Variance Reduction for Matrix Games

Yair Carmon, Yujia Jin, Aaron Sidford and Kevin Tian
Stanford University
{yairc,yujiajin,sidford,kjtian}@stanford.edu

Abstract

We present a suite of randomized primal-dual algorithms which solve the problem $\min_x \max_y y^\top A x$ to additive error ϵ, encompassing many fundamental problems such as matrix games, linear programming, perceptron / SVM, minimum enclosing ball and linear regression. For matrix A with larger dimension n and $\text{nnz}(A)$ nonzero entries, we provide algorithms that run in time $\text{nnz}(A) + \sqrt{\text{nnz}(A)n}L/\epsilon$, where L is a domain-dependent norm of A. For matrix games and SVM, this is a factor $\sqrt{\text{nnz}(A)/n}$ improvement over the best known deterministic methods, and (for ϵ sufficiently small) a factor $\sqrt{n/\text{nnz}(A)}/\epsilon$ improvement over previously known stochastic methods. Furthermore, we show how to exploit (numerical) sparsity of A and develop variance-reduced coordinate methods that can improve running times by up to an additional $\sqrt{n}$ factor. Our development consists of (i) a variance reduction framework for general convex-concave problems using Nemirovski's "conceptual prox method," (ii) low-variance gradient estimators based on "sampling from the difference" between the current iterate and a reference point, and (iii) data structures supporting efficient gradient steps using these estimators.

1 Background

Minimax problems—or games—of the form $\min_x \max_y f(x, y)$ are ubiquitous in economics, statistics, optimization and machine learning. In recent years, minimax formulations for neural network training rose to prominence [14, 20], leading to intense interest in algorithms for solving large scale minimax games [10, 13, 18, 9, 16, 21]. However, the algorithmic toolbox for minimax optimization is not as complete as the one for minimization. Variance reduction, a technique for improving stochastic gradient estimators by introducing control variates, stands as a case in point. A multitude of variance reduction schemes exist for finite-sum minimization [17, 26, 1, 4, 11], and their impact on complexity is well-understood [32]. In contrast, only a few works apply variance reduction to finite-sum minimax problems [3, 30, 6, 23], and the potential gains from variance reduction are not well-understood.

We take a step towards closing this gap by designing variance-reduced minimax game solvers that offer acceleration similar to optimal variance reduction methods for finite-sum minimization. To achieve this, we focus on the fundamental class of bilinear minimax games,

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} y^\top A x + b^\top y + c^\top y, \text{ where } A \in \mathbb{R}^{m \times n}; \tag{1}$$

to simplify the exposition, we set b and c to zero throughout. We study the complexity of finding an ϵ-approximate saddle point (Nash equilibrium), namely x, y with $\max_{v \in \mathcal{Y}} v^\top A x - \min_{u \in \mathcal{X}} y^\top A u \le \epsilon$.

Section 2 for this extended abstract describes our NeurIPS 2019 paper [5]. Section 3 describes additional results currently under submission.

Abstract for: Smooth Games Optimization and Machine Learning Workshop (NeurIPS 2019), Vancouver, Canada.

When $\mathcal{X}, \mathcal{Y}$ are both probability simplices (referred to as an ℓ_1-ℓ_1 game hereinafter), the problem corresponds to finding an approximate (mixed) equilbrium in a matrix game, a central object in game theory and economics. Matrix games are also fundamental to algorithm design due in part to their equivalence to linear programming [8]. When $\mathcal{X}$ is a simplex and $\mathcal{Y}$ is an ℓ_2 ball (an ℓ_2-ℓ_1 game), the corresponding problem finds an approximate maximum-margin linear classifier (hard-margin SVM), a fundamental task in machine learning and statistics [22]. The problem also encompasses the well-known geometric problems of minimum enclosing ball and maximum inscribing ball [2]. Finally, when $\mathcal{X}, \mathcal{Y}$ are both unit balls in ℓ_2 (an ℓ_2-ℓ_2 game), the corresponding problem includes standard linear regression as a special case.

The main contribution of our work is to give improved algorithms for minimax problems in these domains. In Section 2 we describe our basic variance reduction and gradient estimation approach. In Section 3 we extend our approach to exploit (numerical) sparsity in the matrix A.

2 Our approach

We now describe the main ingredients in the design and analysis of our variance reduction algorithm. In Appendix A we provide complete pseudo-code for the ℓ_1-ℓ_1 instantiation this algorithm.

2.1 General variance reduction framework

Our starting point is Nemirovski's "conceptual prox-method" [24] for solving $\min_x \max_y f(x, y)$, where $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is convex in x and concave in y. The method solves a sequence of subproblems parameterized by $\alpha > 0$, each of the form

$$\text{find } x, y \text{ s.t. } \forall x', y' \ \langle \nabla_x f(x, y), x - x' \rangle - \langle \nabla_y f(x, y), y - y' \rangle \leq \alpha V_{x_0}(x') + \alpha V_{y_0}(y') \tag{2}$$

for some $w_0 = (x_0, y_0) \in \mathcal{X} \times \mathcal{Y}$, where V is a suitable Bregman divergence: squared Euclidean distance for ℓ_2 and KL divergence for ℓ_1. Combining each subproblem solution with an extragradient step, the conceptual prox method solves the original problem to ϵ accuracy by solving $\widetilde{O}(\alpha/\epsilon)$ subproblems.

When $g(x, y) \coloneqq (\nabla_x f(x, y), -\nabla_y f(x, y))$ is L-Lipschitz, a single (exact) gradient step suffices to solve (2) with $\alpha = L$; this gives rise to the well-known mirror-prox method that converges in $\widetilde{O}(L/\epsilon)$ iterations [24]. We take a different route, and solve the problem (2) with $\alpha \ll L$ using a stochastic gradient method. In particular, we prove that if a gradient estimator admits the "variance" bound

$$\mathbb{E} \left\| \tilde{g}(x, y) - \nabla f(x_0, y_0) \right\|_*^2 \leq L^2 \left(\|x - x_0\|^2 + \|y - y_0\|^2 \right) \text{ for some } L > 0, \tag{3}$$

then $O(L^2/\alpha^2)$ regularized stochastic mirror descent steps solve (2) in a suitable probabilistic sense. Writing $w = (x, y)$ and $V_{w'}(w) \coloneqq V_{x'}(x) + V_{y'}(y)$, these steps take the form

$$w_t \leftarrow \operatorname*{arg\,min}_{w \in \mathcal{Z}} \left\{ \langle \tilde{g}(w_{t-1}), w \rangle + \frac{1}{\eta} V_{w_{t-1}}(w) + \frac{\alpha}{2} V_{w_0}(w) \right\}, \tag{4}$$

where $\eta = \alpha / L^2$ is a step-size parameter. To utilize this framework, we then design computationally efficient gradient estimators that satisfy (3) and then choose α to balance the cost of $O(L^2/\alpha^2)$ stochastic gradient estimates with the cost of a single exact gradient computation.

2.2 Variance-reduced row/column gradient estimators for bilinear games

We now focus on bilinear problems, where $f(x, y) = y^\top A x$ and the gradient mapping is $g(x, y) = (A^\top y, -Ax)$. We wish to estimate the gradient at (x, y), having already computed the exact gradient at the reference point (x_0, y_0). We consider estimators that access only a single random row and column of A, and are therefore cheap to compute. An estimator $\tilde{g}$ is defined by two distributions $p \in \Delta^m$, $q \in \Delta^n$, and has the following form: independently draw $i \sim p$ and $j \sim q$, and set

$$\tilde{g}(x, y) \coloneqq \left(A^\top y_0 + A_{i:} \frac{y_i - [y_0]_i}{p_i}, -Ax_0 - A_{:j} \frac{x_j - [x_0]_j}{q_j} \right), \tag{5}$$

where $A_{i:}$ and $A_{:j}$ respectively denote the ith and jth columns of $A^\top$ and A.

Clearly, estimators of the form (5) are unbiased for any p, q. Their variance, however, depends crucially on the domain geometry and choices of p and q. Variance reduction schemes typically use p and q that are either uniform or proportional to the entries of A [17, 33, 1, 3]. However, in our setting these choices do not produce the required bound (3). Clarkson et al. [7] propose stochastic gradient estimators that sample rows and columns based on the current iterates. We extend this approach to reduced-variance gradient estimators by "sampling from difference" between the current iterate and the reference point. For variables x in an the simplex (ℓ_1 setup), we choose

$$q_j = \frac{|x_j - [x_0]_j|}{\|x - x_0\|_1}. \tag{6}$$

Similarly, for variables x in a Euclidean ball (ℓ_2 setup), we choose

$$q_j = \frac{(x_j - [x_0]_j)^2}{\|x - x_0\|_2^2}. \tag{7}$$

We show that these distributions yield the required variance bound (3) with values of $L = L_{\text{row-col}}$ given by

$$L_{\text{row-col}} = \begin{cases} \max_{ij} |A_{ij}| & \ell_1\text{-}\ell_1 \text{ games} \\ \max_i \|A_{i:}\|_2 & \ell_2\text{-}\ell_1 \text{ games} \\ \|A\|_{\text{F}} & \ell_2\text{-}\ell_2 \text{ games.} \end{cases} \tag{8}$$

For ℓ_1-ℓ_1 and ℓ_2-ℓ_1 games, $L_{\text{row-col}} = L_{\text{exact}}$, the Lipschitz constant of g. Consequently, even an exact gradient computation would not yield a tighter constant in (3) and in that sense $L_{\text{row-col}}$ is optimal in these settings. For ℓ_2-ℓ_1 games, obtaining the optimal constant requires using a local norms argument as well as gradient clipping similarly to the proposal of [7]. For ℓ_2-ℓ_2 games, $L_{\text{exact}} = \|A\|_{\text{op}}$ and consequently there is a gap between the constant we obtain and that obtainable by exact gradients. This gap is to be expected, as it appears in essentially all stochastic methods for linear regression [31, 17, 27, 12, 19, 28, 26, 1].

2.3 Method complexity

An exact gradient computation takes $O(\text{nnz}(A))$ time (matrix-vector products with A and $A^\top$), and with the gradient estimator (5) each stochastic mirror descent step of the form (4) takes $O(n + m)$ time. To achieve accuracy ϵ our algorithm performs $\widetilde{O}(\alpha/\epsilon)$ outer loops, each one taking $O(\text{nnz}(A))$ time for computing two exact gradients (one for variance reduction and one for an extragradient step), plus an additional $O((m + n)L_{\text{row-col}}^2/\alpha^2)$ time for the inner mirror descent iterations, with $L_{\text{row-col}}$ as in (8). The total runtime is therefore

$$\widetilde{O}\left(\left(\text{nnz}(A) + \frac{(m+n)L_{\text{row-col}}^2}{\alpha^2}\right)\frac{\alpha}{\epsilon}\right).$$

By setting α optimally to be $\max\{\epsilon, L_{\text{row-col}}\sqrt{(m+n)/\text{nnz}(A)}\}$, we obtain the runtime

$$\widetilde{O}\left(\text{nnz}(A) + \sqrt{\text{nnz}(A)\cdot(m+n)}\cdot\frac{L_{\text{row-col}}}{\epsilon}\right). \tag{9}$$

In comparison, accelerated linear-time methods run in time $\widetilde{O}(\text{nnz}(A)L_{\text{exact}}/\epsilon)$ [24, 25]; in ℓ_1-ℓ_1 and ℓ_2-ℓ_1 games we have $L_{\text{row-col}} = L_{\text{exact}}$ and the guarantee (9) is at least as good. In the square dense case ($\text{nnz}(A) \approx n^2 = m^2$) it is a factor $\sqrt{n}$ improvement. Optimal variance-reduced finite-sum minimization methods improve the fast gradient method by the same factor [33, 1]. Moreover, our result improves the sublinear $\widetilde{O}\left((n+m)L_{\text{row-col}}^2/\epsilon^2\right)$ runtime of [15, 7] when $\epsilon/L_{\text{row-col}} \leq \sqrt{(m+n)/\text{nnz}(A)}$, i.e. when $(n+m)L_{\text{row-col}}^2\epsilon^{-2} = \Omega(\text{nnz}(A))$ is not truly sublinear. Balamurugan and Bach [3] develop variance-reduction schemes that apply in the Euclidean (ℓ_2-ℓ_2) case only; our method improves their best runtime guarantee by a factor of $\log \epsilon^{-1}$.

3 Improved runtimes for sparse instances

3.1 Row and column sparsity

Suppose the matrix A has at most $\text{rcs}(A)$ nonzero elements in every row and column. We develop gradient estimators of the form (5) that, together with appropriate data structures, allow us to

implement each regularized stochastic mirror descent step (4) in amortized time $\widetilde{O}(\text{rcs}(A))$. This improves the runtime guarantee (9) to

$$\widetilde{O}\left(\text{nnz}(A) + \sqrt{\text{nnz}(A) \cdot \text{rcs}(A)} \cdot \frac{L_{\text{row-col}}}{\epsilon}\right). \tag{10}$$

To achieve this, we solve two technical challenges.

Maintaining the iterates. Even though rows and columns of A are sparse, the stochastic gradient estimator (5) and the regularization term in (4) add constant, dense component to the updates. Therefore, computing each step in time proportional to $\text{rcs}(A)$ requires a nontrivial data structure, particularly for the ℓ_1 domain where projecting to the simplex requires maintaining a sum of exponentials. We design a data structure that *efficiently maintains a Taylor expansion* of these exponentials and thus approximates them to high accuracy.

Maintaining the sampling distribution. To exploit sparsity, we require a data structure that lets us efficiently generate row/column samples from the dynamically changing distributions (6) and (7) (in time $\widetilde{O}(\text{rcs}(A))$ per sample). This too is particularly challenging in the ℓ_1 domain: it is unclear how to efficiently update a sample generator for the "difference" distribution (6), due to the absolute value of the difference and the fact that all entries of x change at every step. We sidestep this issue by *"sampling from the sum"* instead, using

$$q_j = \tfrac{1}{3}x_j + \tfrac{2}{3}[x_0]_j.$$

Sampling from the sum is much simpler; it is a mixture of the current iterate and the reference point, and we can maintain efficient sample generators for both. To show this sampling strategy is valid, we prove that it satisfies a version of the variance bound (3) where the squared ℓ_1 norm in the RHS is replaced with KL divergence, which suffices for the convergence analysis.

3.2 Coordinate methods

In some cases, A is *numerically sparse*: with few large elements and many small (but nonzero) elements. In these cases $\text{rcs}(A) = \Theta(n+m)$ and the developments in the previous section provide little benefit, and yet we intuitively expect numerical sparsity to allow faster runtimes. To achieve this, we introducing coordinate-wise gradient estimates, whose stochastic part is 1-sparse. For any two distributions $p, q \in \Delta^{m \times n}$ over the entries of A, we draw $i^{\mathsf{x}}, j^{\mathsf{x}} \sim p$ and $i^{\mathsf{y}}, j^{\mathsf{y}} \sim q$ and estimate the gradient as

$$\tilde{g}(x,y) \coloneqq \left(A^\top y_0 + e_{j^{\mathsf{x}}} A_{i^{\mathsf{x}} j^{\mathsf{x}}} \frac{y_{i^{\mathsf{x}}} - [y_0]_{i^{\mathsf{x}}}}{p_{i^{\mathsf{x}} j^{\mathsf{x}}}}, -Ax_0 - e_{i^{\mathsf{y}}} A_{i^{\mathsf{y}} j^{\mathsf{y}}} \frac{x_{j^{\mathsf{y}}} - [x_0]_{j^{\mathsf{y}}}}{q_{i^{\mathsf{y}} j^{\mathsf{y}}}}\right), \tag{11}$$

where e_j denotes the jth standard basis vector.

In Appendix B we list distributions p, q that satisfy the variance bound (3) (in suitable local norms) with $L = L_{\text{coord}}$ given by

$$L_{\text{coord}} = \begin{cases} \max\left\{\max_i \|A_{i:}\|_2, \max_j \|A_{:j}\|_2\right\} & \ell_1\text{-}\ell_1 \text{ games} \\ \max\left\{\max_i \|A_{i:}\|_1, \|A\|_{\text{F}}\right\} & \ell_2\text{-}\ell_1 \text{ games} \\ \max\left\{\sqrt{\sum_i \|A_{i:}\|_1^2}, \sqrt{\sum_j \|A_{:j}\|_1^2}\right\} & \ell_2\text{-}\ell_2 \text{ games.} \end{cases} \tag{12}$$

Furthermore, for these distributions we may use our sparsity-supporting data structures to implement the stochastic mirror steps (4) with the coordinate estimator (11) in amortized time $\tilde{O}(1)$ per step, obtaining total running time

$$\widetilde{O}\left(\text{nnz}(A) + \sqrt{\text{nnz}(A)} \cdot \frac{L_{\text{coord}}}{\epsilon}\right). \tag{13}$$

For ℓ_1-ℓ_1 and ℓ_2-ℓ_2 games we have $L_{\text{row-col}} \le L_{\text{coord}} \le \sqrt{\text{rcs}(A)} L_{\text{row-col}}$ so the runtime (13) is never more than (10) by logarithmic factors. For ℓ_2-ℓ_1 games we have $L_{\text{row-col}} \le L_{\text{coord}} \le \sqrt{n+m} L_{\text{row-col}}$, so (13) might be worse than (10) but never much worse than (9). In all cases, for numerically sparse instances with $L_{\text{coord}} \approx L_{\text{row-col}}$, we obtain a speedup of $\sqrt{n+m}$.

References

[1] Z. Allen-Zhu. Katyusha: the first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1200–1205, 2017.

[2] Z. Allen-Zhu, Z. Liao, and Y. Yuan. Optimization algorithms for faster computational geometry. In *43rd International Colloquium on Automata, Languages, and Programming*, pages 53:1–53:6, 2016.

[3] P. Balamurugan and F. R. Bach. Stochastic variance reduction methods for saddle-point problems. In *Advances in Neural Information Processing Systems*, 2016.

[4] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

[5] Y. Carmon, Y. Jin, A. Sidford, and K. Tian. Variance reduction for matrix games. In *Advances in Neural Information Processing Systems*, 2019.

[6] T. Chavdarova, G. Gidel, F. Fleuret, and S. Lacoste-Julien. Reducing noise in GAN training with variance reduced extragradient. In *Advances in Neural Information Processing Systems*, 2019.

[7] K. L. Clarkson, E. Hazan, and D. P. Woodruff. Sublinear optimization for machine learning. In *51th Annual IEEE Symposium on Foundations of Computer Science*, pages 449–457, 2010.

[8] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1953.

[9] C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng. Training GANs with optimism. In *International Conference on Learning Representations*, 2019.

[10] Y. Drori, S. Sabach, and M. Teboulle. A simple algorithm for a class of nonsmooth convex-concave saddle-point problems. *Operations Research Letters*, 43(2):209–214, 2015.

[11] C. Fang, C. J. Li, Z. Lin, and T. Zhang. SPIDER: near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, 2018.

[12] R. Frostig, R. Ge, S. Kakade, and A. Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2540–2548, 2015.

[13] G. Gidel, T. Jebara, and S. Lacoste-Julien. Frank-Wolfe algorithms for saddle point problems. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

[14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

[15] M. D. Grigoriadis and L. G. Khachiyan. A sublinear-time randomized approximation algorithm for matrix games. *Operation Research Letters*, 18(2):53–58, 1995.

[16] C. Jin, P. Netrapalli, and M. I. Jordan. Minmax optimization: Stable limit points of gradient descent ascent are locally optimal. *arXiv preprint arXiv:1902.00618*, 2019.

[17] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 2013.

[18] O. Kolossoski and R. D. Monteiro. An accelerated non-Euclidean hybrid proximal extragradient-type algorithm for convex-concave saddle-point problems. *Optimization Methods and Software*, 32(6):1244–1272, 2017.

[19] H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, 2015.

[20] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[21] P. Mertikopoulos, H. Zenati, B. Lecouat, C.-S. Foo, V. Chandrasekhar, and G. Piliouras. Mirror descent in saddle-point problems: Going the extra (gradient) mile. In *International Conference on Learning Representations*, 2019.

[22] M. Minsky and S. Papert. *Perceptrons—an introduction to computational geometry*. MIT Press, 1987.

[23] K. Mishchenko, D. Kovalev, E. Shulgin, P. Richtárik, and Y. Malitsky. Revisiting stochastic extragradient. *arXiv preprint arXiv:1905.11373*, 2019.

[24] A. Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[25] Y. Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programing*, 109(2-3):319–344, 2007.

[26] M. W. Schmidt, N. L. Roux, and F. R. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programing*, 162(1-2):83–112, 2017.

[27] S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss. *Journal of Machine Learning Research*, 14(1):567–599, 2013.

[28] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programing*, 155(1-2):105–145, 2016.

[29] S. Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[30] Z. Shi, X. Zhang, and Y. Yu. Bregman divergence for stochastic variance reduction: Saddle-point and adversarial prediction. In *Advances in Neural Information Processing Systems*, 2017.

[31] T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262, 2009.

[32] B. E. Woodworth and N. Srebro. Tight complexity bounds for optimizing composite objectives. In *Advances in Neural Information Processing Systems*, 2016.

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

A Pseudo-code for ℓ_1-ℓ_1 games

Algorithm 1: Variance reduction for ℓ_1-ℓ_1 games

Input: Matrix $A \in \mathbb{R}^{m \times n}$ with ith row $A_{i:}$ and jth column $A_{:j}$, target accuracy ϵ
Output: A point with expected duality gap below ϵ

$L \leftarrow \max_{ij} |A_{ij}|$, $\alpha \leftarrow L\sqrt{\frac{n+m}{\mathrm{nnz}(A)}}$, $K \leftarrow \left\lceil \frac{\log(nm)\alpha}{\epsilon} \right\rceil$, $\eta \leftarrow \frac{\alpha}{10L^2}$, $T \leftarrow \left\lceil \frac{4}{\eta\alpha} \right\rceil$, $z_0 \leftarrow (\frac{1}{n}\mathbf{1}_n, \frac{1}{m}\mathbf{1}_m)$

for $k = 1, \ldots, K$ **do**

  ▷ *Relaxed oracle query:*

  $(x_0, y_0) \leftarrow (z^{\mathsf{x}}_{k-1}, z^{\mathsf{y}}_{k-1})$, $(g^{\mathsf{x}}_0, g^{\mathsf{y}}_0) \leftarrow (A^\top y_0, -Ax_0)$

  for $t = 1, \ldots, T$ **do**

    ▷ *Gradient estimation:*

    Sample $i \sim p$ where $p_i = \dfrac{|[y_{t-1}]_i - [y_0]_i|}{\|y_{t-1} - y_0\|_1}$, sample $j \sim q$ where $q_j = \dfrac{|[x_{t-1}]_j - [x_0]_j|}{\|x_{t-1} - x_0\|_1}$

    Set $\tilde{g}_{t-1} = g_0 + \left(A_{i:} \dfrac{[y_{t-1}]_i - [y_0]_i}{p_i}, -A_{:j} \dfrac{[x_{t-1}]_j - [x_0]_j}{q_j} \right)$

    ▷ *Mirror descent step:*

    $x_t \leftarrow \Pi_{\mathcal{X}} \left(\dfrac{1}{1 + \eta\alpha/2} \left(\log x_{t-1} + \dfrac{\eta\alpha}{2} \log x_0 - \eta \tilde{g}^{\mathsf{x}}_{t-1} \right) \right)$ ▷ $\Pi_{\mathcal{X}}(v) = \frac{e^v}{\|e^v\|_1}$

    $y_t \leftarrow \Pi_{\mathcal{Y}} \left(\dfrac{1}{1 + \eta\alpha/2} \left(\log y_{t-1} + \dfrac{\eta\alpha}{2} \log y_0 - \eta \tilde{g}^{\mathsf{y}}_{t-1} \right) \right)$ ▷ $\Pi_{\mathcal{Y}}(v) = \frac{e^v}{\|e^v\|_1}$

  $z_{k-1/2} \leftarrow \dfrac{1}{T} \sum_{t=1}^{T} (x_t, y_t)$

  ▷ *Extragradient step:*

  $z^{\mathsf{x}}_k \leftarrow \Pi_{\mathcal{X}} \left(\log z^{\mathsf{x}}_{k-1} - \frac{1}{\alpha} A^\top z^{\mathsf{y}}_{k-1/2} \right)$

  $z^{\mathsf{y}}_k \leftarrow \Pi_{\mathcal{Y}} \left(\log z^{\mathsf{y}}_{k-1} + \frac{1}{\alpha} A z^{\mathsf{x}}_{k-1/2} \right)$

return $\dfrac{1}{K} \sum_{k=1}^{K} z_{k-1/2}$

B More information on coordinate methods

In Table 1 we summarize the sampling distributions we use for the coordinate gradient estimator (11) in the various setups we consider. For ease of reference, we also state the constant variance bounds constant L_{coord} we obtain, using the notation $a \vee b \coloneqq \max\{a, b\}$. The running time in each case is $\widetilde{O}\left(\text{nnz}(A) + \sqrt{\text{nnz}(A)} \cdot \frac{L_{\text{coord}}}{\epsilon}\right)$.

Table 1: Sampling distributions and variance bound constant for variance reduced coordinate methods.

Setup	p_{ij}	q_{ij}	L_{coord}
ℓ_1-ℓ_1	$\left(\frac{1}{3}y_i + \frac{2}{3}[y_0]_i\right) \frac{A_{ij}^2+\alpha^2}{\|A_{i:}\|_2^2+n\alpha^2}$	$\left(\frac{1}{3}x_j + \frac{2}{3}[x_0]_j\right) \frac{A_{ij}^2+\alpha^2}{\|A_{:j}\|_2^2+m\alpha^2}$	$\max_i \|A_{i:}\|_2 \vee \max_j \|A_{:j}\|_2$
ℓ_2-ℓ_1	$\left(\frac{1}{3}y_i + \frac{2}{3}[y_0]_i\right) \frac{\|A_{ij}\|}{\|A_{i:}\|_1}$	$\frac{A_{ij}^2+\alpha^2(x_j-[x_0]_j)^2}{\|A\|_{\text{F}}^2+\alpha^2\|x-x_0\|_2^2}$	$\max_i \|A_{i:}\|_1 \vee \|A\|_{\text{F}}$
ℓ_2-ℓ_2	$\frac{\|A_{i:}\|_1^2}{\sum_k \|A_{k:}\|_1^2} \cdot \frac{\|A_{ij}\|}{\|A_{i:}\|_1}$	$\frac{\|A_{:j}\|_1^2}{\sum_k \|A_{:k}\|_1^2} \cdot \frac{\|A_{ij}\|}{\|A_{:j}\|_1}$	$\sqrt{\sum_i \|A_{i:}\|_1^2} \vee \sqrt{\sum_j \|A_{:j}\|_1^2}$

We make the following remarks:

1. The performance bounds for our proposed coordinate methods are novel even without variance reduction. In particular, taking $\alpha = \epsilon$ produces methods that run in time $\widetilde{O}\left(\text{nnz}(A) + L_{\text{coord}}^2/\epsilon^2\right)$. In a number of regimes, this runtime improves on the best known fully stochastic methods of Clarkson et al. [7].
2. For ℓ_1 domains, obtaining the current value of L_{coord} relies crucially on bounding the estimation error in a local norm (depending on the current coordinate) [cf. 29, Section 2.8]. To apply local norm bounds in mirror descent analysis, the quantity $\eta \|\tilde{g}\|_\infty$ must be of the order of unity. The role of the terms proportional to α in Table 1 is to guarantee this.
3. The distributions described in Table 1 have marginals that are easy to sample from. For example, in the ℓ_2-ℓ_1 setup, to sample from p we first sample $i \sim \frac{1}{3}y + \frac{2}{3}y_0$ and then sample j with probability proportional to the magnitude of the elements in $A_{i:}$.
4. Many other efficient sampling schemes satisfy the same variance bounds. For example, in the ℓ_2-ℓ_2 setup we could also use $p_{ij} = \frac{(y_i-[y_0]_i)^2}{\|y-y_0\|_2^2} \cdot \frac{|A_{ij}|}{\|A_{:j}\|_1}$.
5. In every setup, it is possible to write in closed form the tightest possible variance bound (3) attainable via estimators of the form (11), and the distributions p, q that attain it. While sampling from these minimum variance distributions is in most cases intractable, for ℓ_1-ℓ_1 and ℓ_2-ℓ_2 games the minimum-variance estimator has the same constant L_{coord} appearing in Table 1, and in that sense our sampling scheme is optimal. For ℓ_2-ℓ_1 games, the optimal p, q produce the constant $\max_i \|A_{i:}\|_1 \vee \||A|\|_{\text{op}}$, where $\||A|\|_{\text{op}}$ denotes the operator norm of the element-wise absolute values of A. This constant can be significantly smaller than the value of L_{coord} we obtain. How to attain the optimal constant with probabilities that can be sampled efficiently—and whether it is at all possible—remains an open question.