Gradient Methods for Stochastic Optimization in Relative Scale

Anton Rodomanov

(Joint work with Y. Nesterov)

UCLouvain, Belgium

May 31, 2023

SIAM Conference on Optimization (OP23), Seattle

Outline

1. Introduction
2. Problem Formulation
3. Gradient Method
4. Oracle for Maximal Eigenvector
5. Dual Averaging Method
6. Numerical Experiments
7. Application: MaxCut Problem
8. Conclusions

Outline

1. Introduction
2. Problem Formulation
3. Gradient Method
4. Oracle for Maximal Eigenvector
5. Dual Averaging Method
6. Numerical Experiments
7. Application: MaxCut Problem
8. Conclusions

Optimization in Relative Scale

Problem

$$f^* := \min_{x \in Q} f(x),$$

where $f : \mathbb{E} \to \mathbb{R}$ is a strictly positive convex function and $Q \subseteq \mathbb{E}$ is a simple convex set.

Goal: Find approximate solution $\bar{x} \in Q$ with relative accuracy $\delta \in (0, 1)$:

$$f(\bar{x}) \leq (1 + \delta) f^*.$$

Main feature: Complexity of methods does not depend on f^* (or data defining the problem).

Relative Accuracy vs Absolute Accuracy

Note:

- δ-relatively inexact solution $\iff$ ϵ-absolutely inexact solution:

$$f(\bar{x}) \leq (1+\delta) f^* \quad \iff \quad f(\bar{x}) - f^* \leq \epsilon, \qquad \epsilon := \delta f^*.$$

- However, usually we cannot achieve what we want "for free" by simply reusing existing methods and complexity results.

Example (Gradient Method). To achieve δ-relative accuracy, it needs

$$N = \frac{LR_0^2}{\epsilon} = \frac{LR_0^2}{\delta f^*}$$

iterations, where L is the Lipschitz constant of ∇f and $R_0 := \|x_0 - x^*\|$. Depending on f^*, N could be very large, even if δ is moderate!

We need special methods.

Context and Contributions

- There already exist several gradient-type methods for optimization in relative scale: (Nesterov, 2008; Nesterov, 2009; Nesterov, 2010).
- However, they all work with an exact oracle (exact computations of objective function and/or its gradient).

This work

New first-order methods with inexact stochastic oracle.

Motivating Example

Spectral Linear Regression (SLR) problem

$$\min_{x \in \mathbb{R}^d} \|A(x) - C\|, \qquad A(x) := \sum_{i=1}^{d} x_i A_i,$$

where $A_1, \ldots, A_d, C \in \mathbb{R}^{n \times m}$ $(n \leq m)$, $\|\cdot\|$ is the matrix spectral norm.

- Computing subgradient requires computing a pair of leading singular vectors of an $n \times m$ matrix.
- Cost of exact computation: $O(n^3) \implies$ infeasible for large n (and difficult to exploit sparsity).
- But approximate singular vectors are much more affordable: they require only a small number of matrix-vector products.
- The corresponding linear algebra methods are typically randomized, so we obtain inexact approximate subgradients.

Outline

1. Introduction
2. Problem Formulation
3. Gradient Method
4. Oracle for Maximal Eigenvector
5. Dual Averaging Method
6. Numerical Experiments
7. Application: MaxCut Problem
8. Conclusions

Problem Formulation

Problem

$$\min_{x \in Q} f(x),$$

where $f : \mathbb{E} \to \mathbb{R}$ is a convex function and $Q \subseteq \mathbb{E}$ is a simple convex set.

Main assumptions:

- f has quadratic growth: there exists $x_0 \in Q$ and $\gamma_0 > 0$ such that

$$f(x) \geq \gamma_0 \|x - x_0\|_B^2, \qquad \forall x \in Q,$$

  where $\|h\|_B := \langle Bh, h \rangle^{1/2}$.
- We have a δ-relatively inexact stochastic subgradient oracle $\hat{g}$:

$$f(y) \geq (1 - \delta) f(x) + \langle \mathbb{E}_\xi[\hat{g}(x, \xi)], y - x \rangle, \qquad \forall x, y \in Q.$$

- The size of $\hat{g}$ is relatively bounded:

$$\mathbb{E}_\xi[(\|\hat{g}(x, \xi)\|_B^*)^2] \leq 2Lf(x), \qquad \forall x \in Q.$$

Example: Squared Spectral Norm

$$F(X) := \|X\|^2 = \lambda_{\max}(XX^T), \qquad X \in \mathbb{R}^{n \times m}, \qquad n \leq m.$$

Quadratic growth: We have (w.r.t. Frobenius norm):

$$\gamma_0 = \frac{1}{n}, \qquad X_0 = 0.$$

Relatively inexact stochastic subgradient:

$$\hat{G}(X) := 2\hat{u}\hat{u}^T X, \qquad \hat{u} \sim \text{MaxEV}(XX^T, \delta).$$

δ-relatively inexact stochastic eigenvector

Given a matrix $A \in \mathbb{S}^n_+$, compute $\hat{u} \sim \text{MaxEV}(A, \delta)$ such that

$$\mathbb{E}\langle A\hat{u}, \hat{u}\rangle \geq (1-\delta)\lambda_{\max}(A), \qquad \|\hat{u}\| = 1.$$

Relative boundedness: $\hat{G}$ is relatively bounded w.r.t. F with $L = 2$:

$$\|\hat{G}(X)\|_F^2 = 4\langle XX^T\hat{u}, \hat{u}\rangle \leq 4\lambda_{\max}(XX^T) = 4F(X).$$

Composition with Affine Mapping

Consider

$$f(x) = F(Ax + b), \qquad x \in \mathbb{E},$$

where $A\colon \mathbb{E} \to \mathbb{E}_1$, $b \in \mathbb{E}_1$, and F satisfies our assumptions:

- F has quadratic growth w.r.t. $\|\cdot\|_{B_1}$ with parameters γ_0 and y_0.
- We have δ-relative stochastic oracle $\hat{G}$ for F.
- Oracle $\hat{G}$ is uniformly relatively bounded with constant L.

Define the seminorm induced by $B = A^* B_1 A$:

$$\|x\|_B = \|Ax\|_{B_1}, \quad \forall x \in \mathbb{E}$$

and stochastic oracle

$$\hat{g}(x) := A^* \hat{G}(Ax + b), \qquad x \in \mathbb{E}.$$

Then, all properties are satisfied with the same constants γ_0, L, and

$$x_0 = \operatorname*{argmin}_{x \in Q} \|Ax + b - y_0\|_{B_1}.$$

Outline

1. Introduction
2. Problem Formulation
3. Gradient Method
4. Oracle for Maximal Eigenvector
5. Dual Averaging Method
6. Numerical Experiments
7. Application: MaxCut Problem
8. Conclusions

Gradient Method with Relative Inexact Stochastic Oracle

$$\hat{g}_k \sim \hat{g}(x_k), \quad x_{k+1} = \mathrm{Prox}_{Q,B}(x_k, a_k \hat{g}_k), \qquad k \geq 0,$$

where $\mathrm{Prox}_{Q,B}(x, s) := \mathrm{argmin}_{y \in Q}\{\langle s, y\rangle + \frac{1}{2}\|y - x\|_B^2\}$, and $(a_k)_{k=0}^{\infty}$ are deterministic step sizes.

Output point: Suppose $a_k < \frac{1-\delta}{L}$ for all $k \geq 0$. Define

$$\bar{x}_k := \frac{1}{C_k} \sum_{i=0}^{k-1} c_i x_i, \qquad C_k := \sum_{i=0}^{k-1} c_i, \qquad c_i := a_i(1 - \delta - L a_i).$$

Theorem. For any $k \geq 0$, we have

$$(1 - \Delta_k)\, \mathbb{E}\, f(\bar{x}_k) \leq f^*, \qquad \Delta_k := \delta + \frac{1 - \delta + 2\gamma_0 L \sum_{i=0}^{k-1} a_i^2}{1 + 2\gamma_0 \sum_{i=0}^{k-1} a_i}.$$

Choice of Stepsizes

Optimal step sizes for a fixed horizon $N \geq 1$

$$a_k \equiv \frac{1-\delta}{\sqrt{2\gamma_0 NL(1-\delta)+L^2}+L}, \qquad k \geq 0.$$

Convergence rate:

$$\Delta_N \leq \delta + \sqrt{\frac{2L}{\gamma_0 N}} \quad \Longrightarrow \quad \Delta_N \leq 2\delta, \quad \forall N \geq N(\delta) := \frac{2L}{\gamma_0 \delta^2}$$

Note: For SLR, we have $L = 2$ and $\gamma_0 = \frac{1}{n}$, so $N(\delta) = \frac{4n}{\delta^2}$ does not depend on data defining the problem.

Constant step size based on target accuracy $\delta \in (0, \frac{2}{3})$

$$a_k \equiv \frac{\delta}{2L}, \qquad k \geq 0.$$

$\Longrightarrow$ same rate for Δ_N.

Outline

1. Introduction
2. Problem Formulation
3. Gradient Method
4. Oracle for Maximal Eigenvector
5. Dual Averaging Method
6. Numerical Experiments
7. Application: MaxCut Problem
8. Conclusions

Stochastic Oracle for Maximal Eigenvector

δ-relatively inexact stochastic eigenvector ($\delta \in (0, 1)$)

Given a matrix $A \in \mathbb{S}^n_+$, compute $\hat{u} \sim \text{MaxEV}(A, \delta)$ such that

$$\mathbb{E}\langle A\hat{u}, \hat{u}\rangle \geq (1 - \delta)\lambda_{\max}(A), \qquad \|\hat{u}\| = 1.$$

How to compute $\text{MaxEV}(A, \delta)$ efficiently?

Power Method

Let $A \in \mathbb{S}^n_+$. For an integer degree $p \geq 1$, define

$$\hat{u}_p := \frac{A^p \xi}{\|A^p \xi\|}, \qquad \xi \sim \mathrm{Unif}(\mathcal{S}^{n-1}).$$

Should be computed in a numerically stable way:

Power Method

$$\hat{u}_0 := \xi, \qquad \hat{u}_{k+1} := \frac{A\hat{u}_k}{\|A\hat{u}_k\|}, \quad k = 0, \ldots, p-1.$$

Complexity: p matrix-vector products.

Main result (Kuczyński and Woźniakowski, 1992)

Suppose $n \geq 8$ and $p \geq 2$. Then,

$$\mathbb{E}\langle A\hat{u}_p, \hat{u}_p \rangle \geq (1 - \delta_p)\lambda_{\max}(A), \qquad \delta_p = 0.871 \frac{\ln n}{p}.$$

Lanczos Method

Krylov subspace: $\mathcal{K}_p(A, \xi) := \mathrm{span}(\xi, A\xi, A^2\xi, \ldots, A^p\xi)$.

Lanczos Algorithm

$$\hat{u}_p \in \mathrm{Argmax}\big\{\langle Ax, x\rangle : x \in \mathcal{K}_p(A, \xi) \cap \mathcal{S}^{n-1}\big\}, \qquad \xi \sim \mathrm{Unif}(\mathcal{S}^{n-1}).$$

Accuracy estimate (Kuczyński and Woźniakowski, 1992)

Suppose $n \geq 8$, $p \geq 3$. Then,

$$\mathbb{E}\langle A\hat{u}_p, \hat{u}_p\rangle \geq (1 - \delta_p)\lambda_{\max}(A), \qquad \delta_p = 2.575\left(\frac{\ln n}{p}\right)^2.$$

Implementing Lanczos Method

Lanczos Algorithm: $\hat{u} = \text{LanczosAlg}(A, p)$

$(T, Q) \coloneqq \text{LanczosTridiag}(A, \xi, p)$ with $\xi \sim \text{Unif}(\mathcal{S}^{n-1})$.

$\hat{x} = \text{MaxEVTridiag}(T)$.

return $\hat{u} \coloneqq Q\hat{x}$.

Lanczos Tridiagonalization: $(T, Q) = \text{LanczosTridiag}(A, \xi, p)$

$q_0 \coloneqq \xi, \quad \alpha_0 \coloneqq \langle Aq_0, q_0 \rangle, \quad r_0 \coloneqq Aq_0 - \alpha_0 q_0.$

for $k = 0, \ldots, p-1$ **do**

$\beta_k \coloneqq \|r_k\|, \quad q_{k+1} \coloneqq r_k/\beta_k, \quad \alpha_{k+1} \coloneqq \langle Aq_{k+1}, q_{k+1} \rangle.$

$r_{k+1} \coloneqq Aq_{k+1} - \alpha_{k+1} q_{k+1} - \beta_k q_k.$

return $T \coloneqq \text{Tridiag}(\alpha_0, \ldots, \alpha_p; \beta_0, \ldots, \beta_{p-1}), \quad Q \coloneqq [q_0, \ldots, q_p].$

Complexity: $p + 1$ matrix-vector products $+ O(np)$.

Outline

1. Introduction
2. Problem Formulation
3. Gradient Method
4. Oracle for Maximal Eigenvector
5. Dual Averaging Method
6. Numerical Experiments
7. Application: MaxCut Problem
8. Conclusions

Drawbacks of Previously Considered Gradient Method

In the previously considered Gradient Method, we first decide on the target relative accuracy δ we want to achieve and then use at each iteration:

- Fixed oracle accuracy $\delta' := \delta/2$.
- Constant step size $a_k \equiv \delta'/(2L)$.

Disadvantages:

- Need to know δ in advance.
- The method is essentially short-step.
- Always ask for the same high oracle accuracy.

Natural idea: Use time-varying step sizes and oracle accuracies
$\Longrightarrow$ Dual Averaging Method (same worst-case complexity).

Dual Averaging Method

$$v_{k+1} = \operatorname{argmin}_{x \in Q} \left\{ \sum_{i=1}^{k} a_i [(1-\delta_i) f(w_i) + \langle g_i, x - w_i \rangle] + \frac{\beta_k}{2} \|x - x_0\|_B^2 \right\}.$$

DualAvg $\left(x_0, L, (a_k)_{k=1}^{\infty}, (\beta_k)_{k=0}^{\infty}, (\delta_k)_{k=1}^{\infty}\right)$.

$v_0 := x_0, \quad \bar{g}_0 := 0 \ (\in \mathbb{E}^*), \quad A_0 := C_0 := 0 \ (\in \mathbb{R})$.

for $k \geq 0$ **do**

$w_{k+1} := \left(\beta_k v_k + (\beta_{k+1} - \beta_k) x_0\right) / \beta_{k+1}, \quad g_{k+1} \sim g(w_{k+1}, \delta_{k+1})$.

$A_{k+1} := A_k + a_{k+1}, \quad \bar{g}_{k+1} := (A_k \bar{g}_k + a_{k+1} g_{k+1}) / A_{k+1}$.

$v_{k+1} := \operatorname{Prox}_{Q,B}\left(x_0, \frac{A_{k+1}}{\beta_{k+1}} \bar{g}_{k+1}\right)$.

$c_{k+1} := a_{k+1}\left(1 - \delta_{k+1} - \frac{L a_{k+1}}{\beta_{k+1}}\right), \quad C_{k+1} := C_k + c_{k+1}$.

$x_{k+1} := (C_k x_k + c_{k+1} w_{k+1}) / C_{k+1}$.

Choice of parameters:

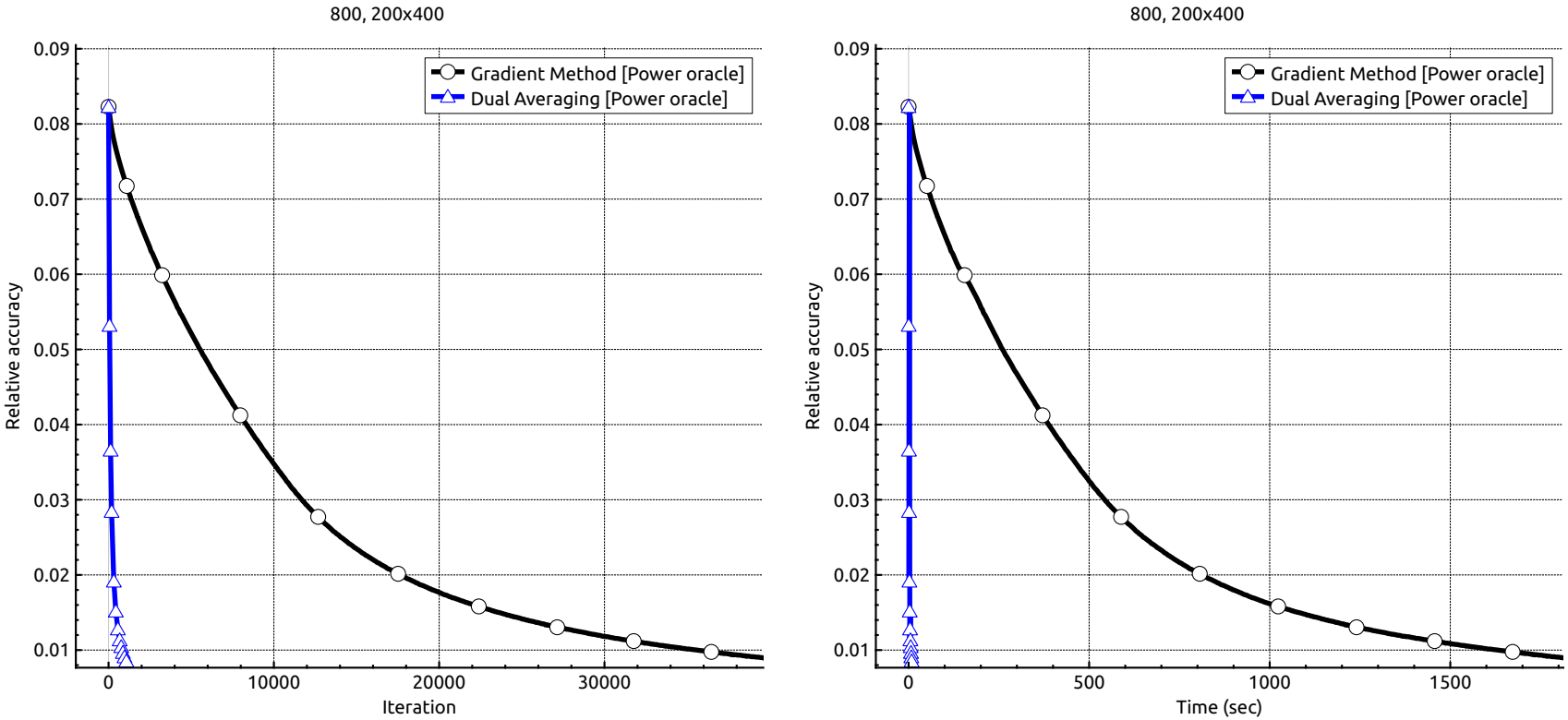
$$a_k := 1, \qquad \beta_k := \sqrt{8\gamma_0 L k} + 2L, \qquad \delta_k := \frac{L a_k}{\beta_k} = \frac{1}{\sqrt{8\gamma_0 k / L} + 2}.$$

Outline

1. Introduction
2. Problem Formulation
3. Gradient Method
4. Oracle for Maximal Eigenvector
5. Dual Averaging Method
6. Numerical Experiments
7. Application: MaxCut Problem
8. Conclusions

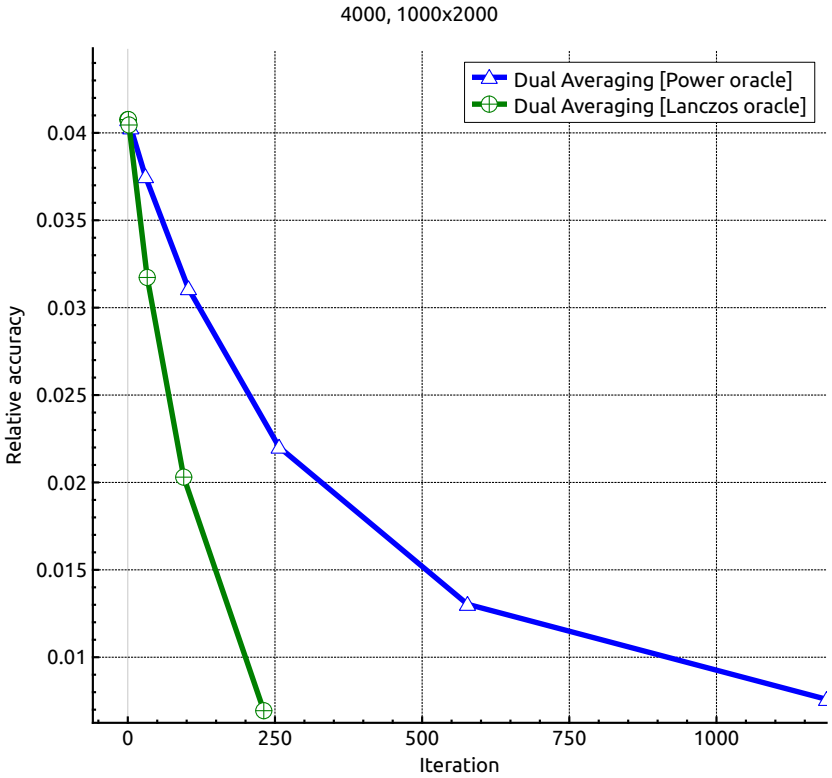
Gradient Method vs Dual Averaging

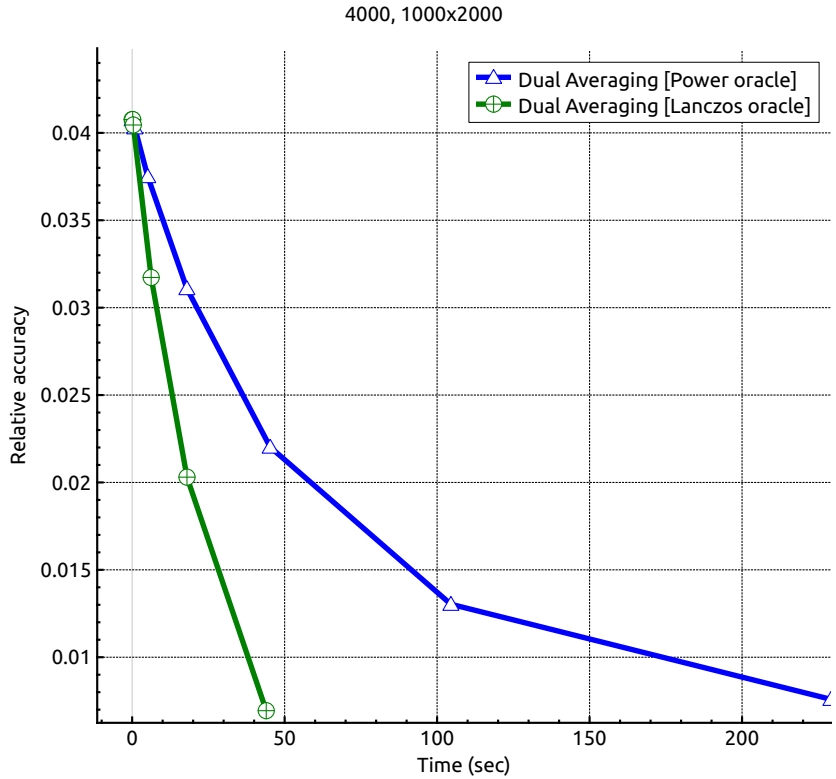
Problem: Spectral linear regression ($\delta = 0.01$ for Gradient Method).



- Dual Averaging is by *orders of magnitude* faster.
- Actual # iterations is by *orders of magnitude* smaller than the theoretical one: $N(\delta) = 8\,080\,605$.

Power Oracle vs Lanczos Oracle





Outline

1. Introduction
2. Problem Formulation
3. Gradient Method
4. Oracle for Maximal Eigenvector
5. Dual Averaging Method
6. Numerical Experiments
7. Application: MaxCut Problem
8. Conclusions

MaxCut Problem

Let $G = (V, E)$ be an undirected weighted graph with $V = \{1, \ldots, n\}$ and weights $w(\{i, j\}) > 0$ for each edge $\{i, j\} \in E$.

MaxCut problem

$$c^* = \frac{1}{4} \max_{x \in B^n} \langle Ax, x \rangle,$$

where B^n is the boolean hypercube:

$$B^n := \{x \in \mathbb{R}^n : x_i^2 = 1, \ i = 1, \ldots, n\},$$

and $A \in \mathbb{S}_+^n$ is the Laplacian matrix of G:

$$A_{i,j} := \begin{cases} \sum_{k:\, \{i,k\} \in E} w(\{i, k\}), & \text{if } i = j, \\ -w(\{i, j\}), & \text{if } \{i, j\} \in E, \\ 0, & \text{otherwise.} \end{cases}$$

Note: NP-complete! But can be efficiently approximated.

SDP Relaxation

MaxCut problem:

$$s^* := \max_{x \in B^n} \langle Ax, x \rangle.$$

SDP relaxation:

$$f^* := \underbrace{\min_{z \in \mathbb{R}^n} \left\{ \sum_{i=1}^{n} z_i : A \preceq D(z) \right\}}_{\text{Dual SDP relaxation}} = \underbrace{\max_{Y \in \mathbb{S}^n} \left\{ \langle A, Y \rangle : Y \succeq 0, \ d(Y) = e \right\}}_{\text{Primal SDP relaxation}},$$

where $e := (1, \ldots, 1)^T \in \mathbb{R}^n$.

Accuracy of relaxation (Goemans and Williamson, 1995)

$$0.878 \cdot f^* \leq s^* \leq f^*.$$

SDP Relaxation: Alternative Form

Problem

$$f^* = \min_{x \in Q} \left[f(x) := \lambda_{\max}\big(S(x)\big) \right], \qquad S(x) := D(x) A D(x),$$

where

$$Q := \left\{ x \in \mathbb{R}^n_{++} : \sum_{i=1}^n \frac{1}{x_i^2} \leq 1 \right\}.$$

Oracle:

$$\hat{g}(x) := 2[A(x \odot \hat{u})] \odot \hat{u} \ \left[= 2d(AD(x)\hat{u}\hat{u}^T) \right], \quad \hat{u} \sim \text{MaxEV}\big(S(x), \delta\big).$$

Choice of norm: $B = D(A)$. Then, all our assumptions are satisfied with

$$\gamma_0 = \frac{1}{n}, \qquad x_0 = \operatorname*{argmin}_{x \in Q} \|x\|_B = \text{Proj}_{Q,B}(0), \qquad L = 2.$$

Note: Since B is diagonal, projection is cheap: $O(n)$.

Final Guarantee

We get $x_k \in Q$ with $(1-\delta)\,\mathbb{E}\, f(x_k) \leq f^*$ in at most $N(\delta)$ iterations,

$$N(\delta) = O\Big(\frac{L}{\gamma_0 \delta^2}\Big) = O\Big(\frac{n}{\delta^2}\Big).$$

Result

For $\hat{f}_k := (1-\delta)^{-1}\langle S(x_k)\hat{u}, \hat{u}\rangle,\ \ \hat{u} \sim \mathrm{MaxEV}\big(S(x_k), \delta\big)$, we have

$$\alpha\, \mathbb{E}\, \hat{f}_k \leq s^* \leq \mathbb{E}\, \hat{f}_k, \qquad \alpha := 0.878(1-\delta)^2.$$

Total worst-case running time:

$$N(\delta) \times \underbrace{O\Big(\frac{\ln n}{\sqrt{\delta}}\Big)}_{\substack{\text{Lanczos number of} \\ \text{mat-vec products}}} \times \underbrace{O(|E|)}_{\substack{\text{Cost of} \\ \text{mat-vec product}}} = O\Big(\frac{n|E|\ln n}{\delta^{5/2}}\Big).$$

Note: No need for very small δ: $\delta = 0.01 \implies \alpha \approx 0.86$ (instead of $\alpha' = 0.878$ for exact solution of SDP).

Outline

1. Introduction
2. Problem Formulation
3. Gradient Method
4. Oracle for Maximal Eigenvector
5. Dual Averaging Method
6. Numerical Experiments
7. Application: MaxCut Problem
8. Conclusions

Conclusions

Overview:

- New concept of relatively inexact stochastic subgradient.
- Arises naturally when computing inexact eigenvectors by using the Power method or the Lanczos algorithm.
- Two gradient methods for large-scale optimization in relative-scale: Gradient Method with fixed stepsize and oracle accuracy and the more practical Dual Averaging.

Paper

`https://arxiv.org/abs/2301.08352`

Thank you!

References


M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, Nov. 1995. ISSN: 0004-5411. DOI: 10.1145/227683.227684.


J. Kuczyński and H. Woźniakowski. Estimating the Largest Eigenvalue by the Power and Lanczos Algorithms with a Random Start. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1094–1122, Oct. 1992. DOI: 10.1137/0613066.


Y. Nesterov. Rounding of convex sets and efficient gradient methods for linear programming problems. *Optimization Methods and Software*, 23(1):109–128, Feb. 2008. DOI: 10.1080/10556780701550059.


Y. Nesterov. Unconstrained convex minimization in relative scale. *Mathematics of Operations Research*, 34(1):180–193, 2009. ISSN: 0364765X, 15265471.


Y. Nesterov. Barrier subgradient method. *Mathematical Programming*, 127(1):31–56, Oct. 2010. DOI: 10.1007/s10107-010-0421-3.