Supplementary Materials for XOR-SGD: Provable Convex Stochastic Optimization for Decision-making under Uncertainty

Fan Ding[1] **Yexiang Xue**[1]

[1]Department of Computer Science, Purdue University, West Lafayette, Indiana, USA

A XOR-SAMPLING FOR THE WEIGHTED CASE

The text here provides a synopsis for the approach in Ermon et al. [2013]. We still encourage the readers to read the original text for a better explanation. Let $w(\theta), p(\theta)$ and Z as defined before, the high-level idea of XOR-Sampling is to first dicretize $w(\theta)$ to $w'(\theta)$ as in Definition 1, followed by embedding the weighted $w'(\theta)$ to the unweighted space Δ_w. Finally, XOR-sampling uses counting based on hashing and randomization to sample uniformly from Δ_w.

Definition 1. *Assume $w(\theta)$ has both upper and lower bound, namely, $M = \max_\theta w(\theta)$ and $m = \min_\theta w(\theta)$. Let $b \geq 1, \epsilon > 0, r = 2^b/(2^b - 1)$ and $l = \lceil \log_r(2^n/\epsilon) \rceil$. Partition the configurations into the following weight based disjoint buckets: $\mathcal{B}_i = \{\theta | w(\theta) \in (\frac{M}{r^{i+1}}, \frac{M}{r^i}]\}, i = 0, \ldots, l-1$ and $\mathcal{B}_l = \{\theta | w(\theta) \in (0, \frac{M}{r^l}]\}$. The discretized weight function $w' : \{0,1\}^n \to \mathbb{R}^+$ is defined as follows: $w'(\theta) = \frac{M}{r^{i+1}}$ if $\theta \in \mathcal{B}_i, i = 0, \ldots, l-1$ and $w'(\theta) = 0$ if $\theta \in \mathcal{B}_l$. This leads to the corresponding discretized probability distribution $p'(\theta) = w'(\theta)/Z'$ where Z' is the normalization constant of $w'(\theta)$.*

For the weighted case, the goal of XOR-sampling is to guarantee that the probability of sampling one θ is proportional to the unnormalized density (up to a multiplicative constant). By Definition 1, we obtain a distribution $p'(x)$ which satisfying $\frac{1}{\rho}p(x) \leq p'(x) \leq \rho p(x)$ where $\rho = \frac{r^2}{1-\epsilon}$. Then, XOR-sampling implements a horizontal slice technique to transform a weighted problem into an unweighted one. For the easiness of illustration, we denote $M' = \max_\theta w'(\theta)$ and m' as the smallest non-zero value of $w'(\theta)$. Then consider the simple case where $b = 1$ and $r = 2$, where we have $M' = 2^{l-1}m'$. Let $\delta = (\delta_0, \ldots, \delta_{l-2})^T \in \{0,1\}^{l-1}$ be a binary vector of length $l-1$, XOR-sampling samples (θ, δ) uniformly at random from the following set Δ_w using the unweighted version of sampling based on hashing and randomization:

$$\Delta_w = \{(\theta, \delta) : w'(\theta) \leq 2^{i+1}m' \Rightarrow \delta_i = 0\}. \quad (1)$$

If we sample (θ, δ) uniformly at random from Δ_w and then only return θ, it can be proved that the probability of sampling θ from $w'(\theta)$ is proportional to $m'2^{i-1}$ when $w(\theta)$ is sandwiched between $m'2^{i-1}$ and $m'2^i$. Therefore, this technique leads to the constant approximation guarantee of XOR-Sampling. The precise statement of the guarantee is in Theorem 2. For general case of b and r, please refer to Ermon et al. [2013].

Setting $\epsilon\eta_\phi$ to Zero In Definition 1 we can make b larger and ϵ smaller enough, then there will be a possibly large but finite value of l such that $\frac{M}{r^l}$ is smaller than m , which leads $\mathcal{B}_l$ to be empty and $\epsilon\eta_\phi$ to be zero.

B PROOFS

B.1 PROOF OF LEMMA 1

We define two functions $g_k^+ = \max\{g_k, \mathbf{0}\}$ and $g_k^- = \min\{g_k, \mathbf{0}\}$ where $\mathbf{0}$ is a vector of all 0 which has the same dimension as g_k. We have $g_k = g_k^+ + g_k^-$. We define both $\nabla f(x_k)^+$ and $\nabla f(x_k)^-$ in the similar way. Then Lemma 1 gives the new bounds of two terms assuming the constant bound on the gradient, which are essential to the proof of convergence rate. The proof of Lemma 1 is as follows:

Proof. (Lemma 1) Since we have the constant bound that

$$\frac{1}{c}\nabla f(x_k)^+ \leq \mathbb{E}[g_k^+] \leq c\nabla f(x_k)^+. \quad (2)$$

$$c\nabla f(x_k)^- \leq \mathbb{E}[g_k^-] \leq \frac{1}{c}\nabla f(x_k)^-. \quad (3)$$

and because of $g_k^+ \geq \mathbf{0}$ and $g_k^- \leq \mathbf{0}$ we can obtain

$$\begin{aligned}\frac{1}{c}||\mathbb{E}[g_k^+]||_2^2 &= \frac{1}{c}\langle \mathbb{E}[g_k^+], \mathbb{E}[g_k^+]\rangle \leq \langle \nabla f(x_k)^+, \mathbb{E}[g_k^+]\rangle \\ &\leq c\langle \mathbb{E}[g_k^+], \mathbb{E}[g_k^+]\rangle = c||\mathbb{E}[g_k^+]||_2^2. \\ \frac{1}{c}||\mathbb{E}[g_k^-]||_2^2 &= \frac{1}{c}\langle \mathbb{E}[g_k^-], \mathbb{E}[g_k^-]\rangle \leq \langle \nabla f(x_k)^-, \mathbb{E}[g_k^-]\rangle \\ &\leq c\langle \mathbb{E}[g_k^-], \mathbb{E}[g_k^-]\rangle = c||\mathbb{E}[g_k^-]||_2^2.\end{aligned}$$

which exactly means

$$\frac{1}{c}||\mathbb{E}[g_k]||_2^2 \leq \langle \nabla f(x_k), \mathbb{E}[g_k]\rangle \leq c||\mathbb{E}[g_k]||_2^2.$$

To prove the second inequality, we need to take advantage of the convexity of f. Denote $[x_k - x^*]^+ = \max\{x_k - x^*, \mathbf{0}\}$ and $[x_k - x^*]^- = \min\{x_k - x^*, \mathbf{0}\}$, we know $x_k - x^* = [x_k - x^*]^+ + [x_k - x^*]^-$. In addition, because f is convex, the index set of non-zero entries of $[x_k - x^*]^+$ and $\nabla f(x_k)^+$ is the same. The index set of non-zero entries of $[x_k - x^*]^-$ and $\nabla f(x_k)^-$ is also the same. In addition, because of Equation 2 and 3, the index set of non-zero entries of $\mathbb{E}[g_k^+]$ ($\mathbb{E}[g_k^-]$) is the same with $\nabla f(x_k)^+$ ($\nabla f(x_k)^-$). Combining these facts with Equations 2 and 3, we have

$$\begin{aligned}\frac{1}{c}\langle \mathbb{E}[g_k^+], [x_k - x^*]^+\rangle &\leq \langle \nabla f(x_k)^+, [x_k - x^*]^+\rangle \\ &\leq c\langle \mathbb{E}[g_k^+], [x_k - x^*]^+\rangle. \\ \frac{1}{c}\langle \mathbb{E}[g_k^-], [x_k - x^*]^-\rangle &\leq \langle \nabla f(x_k)^-, [x_k - x^*]^-\rangle \\ &\leq c\langle \mathbb{E}[g_k^-], [x_k - x^*]^-\rangle.\end{aligned}$$

Combining these two equations, we have

$$\frac{1}{c}\langle \mathbb{E}[g_k], x_k - x^*\rangle \leq \langle \nabla f(x_k), x_k - x^*\rangle \leq c\langle \mathbb{E}[g_k], x_k - x^*\rangle.$$

This completes the proof. □

B.2 PROOF OF THEOREM 4

Proof. (Theorem 4) Since we use N samples at each iteration, we have $\overline{g_k} = \frac{1}{N}\sum_{i=1}^N g_k^i$ and $\mathbb{E}[\overline{g_k}] = \mathbb{E}[g_k^i]$. In each iteration k we can adjust the parameters in XOR-Sampling to make the tail $\epsilon\eta_\phi$ zero, then for each sample g_k^i we can obtain from Theorem 2 that

$$\frac{1}{\rho\kappa}\mathbb{E}_\theta[\nabla f(x_k, \theta)]^+ \leq \mathbb{E}[g_k^{i+}] \leq \rho\kappa\mathbb{E}_\theta[\nabla f(x_k, \theta)]^+. \quad (4)$$

$$\rho\kappa\mathbb{E}_\theta[\nabla f(x_k, \theta)]^- \leq \mathbb{E}[g_k^{i-}] \leq \frac{1}{\rho\kappa}\mathbb{E}_\theta[\nabla f(x_k, \theta)]^-. \quad (5)$$

The variance of each sample g_k^i can also be bounded by

$$\begin{aligned}&Var(g_k^i) \\ &= \mathbb{E}_{\theta' \sim p'(\theta')}[||\nabla f(x_k, \theta')||_2^2] - ||\mathbb{E}_{\theta' \sim p'(\theta')}[\nabla f(x_k, \theta')]||_2^2, \\ &\leq \rho\kappa\mathbb{E}_{\theta \sim p(\theta)}[||\nabla f(x_k, \theta)||_2^2], \\ &= \rho\kappa(Var(\nabla f(x_k, \theta)) + ||\mathbb{E}_{\theta \sim p(\theta)}[\nabla f(x_k, \theta)]||_2^2), \\ &\leq \rho\kappa(\sigma^2 + \varepsilon^2).\end{aligned}$$

Denote $\overline{g_k}^+ = \max\{\overline{g_k}, \mathbf{0}\}$ and $\overline{g_k}^- = \min\{\overline{g_k}, \mathbf{0}\}$. Clearly, $g_k^{i+} \geq 0$ and $g_k^{i-} \leq 0$. Moreover, for a given dimension, either $g_k^{i+} = 0$ for that dimension or $g_k^{i-} = 0$. Evaluating $\overline{g_k}$ dimension by dimension, we can see that $\overline{g_k}^+ = \frac{1}{N}\sum_{i=1}^N g_k^{i+}$ and $\overline{g_k}^- = \frac{1}{N}\sum_{i=1}^N g_k^{i-}$. Combined with Equation 4 and 5, we know

$$\frac{1}{\rho\kappa}\mathbb{E}_\theta[\nabla f(x_k, \theta)]^+ \leq \mathbb{E}[\overline{g_k}^+] \leq \rho\kappa\mathbb{E}_\theta[\nabla f(x_k, \theta)]^+.$$

$$\rho\kappa\mathbb{E}_\theta[\nabla f(x_k, \theta)]^- \leq \mathbb{E}[\overline{g_k}^-] \leq \frac{1}{\rho\kappa}\mathbb{E}_\theta[\nabla f(x_k, \theta)]^-.$$

Because $\mathbb{E}[\overline{g_k}] = \mathbb{E}[g_k^i]$, we also have

$$Var(\overline{g_k}) = \frac{1}{N^2}Var(\sum_{i=1}^N g_k^i) = \frac{Var(g_k^i)}{N}.$$

Then the variance of $\overline{g_k}$ can be bounded as

$$Var(\overline{g_k}) \leq \frac{\rho\kappa(\sigma^2 + \varepsilon^2)}{N}.$$

Therefore, we can then apply Theorem 3 to get the result in equation 5.

$$\begin{aligned}&\mathbb{E}_{\overline{x_K}}[\mathbb{E}_\theta[f(\overline{x_K}, \theta)]] - \mathbb{E}_\theta[f(x^*, \theta)] \\ &\leq \frac{\rho\kappa||x_0 - x^*||_2^2}{2tK} + \frac{t\max_k\{Var(\overline{g_k})\}}{\rho\kappa}, \\ &\leq \frac{\rho\kappa||x_0 - x^*||_2^2}{2tK} + \frac{t(\sigma^2 + \varepsilon^2)}{N}.\end{aligned}$$

which can also be written as

$$\mathbb{E}_{\overline{x_K}}[\text{obj}] - OPT \leq \frac{\rho\kappa||x_0 - x^*||_2^2}{2tK} + \frac{t(\sigma^2 + \varepsilon^2)}{N}. \quad (6)$$

This completes the proof. □

C EXPERIMENTS

We evaluate our XOR-SGD algorithm on the inventory management Ziukov [2016], Shapiro and Philpott [2007] and the network design problems Sheldon et al. [2012], Wu et al. [2017, 2016]. For each setting of both applications, to produce a sample, Gibbs sampling first takes 100 steps to burn in, and then draws samples every 30 steps. We fix the iteration step of both BP and BPChain as 20, which is enough for BP to converge. We allow SGD with Gibbs sampling, BP and BPChain to draw more samples than XOR-SGD for a fair comparison. All experiments were conducted using single core architectures on Intel Xeon Gold 6126 2.60GHz machines with 96GB RAM and a wall-time limit of 10 hours. For both applications, we use MRF as probabilistic models for $Pr(\theta)$, which can be seen in the next section. For a fair comparison, once a solution x is generated by either algorithm, we use an exact weighted counter ACE Barton et al. [2016] to evaluate $\mathbb{E}_{\theta \sim Pr(\theta)}f(x, \theta)$ exactly. All objective values reported here are from ACE.

C.1 SETTINGS OF STOCHASTIC INVENTORY MANAGEMENT

Taking into account of the storage constraint, the original problem is equivalent to the following problem:

$$\min_{x\geq 0}\max_{\mu\geq 0}\mathbb{E}_{d\sim Pr(d)}[G(x,d)]+\mu(w^Tx-X). \quad (7)$$

For inventory management problem, we assume each d_i can take two different values, one corresponding to the high demand one corresponding to the low demand. Then, we introduce a new vector θ where $\theta_i = 1$ means d_i is the high value while $\theta_i = 0$ otherwise. In the experiment we range n from 10 to 100 increased by a step size of 10 and draw 10 instances for each setting. Under each setting, we draw every c_i uniformly from $(0, 5]$, h_i uniformly from $(0, 10]$, sample s_i uniformly drawn from $(0, 10]$ and let $b_i = c_i+s_i$. The two values of each d_i are also uniformly drawn from $(0, 10]$. We model $Pr(\theta)$ as a MRF with several cliques. The variables in each clique are highly correlated with each other. For a problem with n products, we draw the number of cliques uniformly from $[n, 2n]$. The domain size of each clique ϕ_α is chosen from the range of $[1, 6]$ at random. The potential function of a clique involving l variables is in the form of a table of size 2^l. The i-th entry of this table, denoted as v_i, is modeled as $v_i = v_{i1} + v_{i2}v_{i3}$, where v_{i1} is uniformly drawn from $(0, 1)$, v_3 uniformly from $(10, 1000)$ and binary variable v_{i2} uniformly randomly drawn from $\{0, 1\}$. Each storage requirement w_i is drawn from $(0, 10]$ uniformly at random. The largest storage limit X is set to be $5n$. We also evaluate our method given different percentages of the largest storage limit, which is shown in Figure 2 (middle). In the SGD algorithm, x is initialized with the absolute value of a Gaussian random variable from $\mathcal{N}(5, 3)$ to ensure it is non-negative.

In terms of the parameters in XOR-Sampling we fix $P = 100, b = 7, \epsilon = 0.01$ and the others the same as in Ermon et al. [2013] to guarantee $\rho\kappa = \sqrt{2}$. Learning rate t is 0.1 at first and divided by 10 after 50 iterations, then further divided by 10 after 100 iterations. η is 10 at first and divided by 10 after 50 iterations, then further divided by 10 after 100 iterations. The total number of both K and M are set to be 200. However, since we run each algorithm on one single core with a wall-time limit of 10 hours for a fair comparison, not all algorithms can complete all iterations. The plots are based on the best results found by each algorithm within the time limit.

C.2 SETTINGS OF STOCHASTIC NETWORK DESIGN

The task in equation 8 is equivalent to solving the following problem:

$$\min_{\Delta g\geq 0}\max_{\mu\geq 0}\ \mathbb{E}_{\theta\sim Pr(\theta)}[\overline{C}(g+\Delta g,\theta)]+\mu(\sum_{e\in E}c_e\Delta g_e-B). \quad (8)$$

Because of the convexity of $\overline{C}(g+\Delta g,\theta)$ and strong duality, both problems have the same optimal solution.

We test our algorithm on a real-world problem, the so-called Flood Preparation problem for the emergency medical services (EMS) on road networks Wu et al. [2016]. The problem setup, including the graph structure and the definition of $Pr(\theta)$, are the same as that in Wu et al. [2016]. The original network is unweighted, hence we set the initial conductance value for each edge as 1. c_e is initialized uniformly from the range $(0, 10)$. The largest budget size B is 1000. We evaluate our method varying the percentage allowed of the largest budget size, which is shown in Figure 3 (middle). In the experiment, each entry of Δg is initialized with the absolute value of a Gaussian random variable from $\mathcal{N}(0, 1)$. Total number of SGD iterations is 2000, while not all algorithms can complete all 2000 iterations within the time limit of 10 hours. The experimental results reported in the plots are based on the best solutions found by each algorithm within the time limit. Learning rate t is 1 at first and divided by 10 after 20 iterations, further by 10 after 100 iterations. Parameters in XOR-Sampling are set to be the same as in the inventory management problem.

The left figure in Figure 3 shows the percentage of savings between SGD with other sampling methods and XOR-SGD among all of the 4 different networks, while the middle and the right figures show the averaged commuting time with regard to different budget sizes and different number of samples, respectively. For the left and the middle figures, we let XOR-SGD take 100 samples in each iteration while SGD with other methods take 10,000. We can see from the left figure that objective optimized by XOR-SGD is at least 5% better than that optimized by other methods for all the 4 different networks. In addition, from the middle and the right figures we know that with the increase of either budget size or the number of samples, our method can find consistently better solutions than the compared methods. In particular, from the right figure we can see even 40 samples in each iteration are enough for XOR-SGD to compete with the result from Gibbs with 20,000 samples. Meanwhile, XOR-SGD also runs faster than the compared method under this situation. In this experiment, XOR-SGD with 40 samples take 1 minutes 40 seconds per SGD iteration, while SGD with 20,000 Gibbs samples need 2.5 minutes per iteration. Since sampling time of both BP and BPChain is no shorter than Gibbs Sampling, we thus conclude that XOR-SGD

outperforms other methods both in efficiency and in the quality of solutions found.

References

John P Barton, Eleonora De Leonardis, Alice Coucke, and Simona Cocco. Ace: adaptive cluster expansion for maximum entropy graphical model inference. *Bioinformatics*, 32(20):3089–3097, 2016.

Stefano Ermon, Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Embed and project: Discrete sampling with universal hashing. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Alexander Shapiro and Andy Philpott. A tutorial on stochastic programming. 2007.

Daniel Sheldon, Bistra Dilkina, Adam N Elmachtoub, Ryan Finseth, Ashish Sabharwal, Jon Conrad, Carla P Gomes, David Shmoys, William Allen, Ole Amundsen, et al. Maximizing the spread of cascades using network design. *arXiv preprint arXiv:1203.3514*, 2012.

Xiaojian Wu, Daniel R Sheldon, and Shlomo Zilberstein. Optimizing resilience in large scale networks. In *Proceedings of the 30th Conference of AAAI*, 2016.

Xiaojian Wu, Yexiang Xue, Bart Selman, and Carla P. Gomes. Xor-sampling for network design with correlated stochastic events. In *Proceedings of the 26th IJCAI*, pages 4640–4647, 2017.

Serhii Ziukov. A literature review on models of inventory management under uncertainty. 2016.