A stochastic smoothing algorithm for semidefinite programming

Authors: Alexandre d'Aspremont and Noureddine El Karoui,

Repoters: Pan ZHOU and Bin HAO

Peking University

pzhou@pku.edu.cn
haobin@pku.edu.cn

December 16, 2014

Overview

Introduction

- Focus on **maximum eigenvalue minimization**

$$\textbf{P:} \qquad \min_{X \in Q} \lambda_{max}(A(X)),$$

where $X \in S_n$.

- The set Q is **convex and simple**, i.e., projections on Q can be computed with low conplexity.

- $A(X)$ is a **simple function**. In most case, $A(X)$ is a linear function of X.
- All semidefinite programs with constant trace can be expressed in this form.

Projected Subgradient

- Solve

$$\textbf{P:} \qquad \min_{X \in Q} \lambda_{max}(A(X))$$

by **Projected Subgradient**.

Input: A starting point $X_0 \in S_n$.

1. for $t = 0$ to $N - 1$ do

2. update

$$X_{t+1} = P_Q(X_t - \gamma \partial \lambda_{max}(A(X_t))).$$

3. end for

Output: A point $X = \frac{1}{N} \sum_{t=1}^{N} X_t$.

- Here, $\gamma > 0$ and $P_Q(\cdot)$ is the Euclidean Projiection on Q.

Projected Subgradient

- The **number of iterations** required to reach a target precision ϵ is

$$N = \frac{D_Q^2 M^2}{\epsilon^2},$$

where D_Q is the diameter of Q and $\|\partial\lambda_{max}(A(X))\| \leq M$ on Q.

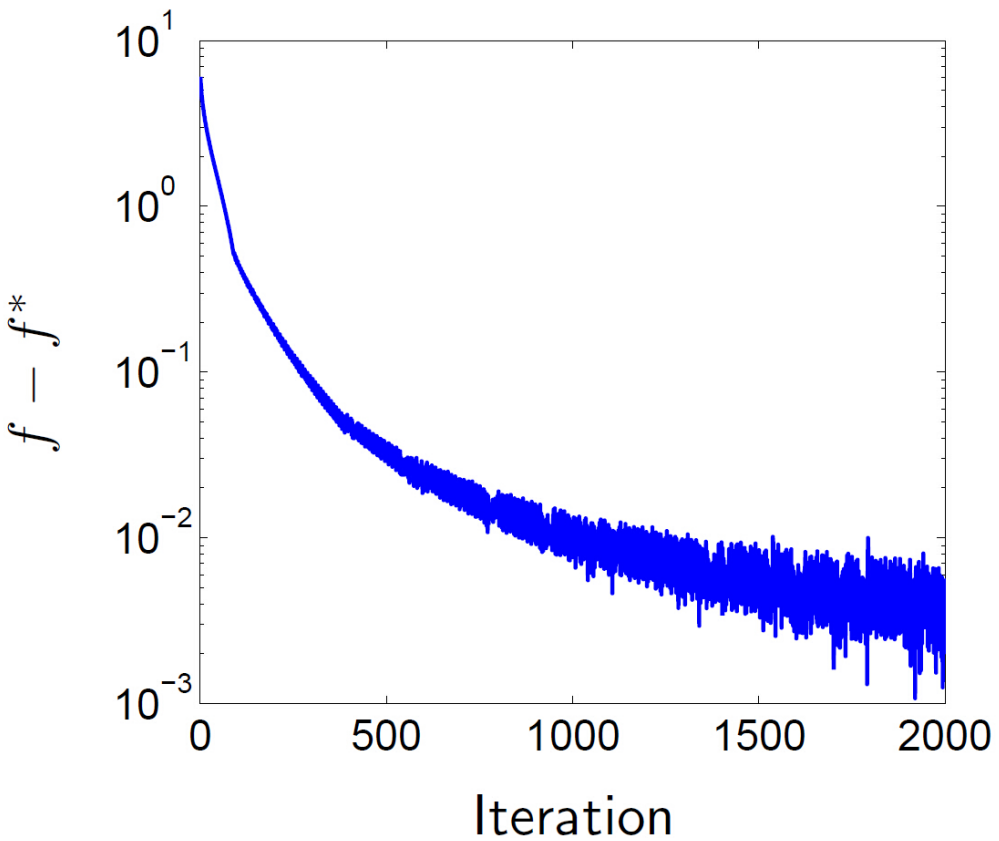
- The **cost per iteration** is the sum of
  - The cost P_Q of computing the Euclidean Projection on Q.
  - The cost of computing $\partial\lambda_{max}(A(X))$ which is $v_1 v_1^T$, where v_1 is the leader eigenvector of $A(X)$.

$$O(\frac{n^2 \log n/\delta^2}{\sqrt{\epsilon}})$$

where ϵ is the precision and $1-\delta$ is the probability of failure.

Projected Subgradient

- Solving $\min_{X \in Q} \lambda_{max}(A(X))$ by Projected Subgradient.
  - Easy to implement.
  - Very poor performance in practice. The $1/\epsilon^2$ dependence is somewhat punishing.
- Example below on MAXCUT.



Smooth method

- Solving $\min_{X \in Q} \lambda_{max}(A(X))$ by **smooth method**.
- We can reglarize the objective and solve [Nesterov, 2007]

$$\min_{X \in Q} f(X) = \mu \log \mathrm{Tr} \left(\exp(\frac{A(X)}{\mu}) \right)$$

  where some regularization parameter $\mu > 0$.
- If we set $\mu = \epsilon / \log n$,

$$\lambda_{max}(A(X)) \leq f(X) \leq \lambda_{max}(A(X)) + \epsilon$$

- The gradient $\nabla f(X)$ is Lipschitz continuous with constant

$$\frac{\|A\|^2 \log n}{\epsilon}$$

  where $\|A\| = \sup_{\|h\| \leq 1} \|A(h)\|_2$.

Smooth method

- The **number of iterations** required to obtain an ϵ by **smooth method** grows as

$$\frac{D_Q \sqrt{\log n}}{\epsilon}$$

- The **cost per iteration** is the sum of
  - The cost P_Q of computing the Euclidean Projection on Q.
  - The cost of computing the mtrix exponentil $exp(A(X)/\mu)$

$$O(n^3).$$

Smooth method

- This means that the two classical complexity options for solving

$$\min_{X \in Q} \lambda_{max}(A(X))$$

- **Subgradient methods**

$$\frac{D_Q^2(n^2 \log n + P_Q)}{\epsilon^2}$$

- **Smooth methods**

$$\frac{D_Q \sqrt{\log n}(n^3 + P_Q)}{\epsilon}$$

Subgradient method VS Smooth method

- This means that the two classical complexity options for solving

$$\min_{X \in Q} \lambda_{max}(A(X))$$

- **Subgradient methods**

$$\frac{D_Q^2(n^2 \log n + P_Q)}{\epsilon^2}$$

- **Smooth methods**

$$\frac{D_Q \sqrt{\log n}(n^3 + P_Q)}{\epsilon}$$

- Keep some of the performance of smooth methods, while lowering the cost of smoothing.
- One possible solution here: **stochastic gradient approximations**.

Stochastic Smoothing Algorithm

- Solve a smooth approximation problem, written as

$$\min_{X \in Q} \mathbf{F}_k(X) = \mathbf{E}[\max_{i=1,\cdots,k} \lambda_{max}(X + \epsilon z_i z_i^T)]$$

where $z_i \overset{i.i.d}{\sim} N(0, I_n)$, $\epsilon > 0$.

- **Property 1:** Approximation results are preserved up to a constant $c_k > 0$

$$\lambda_{max}(X) \leq \mathbf{E}[\max_{i=1,\cdots,k} \lambda_{max}(X + \epsilon z_i z_i^T)] \leq \lambda_{max}(X) + c_k \epsilon$$

- **Property 2:** The function $\mathbf{F}_k(X)$ is smooth and has a Lipschitz continuous gradient

$$\|\nabla \mathbf{F}_k(X) - \nabla \mathbf{F}_k(Y)\|_F \leq L \|X - Y\|_F,$$

where L satisfies

$$L \leq \mathbf{E}[\frac{n}{2\epsilon} \max_{i=1,\cdots,k} 1/u_{i,1}^2] \leq c_k \frac{n}{\epsilon} \quad \text{and} \quad c_k = \frac{k}{\sqrt{2}(k-2)}.$$

Stochastic Smoothing Algorithm

- **Property 3:** The gradient variance of $\mathbf{F}_k(X)$ can be bounded. Let ϕ_{i_0} be the leading eigvector of matrix $X + \frac{\epsilon}{n} z_{i_0} z_{i_0}^T$, where

$$i_0 = \arg \max_{i=1,\cdots,k} \lambda_{max}(X + \epsilon z_i z_i^T).$$

  Then, we have

$$\nabla \mathbf{F}_k(X) = \mathbf{E}(\phi_{i_0} \phi_{i_0}^T) \quad \text{and} \quad \mathbf{E}(\|\phi_{i_0} \phi_{i_0}^T - \nabla \mathbf{F}_k(X)\|) \leq 1.$$

- **Property 4:** The optimal algorithm for stochastic optimazation derived in [Lan, 2012] will produce a matrix X_N such that

$$\mathbf{E}(\mathbf{F}_k(X_N) - \mathbf{F}_k(X^*) \leq \frac{4LD_Q^2}{\alpha N^2} + \frac{4D_Q}{\sqrt{Nq}}$$

  where α is the strong convexity of the prox function. q is the number of independent samples matrixs $\phi\phi^T$ averaged in approximating the gradient.

Maximum Eigenvalue Minimization

Solve maximum eigenvalue minimization after stochastic smoothing

$$\min_{x \in Q} \Psi(X) = \mathrm{E}[\Psi(X, z)] = \mathrm{E}\left[\max_{j=1,\ldots,3} \lambda_{max}\left(X + \frac{\epsilon}{n} z_j z_j^T\right)\right]$$

in the variable $X \in \mathcal{S}_n$ and the z_j are Gaussian.

We use an optimal stochastic minimization algorithm in [Lan, 2009] which is a generalization of the algorithm in Nesterov [1983], with increasing step size.

Maximum Eigenvalue Minimization

Optimal Stochastic Composite Optimization. The algorithm in Lan [2009] solves

$$\min_{x \in Q} \Psi(x) := f(x) + h(x)$$

with the following assumptions

- $f(x)$ has Lipschitz gradient with constant L and $h(x)$ is Lipschitz with constant M,
- we have a **stochastic oracle** $G(x, \xi_t)$ for the gradient, which satisfies

$$\mathrm{E}[G(x, \xi_t)] = g(x) \in \partial \Psi(x),$$

$$\mathrm{E}[\| G(x, \xi_t) - g(x) \|_*^2] \leq \sigma^2.$$

Maximum Eigenvalue Minimization

- Distance generating function $\omega(x)$, i.e. a function such that

$$Q^o = \left\{ x \in Q : \ \exists y \in \mathcal{R}^p, \ x \in \operatorname{argmin}_{u \in Q} [y^T u + \omega(u)] \right\}$$

is convex set.he function $\omega(x)$ is strongly convex on Q^o with modulus α with respect to the norm $\|\cdot\|$, which means

$$(y - x)^T (\nabla \omega(y) - \nabla \omega(x)) \geq \alpha \|y - x\|^2, \quad x, y \in Q^o.$$

We then define a **Bregman distance** $V(x, y)$ on $Q^o \times Q$ as follows:

$$V(x, y) \equiv \omega(y) - [\omega(x) + \nabla \omega(x)^T (y - x)].$$

Maximum Eigenvalue Minimization

- The prox-mapping associated to V is then defined as

$$P_x^{Q,\omega}(y) \equiv \operatorname{argmin}_{z \in Q}\{y^T(z-x) + V(x,z)\}.$$

After N iterations, the iterate x_{N+1} satisfies

$$\mathrm{E}[\Psi(x_{N+1}^{ag}) - \Psi^*] \leq \frac{8LD_{\omega,Q}^2}{N^2} + \frac{4D_{\omega,Q}\sqrt{4M^2+\sigma^2}}{\sqrt{N}}$$

which is optimal. Additional assumptions guarantee convergence w.h.p.

Maximum Eigenvalue Minimization

Stochastic line search.

- The bounds on variance and smoothness are very conservative.
- Line search allows to take full advantage of the smoothness of $\lambda_{\max}(X)$ outside of pathological areas.

Monotonic line search. In Lan [2009], we test

$$\begin{aligned}\Psi(x_{t+1}^{ag}, \xi_{t+1}) \leq & \Psi(x_t^{md}, \xi_t) + \langle G(x_t^{md}, \xi_t), x_{t+1}^{ag} - x_t^{md} \rangle \\ & + \frac{\alpha\beta_t}{4\gamma_t} \|x_{t+1}^{ag} - x_t^{md}\|^2 + 2\mathcal{M}\|x_{t+1}^{ag} - x_t^{md}\|\end{aligned}$$

while decreasing the step size monotonically across iterations.

Require: An initial point $x^{ag} = x_1 = x^w \in \mathcal{R}^n$, an iteration counter $t = 1$, the number of iterations N, line search parameters $\gamma^{min}, \gamma^{max}, \gamma^d, \gamma > 0$, with $\gamma^d < 1$.

1: Set $\gamma = \gamma^{max}$.
2: **for** $t = 1$ to N **do**
3: Define $x_t^{md} = \frac{2}{t+1} x_t + \frac{t-1}{t+1} x_t^{ag}$
4: Call the stochastic gradient oracle to get $G(x_t^{md}, \xi_t)$.
5: **repeat**
6: Set $\gamma_t = \frac{(t+1)\gamma}{2}$.
7: Compute the prox mapping $x_{t+1} = P_{x_t}(\gamma_t G(x_t^{md}, \xi_t))$.
8: Set $x_{t+1}^{ag} = \frac{2}{t+1} x_{t+1} + \frac{t-1}{t+1} x_t^{ag}$.
9: **until** $\Psi(x_{t+1}^{ag}, \xi_{t+1}) \leq \Psi(x_t^{md}, \xi_t) + \langle G(x_t^{md}, \xi_t), x_{t+1}^{ag} - x_t^{md} \rangle + \frac{\alpha\gamma^d}{4\gamma} \|x_{t+1}^{ag} - x_t^{md}\|^2 + 2\mathcal{M}\|x_{t+1}^{ag} - x_t^{md}\|$ or $\gamma \leq \gamma^{min}$. If exit condition fails, set $\gamma = \gamma\gamma^d$ and go back to step 5.
10: Set $\gamma = \max\{\gamma^{min}, \gamma\}$.
11: **end for**

Ensure: A point x_{N+1}^{ag}.

Maximum Eigenvalue Minimization

Details:

- $x^w = \operatorname{argmin}_{x \in Q} \omega(x)$,
- We use the following gradient oracle

$$G(X, z) = \frac{1}{q} \sum_{l=1}^{q} \phi_l \phi_l^T$$

where each ϕ_l is a leading eigenvector of the matrix $X + \frac{\epsilon}{n} z_{i_0} z_{i_0}^T$, with

$$i_0 = \operatorname{argmax}_{i=1,\ldots,k} \lambda_{\max} \left(X + \frac{\epsilon}{n} z_i z_i^T \right),$$

where z_i are i.i.d. Gaussian vectors $z_i \sim \mathcal{N}(0, \mathrm{I}_n)$ and $k > 0$ is a small constant (typically 3) and q is used to control the variance.

Maximum Eigenvalue Minimization

Details:

- We have

$$\lambda_{\max}(X) \leq \mathrm{E}\left[\max_{i=1,\ldots,k} \lambda_{\max}\left(X + \frac{\epsilon}{n} z_i z_i^T\right)\right] \leq \lambda_{\max}(X) + k\epsilon.$$

Maximum Eigenvalue Minimization

For maximum eigenvalue minimization

- We have $\sigma \leq 1$, but we can reduce this by averaging q gradients, to control the tradeoff between smooth and non-smooth terms.
- If we set $q = \max\{1, D_Q/(\epsilon\sqrt{n})\}$ and $N = 2D_Q\sqrt{n}/\epsilon$ we get the following complexity picture

Complexity	Num. of Iterations	Cost per Iteration
Nonsmooth alg.	$O(\frac{D_Q^2}{\epsilon^2})$	$O(p_Q + n^2 \log n)$
Smooth stochastic alg.	$O(\frac{D_Q\sqrt{n}}{\epsilon})$	$O(p_Q + \max 1, \frac{D_Q}{\epsilon\sqrt{n}} n^2 \log n)$
Smoothing alg.	$O(\frac{D_Q\sqrt{\log n}}{\epsilon})$	$O(p_Q + n^3)$

Solving a problem:

$$\max_X \lambda_{max}(A + X), \qquad s.t. \ -\rho \leq X_{ij} \leq \rho.$$

where $X \in S_n$.

n	Stoch. # iters.	Stoch. # eigvs.	ACSA # iters.	ACSA # eigvs.	Det. # iters.	Det. # eigvs.
50	707	1266	51	2550	16	3700
100	1000	1806	50	5000	12	5800
200	1414	2532	55	11000	28	24800
500	2236	8016	60	30000	12	29000
1000	3162	18990	65	65000	12	56000
2000	4472	21444	66	132000	14	132000

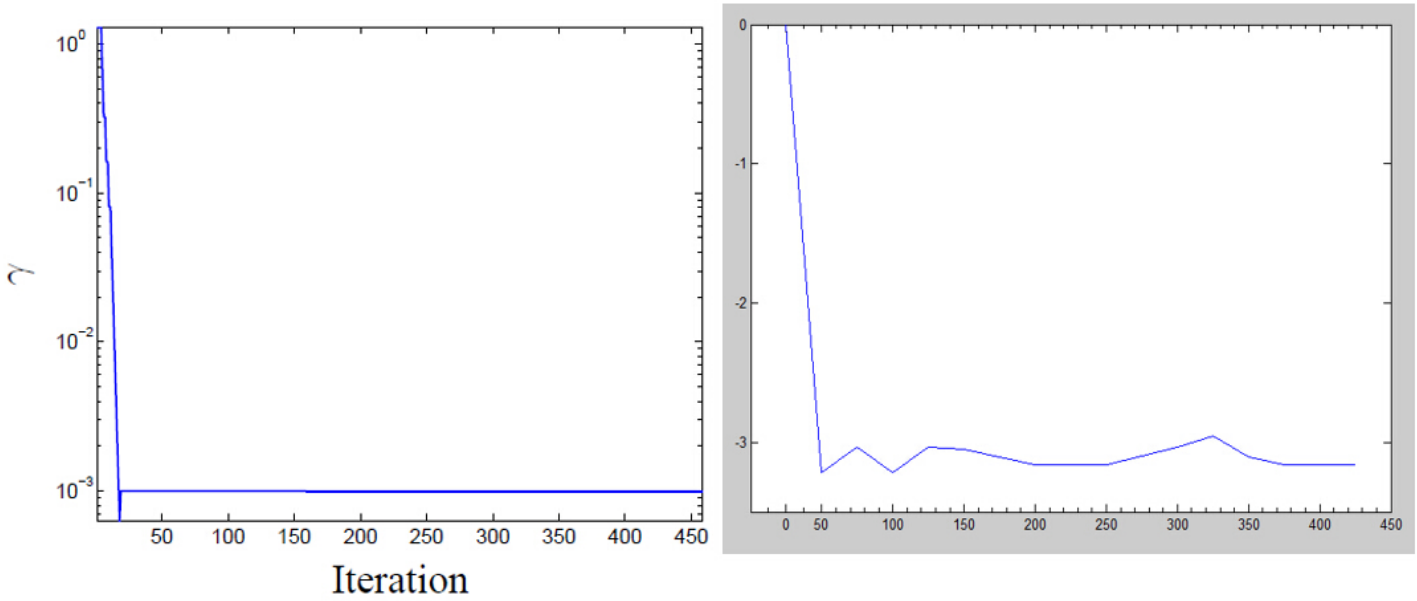
TABLE 2. Number of iterations and total number of eigenvectors computed by Algorithm 1 (Stoch.), the ACSA algorithm in Lan [2012] and the algorithm in [Nesterov, 2007b, §4] (Det.) (both with exponential smoohting) to reach identical objective values when solving the DSPCA relaxation in (29).

n	iters	eigvs
100	1132	2018
500	2684	8506
1000	3744	19612
2000	4798	22094

Solving a problem:

$$\max_X \lambda_{max}(A + X), \qquad s.t. \ -\rho \leq X_{ij} \leq \rho.$$

where $X \in S_n$.



Thank you!

Any questions?