Smooth Optimization for Sparse Semidefinite Programs

Alexandre d'Aspremont
ORFE, Princeton University

Support from EUROCONTROL, NSF and Google.

Introduction

Smooth Optimization:

- Produces smooth (Lipschitz-continuous gradient) approximation of structured semidefinite optimization problems.
- Smooth problem solved using first-order technique in Nesterov (1983).
- Total complexity in $O(1/\epsilon)$ instead of $O(1/\epsilon^2)$.

Smooth semidefinite optimization:

- Difference with I.P. methods: large number of simpler iterations.
- Key step is a **matrix exponential**: can be computed efficiently.

Smoothing technique

Example: **maximum eigenvalue minimization** problem:

$$\min f(x) := \lambda^{\max}(Ax - b)$$

in the variable $x \in \mathbf{R}^n$ with parameters $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$.

Solve **smooth approximation** with:

$$\min f_\mu(x) := \mu \log \left(\mathbf{Tr} \exp \left(\frac{Ax - b}{\mu} \right) \right)$$

where $\log$ and $\exp$ are the matrix (not componentwise) logarithm and exponential, respectively.

Smoothing technique

$f_\mu(Ax-b)$ is a $\mu \log n$-**uniform approximation** of $\lambda^{\max}(Ax-b)$:

$$\lambda^{\max}(Ax-b) \le f_\mu(x) \le \lambda^{\max}(Ax-b) + \mu \log n$$

and the **gradient** of $f_\mu(x)$, given by:

$$\nabla f_\mu(x) := \left(\mathbf{Tr} \exp\left(\frac{Ax-b}{\mu}\right)\right)^{-1} \exp\left(\frac{Ax-b}{\mu}\right)$$

is **Lipschitz continuous** with constant given by:

$$L = \frac{\|A\|^2}{\mu}$$

Smoothing technique

- If we set:

$$\mu = \frac{\epsilon}{2\log n},$$

- solving

$$\min f_\mu(x)$$

  produces an ϵ-approximation of the solution to the original problem.

- Because $\nabla f_m u$ is Lipschitz continuous, Nesterov (1983) shows that the complexity of solving this problem is given by:

$$\frac{4\|A\|}{\epsilon}\sqrt{\frac{\log n\|x^\star\|^2}{2}}$$

Nesterov's method

- Nesterov (2005) shows that this result holds for all problems with a **min-max** format:

$$f(x) = \hat{f}(x) + \max_u \{\langle Tx, u\rangle - \hat{\phi}(u) \ : \ u \in Q_2\}$$

- assuming that:
  - f is defined over a compact convex set $Q_1 \subset \mathbf{R}^n$
  - $\hat{f}(x)$ is convex, differentiable and has a Lipschitz continuous gradient with constant $M \geq 0$
  - T is a linear operator: $T \in \mathbf{R}^{n \times n}$
  - $\hat{\phi}(u)$ is a continuous convex function over some compact set $Q_2 \subset \mathbf{R}^n$.

Nesterov's method

To summarize: if a problem can be written according to this min-max model, the algorithm works as follows. . .

- **Regularization**. Add strongly convex penalty inside the min-max representation to produce an ϵ-approximation of f with Lipschitz continuous gradient (generalized Moreau-Yosida regularization step, see Lemaréchal & Sagastizábal (1997) for example).

- **Optimal first order minimization**. Use optimal first order scheme for Lipschitz continuous functions detailed in Nesterov (1983) to the solve the regularized problem.

Caveat: Only efficient if the subproblems involved in these steps can be solved explicitly or very efficiently. . .

Nesterov's method

- The min-max model makes this an ideal candidate for **robust optimization**
- For fixed problem size, the number of iterations required to get an ϵ solution is given by
$$O\left(\frac{1}{\epsilon}\right)$$
compared to $O\left(\frac{1}{\epsilon^2}\right)$ for generic first-order methods.
- Each iteration has low memory requirements.
- Change in **granularity** of the solver: larger number of cheaper iterations.

Matrix exponential

- The key step at each iteration is computing the gradient:

$$\nabla f_\mu(x) := \left(\mathbf{Tr}\exp\left(\frac{Ax-b}{\mu}\right)\right)^{-1}\exp\left(\frac{Ax-b}{\mu}\right)$$

- This amounts to a **matrix exponential** computation.

- Classic problem. See "Nineteen dubious ways to compute the exponential of a matrix" by Moler & Van Loan (2003).

- In general, Padé approximation techniques are the classic solution. We can do better here because of the matrix structure.

Main Result

When minimizing a function with Lipschitz-continuous gradient using the method in Nesterov (1983), an **approximate gradient** is sufficient to get the $O(1/\epsilon)$ convergence rate:

If the function and gradient approximations satisfy:

$$|f(x) - \tilde{f}(x)| \le \delta \quad \text{and} \quad |\langle \tilde{\nabla} f(x) - \nabla f(x), y\rangle| \le \delta \quad x, y \in Q_1,$$

we have:

$$f(x_k) - f(x^\star) \le \frac{L d(x^\star)}{(k+1)(k+2)\sigma} + 10\delta$$

where L, $d(x^\star)$ and σ are problem constants.

Benefits

- Because the eigenvalues of the gradient matrix decrease exponentially fast, only a few eigenvalues are necessary to compute the gradient with the required precision.

- How few? Pick $X \in \mathbf{S}_n$ with coefs $\mathcal{N}(0, \sigma^2/n)$. Wigner's semicircle law: eigenvalues of X are asy. dist. according to:

$$p(x) = \frac{1}{2\pi\sigma^2}\sqrt{4\sigma^2 - x^2},$$

  in the limit, the proportion of eigenvalues required is given by:

$$P_\lambda \triangleq P\left(e^{\frac{\lambda_i(X) - \lambda^{\max}(X)}{\mu}} \leq \gamma\right) = \int_{-2\sigma}^{2\sigma + \epsilon\frac{\log\gamma}{\log n}} \frac{1}{2\pi\sigma^2}\sqrt{4\sigma^2 - x^2}dx.$$

- With $n = 5000$, $\delta = 10^{-6}$ and $\epsilon = 10^{-2}$, we get $nP_\lambda = 2.3$ eigs.

Numerical performance

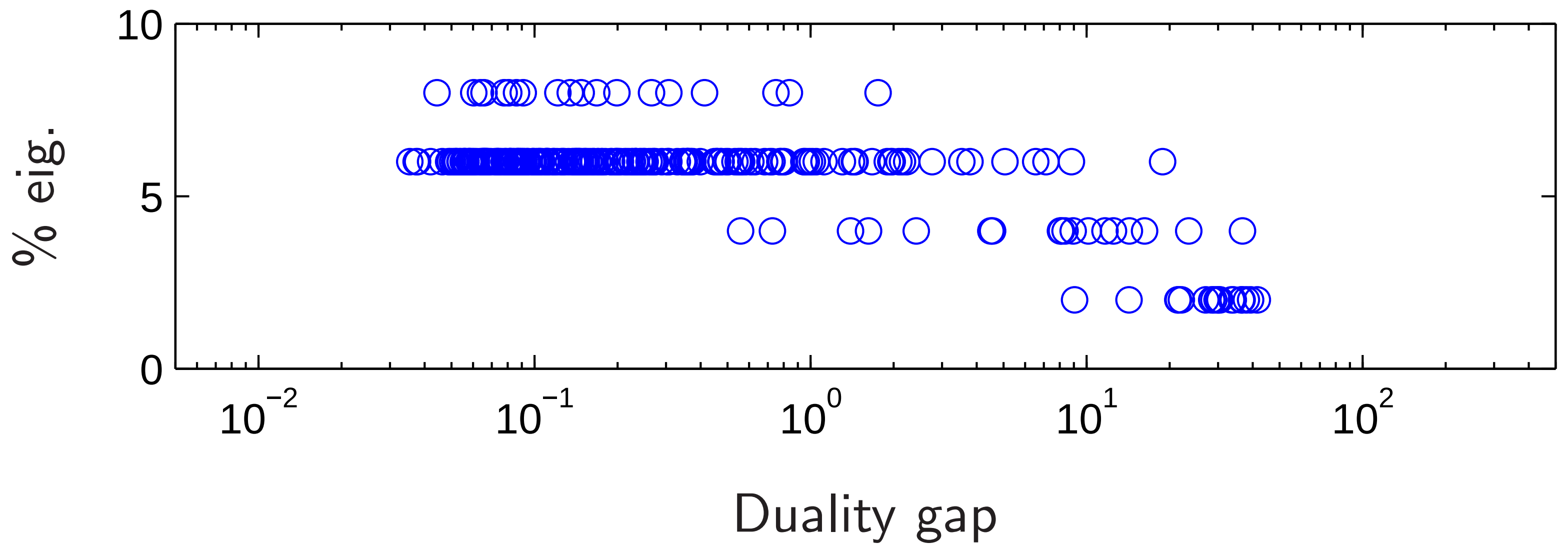


Figure 1: Percentage of eigenvalues required versus duality gap on random max. eigenvalue minimization problems.

Numerical performance

- Consider the following sparse PCA relaxation, from d'Aspremont, El Ghaoui, Jordan & Lanckriet (2005):

$$\begin{array}{ll} \text{minimize} & \lambda^{\max}(C+U) \\ \text{subject to} & |U_{ij}| \leq \rho, \quad i,j=1,\ldots,n, \end{array}$$

- Use ARPACK to compute eigenvalues (sparse eig. package).

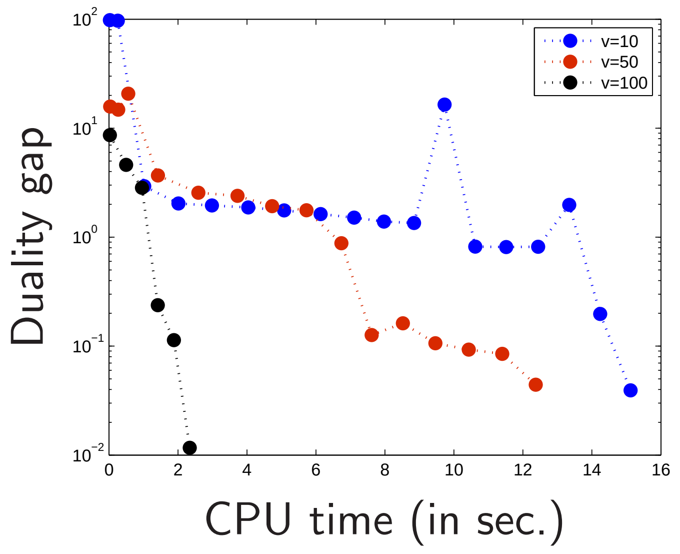
- Generate a 100×100 matrix U with uniformly distributed coefficients.

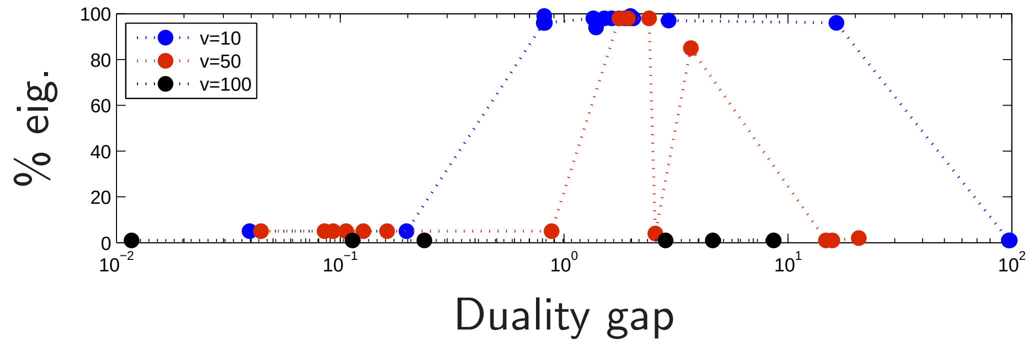
- Let $e \in \mathbf{R}^{100}$ be a sparse vector with:

$$e = (1,0,1,0,1,0,1,0,1,0,0,0,\ldots)$$

and form a test matrix $A = U^T U + vee^T$, where v is a signal-to-noise ratio.

Numerical performance





Left: Duality gap versus CPU time for various values of the signal to noise ratio v. *Right:* Percentage of eigenvalues required versus duality gap for various values of the signal to noise ratio v.

Conclusion

- Smooth first-order minimization with approximate gradient.
- An order of magnitude faster on semidefinite optimization problems.
- Link between problem structure and number of eigs required hard to establish. . .

References

d'Aspremont, A., El Ghaoui, L., Jordan, M. & Lanckriet, G. R. G. (2005), 'A direct formulation for sparse PCA using semidefinite programming', *Advances in Neural Information Processing Systems* **17**, 41–48.

Lemaréchal, C. & Sagastizábal, C. (1997), 'Practical aspects of the Moreau-Yosida regularization: theoretical preliminaries', *SIAM Journal on Optimization* **7**(2), 367–385.

Moler, C. & Van Loan, C. (2003), 'Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later', *SIAM Review* **45**(1), 3–49.

Nesterov, Y. (1983), 'A method of solving a convex programming problem with convergence rate $O(1/k^2)$', *Soviet Mathematics Doklady* **27**(2), 372–376.

Nesterov, Y. (2005), 'Smooth minimization of nonsmooth functions', *Mathematical Programming, Series A* **103**, 127–152.