical Composition Optimization

33rd Conference on Neural Information Processing Systems

Vancouver, Canada

Adithya M. Devraj

Department of ECE, University of Florida, Gainesville, FL

Jianshu Chen

Tencent AI Lab, Bellevue, WA

Outline

Goal in Empirical Composition Optimization

Goal: Find θ^* such that:

$$\theta^* = \arg\min_{\theta} \frac{1}{n_X} \sum_{i=0}^{n_X-1} \phi_i\left(\frac{1}{n_{Y_i}} \sum_{j=0}^{n_{Y_i}-1} f_\theta(x_i, y_{ij}) \right) + g(\theta)$$

where,

- $(x_i, y_{ij}) \in \mathbb{R}^{m_x} \times \mathbb{R}^{m_y}$ is the (i,j)-th data sample
- $f_\theta : \mathbb{R}^{m_x} \times \mathbb{R}^{m_y} \to \mathbb{R}^\ell$ is parameterized by $\theta \in \mathbb{R}^d$
- $\phi_i : \mathbb{R}^\ell \to \mathbb{R}^+$ is a convex *merit function*
- $g(\theta)$ is a μ-strongly convex regularizer

Motivating Examples

- Unsupervised sequence classification [1]:

$$\min_{\theta}\left\{-\sum_{i=0}^{n_X-1} p_{\mathsf{LM}}(x_i)\log\left(\frac{1}{n_Y}\sum_{j=0}^{n_Y-1} f_\theta(x_i,y_j)\right)\right\}$$

  - $p_{\mathsf{LM}}:\mathbb{R}^{m_x}\to[0,1]$: Known language model
  - $f_\theta:\mathbb{R}^{m_x}\times\mathbb{R}^{m_y}\to\mathbb{R}^{\ell}$: Predicted n-gram frequency by a parameterized sequence classifier

- Risk-averse learning[2]:

$$\min_{\theta}\left\{-\frac{1}{n}\sum_{i=0}^{n-1}\langle x_i,\theta\rangle+\frac{1}{n}\sum_{i=0}^{n-1}\left(\langle x_i,\theta\rangle-\frac{1}{n}\sum_{j=0}^{n-1}\langle x_j,\theta\rangle\right)^2\right\}$$

  - $x_i\in\mathbb{R}^d$: Vector consisting of the rewards from d assets at time i
  - $\theta\in\mathbb{R}^d$: Weight vector on the d assets

[1] Y. Liu, J. Chen, & L. Deng, *Unsupervised sequence classification using sequential output statistics*, NeurIPS, 2017

[2] M.Wang, E. Fang, & H. Liu, *Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions*, Mathematical Programming, 2017

Challenges: Biased Gradients

- Gradient of the objective with respect to θ:

$$\frac{1}{n_X}\sum_{i=0}^{n_X-1}\left[\frac{1}{n_{Y_i}}\sum_{j=0}^{n_{Y_i}-1}\frac{\partial}{\partial\theta}f_\theta(x_i,y_{ij})\right]^{T}\left[\phi_i'\left(\frac{1}{n_{Y_i}}\sum_{j=0}^{n_{Y_i}-1}f_\theta(x_i,y_{ij})\right)\right]+\frac{\partial}{\partial\theta}g(\theta)$$

- The following "sampled stochastic gradient" is biased:

$$\left[\frac{\partial}{\partial\theta}f_\theta(x_i,y_{ij})\right]^{T}\left[\phi_i'\big(f_\theta(x_i,y_{ij})\big)\right]+\frac{\partial}{\partial\theta}g(\theta)$$

- *Cannot* directly apply SGD-like techniques

Related Work: A Special Case

- Goal: Find θ^* such that:

$$\min_\theta \frac{1}{n_X} \sum_{i=0}^{n_X-1} \phi_i\left(\frac{1}{n_Y} \sum_{j=0}^{n_Y-1} f_\theta(y_j)\right) \quad \text{vs} \quad \min_\theta \frac{1}{n_X} \sum_{i=0}^{n_X-1} \phi_i\left(\frac{1}{n_{Y_i}} \sum_{j=0}^{n_{Y_i}-1} f_\theta(x_i, y_{ij})\right)$$

- Key difference: Inner function f_θ does not depend on outside summation
- Related algorithms:
  - Compositional SGD [Wang, Fang & Liu, 2017]: Two-time-scale algorithm to deal with the two expectations separately
  - Compositional SVRG [Lian, Wang & Liu, 2017]: SVRG for compositional optimization
  - C-SAGA [Zhang & Xiao, 2019]: SAGA for compositional optimization

Our Approach: Primal Dual Formulation

For $\psi : \mathbb{R}^\ell \to \mathbb{R}$, its convex conjugate $\psi^* : \mathbb{R}^\ell \to \mathbb{R}$ is defined:

$$\psi^*(y) = \sup_{x \in \mathbb{R}^\ell} \left(\langle x, y \rangle - \psi(x) \right) \quad \overset{\text{Strong Duality}}{\Leftrightarrow} \quad \psi(x) = \sup_{y \in \mathbb{R}^\ell} \left(\langle x, y \rangle - \psi^*(y) \right)$$

Applying to our problem: Transformed min-max objective

$$\min_\theta \max_w \left\{ \underbrace{\frac{1}{n_X} \sum_{i=0}^{n_X - 1} \left[\left\langle \frac{1}{n_{Y_i}} \sum_{j=0}^{n_{Y_i}-1} f_\theta(x_i, y_{ij}), w_i \right\rangle - \phi_i^*(w_i) \right] + g(\theta)}_{L(\theta, w)} \right\}$$

- $w := \{w_0, \dots, w_{n_X - 1}\}$
- No more non-linear compositions of empirical averages
- Can sample unbiased gradients with respect to w_i's and θ
- Maximization is decoupled over w_i's

Our Algorithms: Key Ideas (I)

Dual step: Stochastic variance reduced coordinate ascent

- Batch gradient ascent for dual variables: For each $1 \leq i \leq n_X$,

$$w_i^{(k)} = \arg\min_{w_i} \left\{ -\left\langle \frac{1}{n_{Y_i}} \sum_{j=0}^{n_{Y_i}-1} f_{\theta^{(k-1)}}(x_i, y_{ij}), w_i \right\rangle + \phi_i^*(w_i) + \frac{1}{2\alpha_w} \|w_i - w_i^{(k-1)}\|^2 \right\}$$

  - Evaluating the full batch gradient is expensive
  - Updating each of the n_X variables is also expensive

Our Algorithms: Key Ideas (I)

Dual step: Stochastic variance reduced coordinate ascent

- Batch gradient ascent for dual variables: For each $1 \leq i \leq n_X$,

$$w_i^{(k)} = \arg\min_{w_i} \Big\{ -\Big\langle \frac{1}{n_{Y_i}} \sum_{j=0}^{n_{Y_i}-1} f_{\theta^{(k-1)}}(x_i, y_{ij}), w_i \Big\rangle + \phi_i^*(w_i) + \frac{1}{2\alpha_w} \|w_i - w_i^{(k-1)}\|^2 \Big\}$$

  - Evaluating the full batch gradient is expensive
  - Updating each of the n_X variables is also expensive
- Key Idea 1: *Exploit decoupled dual maximization* over w_i's:
  - At each iteration k, randomly sample an index i and update w_i
  - Keep other $\{w_j,\, j \neq i\}$ unchanged

Our Algorithms: Key Ideas (I)

Dual step: Stochastic variance reduced coordinate ascent

- Batch gradient ascent for dual variables: For each $1 \le i \le n_X$,

$$w_i^{(k)} = \arg\min_{w_i} \Big\{ -\Big\langle \frac{1}{n_{Y_i}} \sum_{j=0}^{n_{Y_i}-1} f_{\theta^{(k-1)}}(x_i, y_{ij}), w_i \Big\rangle + \phi_i^*(w_i) + \frac{1}{2\alpha_w} \|w_i - w_i^{(k-1)}\|^2 \Big\}$$

  - Evaluating the full batch gradient is expensive
  - Updating each of the n_X variables is also expensive
- Key Idea 1: *Exploit decoupled dual maximization* over w_i's:
  - At each iteration k, randomly sample an index i and update w_i
  - Keep other $\{w_j, j \neq i\}$ unchanged
- Key Idea 2: Replace the full gradient with respect w_i, with *low variance stochastic gradient*, using the SVRG technique of [Johnson & Zhang, 2013]:

$$\delta_k^w = f_{\theta^{(k-1)}}(x_{i_k}, y_{i_k j_k}) - f_{\tilde{\theta}}(x_{i_k}, y_{i_k j_k}) + \overline{f}_{i_k}(\tilde{\theta})$$

Our Algorithms: Key Ideas (II)

Primal step: Stochastic variance reduced gradient descent

- Batch gradient descent update for primal variable:

$$\theta^{(k)} = \arg\min_{\theta} \left\{ \left\langle \sum_{i=0}^{n_X - 1} \sum_{j=0}^{n_{Y_i}-1} \frac{1}{n_X n_{Y_i}} f'_{\theta^{(k-1)}}(x_i, y_{ij}) w_i^{(k)}, \theta \right\rangle + \frac{1}{2\alpha_\theta} \|\theta - \theta^{(k-1)}\|^2 \right\}$$

  - Once again, computational cost can be very large
- As before, we can use the SVRG technique to replace the full gradient with a low variance stochastic gradient:

$$\delta_k^\theta = f'_{\theta^{(k-1)}}(x_{i'_k}, y_{i'_k j'_k}) \widetilde{w}_{i'_k} - f'_{\tilde{\theta}}(x_{i'_k}, y_{i'_k j'_k}) \widetilde{w}_{i'_k} + L'_\theta(\tilde{\theta}, \widetilde{w})$$

Our Algorithms: Key Ideas (III)

Low complexity stochastic variance reduced estimator

- Stochastic Variance Reduced Gradient (SVRG) for $h(\theta) = \sum_{i=0}^{n-1} h_i(\theta)$:

$$\delta_k = h_{i_k}(\theta) - h_{i_k}(\tilde{\theta}) + h(\tilde{\theta})$$

- *Faster the reference variables $\tilde{\theta}$ and $\widetilde{w}$ are updated, lower the variance of stochastic gradient, and faster the convergence*
  - But also requires more complexity

Our Algorithms: Key Ideas (III)

Low complexity stochastic variance reduced estimator

- Stochastic Variance Reduced Gradient (SVRG) for $h(\theta) = \sum_{i=0}^{n-1} h_i(\theta)$:

$$\delta_k = h_{i_k}(\theta) - h_{i_k}(\tilde{\theta}) + h(\tilde{\theta})$$

- *Faster the reference variables $\tilde{\theta}$ and $\widetilde{w}$ are updated, lower the variance of stochastic gradient, and faster the convergence*
  - But also requires more complexity
- Trick: "Free" full batch gradient update to obtain $L'_\theta(\tilde{\theta}, w^{(k)})$
  - Key Idea 1: Use the fact that a single w_i is updated in each iteration
  - Key Idea 2: Exploit the linearity of objective in dual variables

$$L'(\tilde{\theta}, w^{(k)}) = L'(\tilde{\theta}, w^{(k-1)}) + \frac{1}{n_X} \overline{f}'_{i_k}(\tilde{\theta})\big(w_{i_k}^{(k)} - w_{i_k}^{(k-1)}\big)$$

Our Algorithms: Key Ideas (III)

Low complexity stochastic variance reduced estimator

- Stochastic Variance Reduced Gradient (SVRG) for $h(\theta) = \sum_{i=0}^{n-1} h_i(\theta)$:

$$\delta_k = h_{i_k}(\theta) - h_{i_k}(\tilde{\theta}) + h(\tilde{\theta})$$

- *Faster the reference variables $\tilde{\theta}$ and $\widetilde{w}$ are updated, lower the variance of stochastic gradient, and faster the convergence*
  - But also requires more complexity
- Trick: "Free" full batch gradient update to obtain $L'_\theta(\tilde{\theta}, w^{(k)})$
  - Key Idea 1: Use the fact that a single w_i is updated in each iteration
  - Key Idea 2: Exploit the linearity of objective in dual variables

$$L'(\tilde{\theta}, w^{(k)}) = L'(\tilde{\theta}, w^{(k-1)}) + \frac{1}{n_X} \overline{f}'_{i_k}(\tilde{\theta})\big(w_{i_k}^{(k)} - w_{i_k}^{(k-1)}\big)$$

- Replace $L'_\theta(\tilde{\theta}, \widetilde{w})$ with $L'(\tilde{\theta}, w^{(k)})$ in naive SVRG gradient estimator

Our Algorithms: Key Ideas (IV)

Reducing the memory cost & SVRPDA - II

- Updating $L'(\tilde{\theta}, w^{(k)})$ at each iteration requires storing

$$\overline{f}_i'(\tilde{\theta}) = \sum_{j=0}^{n_{Y_i}-1} \frac{f_{\tilde{\theta}}'(x_i, y_{ij})}{n_{Y_i}}, \qquad 1 \leq i \leq n_X$$

  - Storage complexity can be very high

- Heuristic: Replace $\overline{f}_i'(\tilde{\theta})$ with sampled $f_{\tilde{\theta}}'(x_i, y_{ij})$ in update equation
- Results in low storage complexity, SVRPDA – II

Algorithm 1 SVRPDA-I

Inputs: data $\{(x_i, y_{ij}) : 0 \le i < n_X, 0 \le j < n_{Y_i}\}$; step-sizes α_θ and α_w; # inner iterations M.
Initialization: $\tilde{\theta}_0 \in \mathbb{R}^d$ and $\tilde{w}_0 \in \mathbb{R}^{\ell n_X}$.
for $s = 1, 2, \ldots$ **do**
 Set $\tilde{\theta} = \tilde{\theta}_{s-1}$, $\theta^{(0)} = \tilde{\theta}$, $\tilde{w} = \tilde{w}_{s-1}$, $w^{(0)} = \tilde{w}_{s-1}$, and compute the batch quantities (for each $0 \le i < n_X$):

$$U_0 = \sum_{i=0}^{n_X-1} \sum_{j=0}^{n_{Y_i}-1} \frac{f'_{\tilde{\theta}}(x_i, y_{ij}) w_i^{(0)}}{n_X n_{Y_i}}, \quad \overline{f}_i(\tilde{\theta}) \triangleq \sum_{j=0}^{n_{Y_i}-1} \frac{f_{\tilde{\theta}}(x_i, y_{ij})}{n_{Y_i}}, \quad \overline{f}'_i(\tilde{\theta}) = \sum_{j=0}^{n_{Y_i}-1} \frac{f'_{\tilde{\theta}}(x_i, y_{ij})}{n_{Y_i}}. \tag{11}$$

 for $k = 1$ **to** M **do**
  Randomly sample $i_k \in \{0, \ldots, n_X - 1\}$ and then $j_k \in \{0, \ldots, n_{Y_{i_k}} - 1\}$ at uniform.
  Compute the stochastic variance reduced gradient for dual update:

$$\delta_k^w = f_{\theta^{(k-1)}}(x_{i_k}, y_{i_k j_k}) - f_{\tilde{\theta}}(x_{i_k}, y_{i_k j_k}) + \overline{f}_{i_k}(\tilde{\theta}). \tag{12}$$

  Update the dual variables:

$$w_i^{(k)} = \begin{cases} \arg\min_{w_i} \Big[-\langle \delta_k^w, w_i \rangle + \phi_i^*(w_i) + \frac{1}{2\alpha_w} \| w_i - w_i^{(k-1)} \|^2 \Big] & \text{if } i = i_k \\ w_i^{(k-1)} & \text{if } i \neq i_k \end{cases}. \tag{13}$$

  Update U_k (primal batch gradient at $\tilde{\theta}$ and $w^{(k)}$) according to the following recursion:

$$U_k = U_{k-1} + \frac{1}{n_X} \overline{f}'_{i_k}(\tilde{\theta}) \big(w_{i_k}^{(k)} - w_{i_k}^{(k-1)} \big). \tag{14}$$

  Randomly sample $i'_k \in \{0, \ldots, n_X - 1\}$ and then $j'_k \in \{0, \ldots, n_{Y_{i'_k}} - 1\}$, independent of i_k and j_k, and compute the stochastic variance reduced gradient for primal update:

$$\delta_k^\theta = f'_{\theta^{(k-1)}}(x_{i'_k}, y_{i'_k j'_k}) w_{i'_k}^{(k)} - f'_{\tilde{\theta}}(x_{i'_k}, y_{i'_k j'_k}) w_{i'_k}^{(k)} + U_k. \tag{15}$$

  Update the primal variable:

$$\theta^{(k)} = \arg\min_{\theta} \Big[\langle \delta_k^\theta, \theta \rangle + g(\theta) + \frac{1}{2\alpha_\theta} \| \theta - \theta^{(k-1)} \|^2 \Big]. \tag{16}$$

 end for
 Option I: Set $\tilde{w}_s = w^{(M)}$ and $\tilde{\theta}_s = \theta^{(M)}$.
 Option II: Set $\tilde{w}_s = w^{(M)}$ and $\tilde{\theta}_s = \theta^{(t)}$ for randomly sampled $t \in \{0, \ldots, M-1\}$.
end for
Output: $\tilde{\theta}_s$ at the last outer-loop iteration.

Algorithm 1 SVRPDA-II

Inputs: data $\{(x_i, y_{ij}) : 0 \le i < n_X, 0 \le j < n_{Y_i}\}$; step-sizes α_θ and α_w; # inner iterations M.
Initialization: $\tilde{\theta}_0 \in \mathbb{R}^d$ and $\tilde{w}_0 \in \mathbb{R}^{\ell n_X}$.
for $s = 1, 2, \ldots$ **do**
  Set $\tilde{\theta} = \tilde{\theta}_{s-1}$, $\theta^{(0)} = \tilde{\theta}$, $w^{(0)} = \tilde{w}_{s-1}$, and compute the batch quantities (for each $0 \le i < n_X$):

$$U_0 = \sum_{i=0}^{n_X-1} \sum_{j=0}^{n_{Y_i}-1} \frac{f'_{\tilde{\theta}}(x_i, y_{ij}) w_i^{(0)}}{n_X n_{Y_i}}, \quad \overline{f}_i(\tilde{\theta}) \triangleq \sum_{j=0}^{n_{Y_i}-1} \frac{f_{\tilde{\theta}}(x_i, y_{ij})}{n_{Y_i}}. \tag{4}$$

  for $k = 1$ **to** M **do**
    Randomly sample $i_k \in \{0, \ldots, n_X - 1\}$ and then $j_k \in \{0, \ldots, n_{Y_{i_k}} - 1\}$ at uniform.
    Compute the stochastic variance reduced gradient for dual update:

$$\delta_k^w = f_{\theta^{(k-1)}}(x_{i_k}, y_{i_k j_k}) - f_{\tilde{\theta}}(x_{i_k}, y_{i_k j_k}) + \overline{f}_{i_k}(\tilde{\theta}). \tag{5}$$

    Update the dual variables:

$$w_i^{(k)} = \begin{cases} \arg\min_{w_i} \Big[-\langle \delta_k^w, w_i \rangle + \phi_i^*(w_i) + \frac{1}{2\alpha_w} \|w_i - w_i^{(k-1)}\|^2 \Big] & \text{if } i = i_k \\ w_i^{(k-1)} & \text{if } i \neq i_k \end{cases}. \tag{6}$$

    Update U_k according to the following recursion:

$$U_k = U_{k-1} + \frac{1}{n_X} f'_{\tilde{\theta}}(x_{i_k}, y_{i_k j''_k}) \big(w_{i_k}^{(k)} - w_{i_k}^{(k-1)} \big). \tag{7}$$

    Randomly sample $i'_k \in \{0, \ldots, n_X - 1\}$ and then $j'_k \in \{0, \ldots, n_{Y_{i'_k}} - 1\}$, independent of i_k and j_k, and compute the stochastic variance reduced gradient for primal update:

$$\delta_k^\theta = f'_{\theta^{(k-1)}}(x_{i'_k}, y_{i'_k j'_k}) w_{i'_k}^{(k)} - f'_{\tilde{\theta}}(x_{i'_k}, y_{i'_k j'_k}) w_{i'_k}^{(k)} + U_k. \tag{8}$$

    Update the primal variable:

$$\theta^{(k)} = \arg\min_{\theta} \Big[\langle \delta_k^\theta, \theta \rangle + g(\theta) + \frac{1}{2\alpha_\theta} \|\theta - \theta^{(k-1)}\|^2 \Big]. \tag{9}$$

  end for
  Option I: Set $\tilde{w}_s = w^{(k)}$ and $\tilde{\theta}_s = \theta^{(k)}$.
  Option II: Set $\tilde{w}_s = w^{(k)}$ and $\tilde{\theta}_s = \theta^{(t)}$ for randomly sampled $t \in \{0, \ldots, M-1\}$.
end for
Output: $\tilde{\theta}_s$ at the last outer-loop iteration.

Theory for Convergence

Assumptions

- $g(\theta)$ is μ-strongly convex in θ, and each ϕ_i is $1/\gamma$-smooth
- The merit functions $\phi_i(u)$ are Lipschitz with a uniform constant B_w
- $f_\theta(x_i, y_{ij})$ is B_θ-smooth in θ, and its gradients are uniformly bounded by a constant B_f
- For each given w in its domain, the function $L(\theta, w)$ is convex in θ

Main Result

After s outer-loops, to achieve error $\mathsf{E}\|\tilde{\theta}_s - \theta^*\|^2 < \epsilon$ using SVRPDA-I, total complexity required in terms of "number of oracle calls" is

$$O\big((n_X n_Y + n_X \kappa + n_X)\ln(1/\epsilon)\big)$$

Comparison with Existing Algorithms

- Our goal: $\min_\theta \frac{1}{n_X} \sum_{i=0}^{n_X-1} \phi_i\left(\frac{1}{n_{Y_i}} \sum_{j=0}^{n_{Y_i}-1} f_\theta(x_i, y_{ij})\right) + g(\theta)$

- Special case: $\min_\theta \frac{1}{n_X} \sum_{i=0}^{n_X-1} \phi_i\left(\frac{1}{n_Y} \sum_{j=0}^{n_Y-1} f_\theta(y_j)\right)$

Table: Total complexities of different stochastic composition optimization algorithms. For C-SAGA, $\alpha = 2/3$ with minibatch and $\alpha = 1$ when batch-size=1.

Methods	SVRPDA-I (Ours)	Comp-SVRG	C-SAGA	MSPBE-SVRG & MSPBE-SAGA	ASCVRG
Our problem	$(n_X n_Y + n_X \kappa) \ln \frac{1}{\epsilon}$				
Special	$(n_X + n_Y + n_X \kappa) \ln \frac{1}{\epsilon}$	$(n_X + n_Y + \kappa^3) \ln \frac{1}{\epsilon}$	$(n_X + n_Y + (n_X + n_Y)^\alpha \kappa) \ln \frac{1}{\epsilon}$		$(n_X + n_Y) \ln \frac{1}{\epsilon} + \frac{1}{\epsilon^3}$
$(n_X = 1)$	$(n_Y + \kappa) \ln \frac{1}{\epsilon}$	$(n_Y + \kappa^3) \ln \frac{1}{\epsilon}$	$(n_Y + n_Y^\alpha \kappa) \ln \frac{1}{\epsilon}$	$(n_Y + \kappa^2) \ln \frac{1}{\epsilon}$	$n_Y \ln \frac{1}{\epsilon} + \frac{1}{\epsilon^3}$

Storage Complexity of SVRPDA

Table: Storage complexity of SVRPDA-I and SVRPDA-II

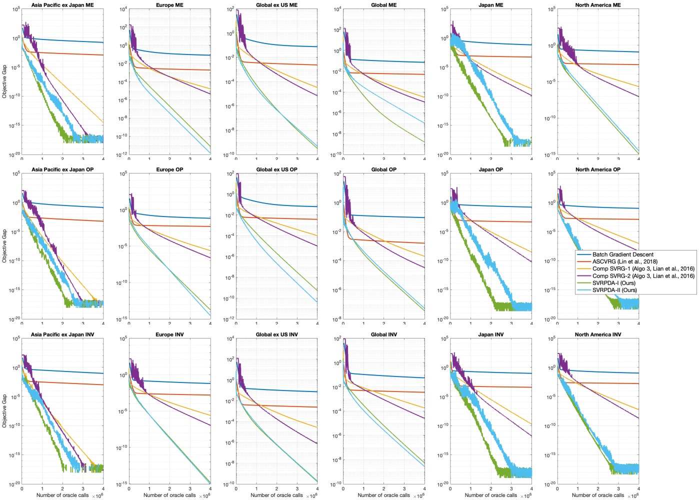
Methods	U_0	$\{\overline{f}_i\}$	$\{\overline{f}'_i\}$	$\theta^{(k)}$	$\tilde{\theta}$	$\{w_i^{(k)}\}$	δ_k^{θ}	δ_k^{w}	Total
SVRPDA-I	$O(d)$	$O(n_X \ell)$	$O(n_X d\ell)$	$O(d)$	$O(d)$	$O(n_X \ell)$	$O(d)$	$O(\ell)$	$O(n_X d\ell)$
SVRPDA-II	$O(d)$	$O(n_X \ell)$		$O(d)$	$O(d)$	$O(n_X \ell)$	$O(d)$	$O(\ell)$	$O(d+n_X \ell)$

Simulation Results: Risk Averse Learning

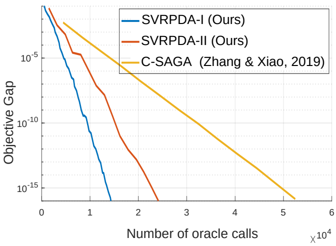
ME datasets

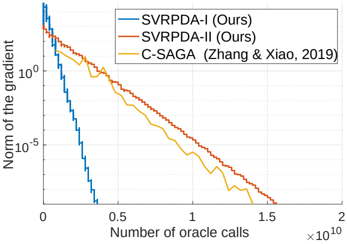
OP datasets

INV datasets



Simulation Results: MDP Policy Evaluation





Summary

- New SVRPDA algorithms are proposed to solve generic stochastic composition optimization algorithms
- Non-asymptotic bound for the error sequence was derived; Showed linear convergence of the algorithm
  - Complexity of SVRPDA was shown to be better than existing algorithms
- Experimental results showed that the algorithm outperforms all existing composition optimization algorithms
- Future work: Extensions to non-convex objectives, mini-batch SVRPDA, accelerated SVRPDA, etc.