First-Order Methods for Fast Eigenvalue Optimization

Guillermo Andrés Angeris John Merriman Sholar

June 12, 2018

Abstract

The problem of eigenvalue optimization appears in several areas: from inverse design, to control, to inference problems on continuous spaces. One instance, motivated by spectral element methods, produces a particular overlapping-block-diagonal sparsity pattern which is handled poorly by most modern SDP solvers. We propose two fast, first-order optimization methods for problems of this form, one of which consistently outperforms both SCS and CVXOPT over a large range of problem parameters.

1 Motivation

Consider the following problem, which comes up in several different contexts (as an inverse-design problem for example). Let's say we wish to manufacture a quantum, one-dimensional device that maximizes the ground state energy of a single, non-interacting particle of mass m within the device. We provide a basis of potentials for this problem $\{V^i\}$, where $V^i : \mathbf{R} \to \mathbf{R}$, such that all potentials of interest for the design can be written as a convex combination of these potentials. Then, the resulting Schrödinger PDE with Hermitian form is, for some potential $V(x) = \sum_i \alpha_i V^i(x)$ with $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$,

$$(\mathcal{H}(\alpha)\psi)(x) \equiv \left(-\frac{\hbar^2}{2m}\partial_x^2 + \sum_i \alpha_i V^i(x)\right)\psi(x) = E_0\psi(x).$$

Here, $\mathcal{H}(\alpha)$ is known as the Hamiltonian (defined as the linear operator above, which is affine over α) and E_0 is the smallest eigenvalue which satisfies the PDE, which we wish to maximize. ψ is the corresponding wavefunction of the solution, but the specifics of ψ will not be used further.

We can phrase this problem of minimizing the energy as the following optimization problem over a functional space,

$$\begin{aligned} \underset{\alpha, \psi, E_0}{\text{maximize}} \quad & E_0 \\ \text{subject to} \quad & \mathcal{H}(\alpha)\psi = E_0\psi, \\ & \sum_i \alpha_i = 1, \\ & \alpha_i \geq 0 \quad \forall i. \end{aligned}$$

Noting that, in general, the variables ψ and E_0 are completely determined by given values of α_i, so finding an optimal set of α_i is enough.

In this form, it is not clear the problem is convex, but (in a similar way to finite spaces), let H be the Hilbert space in question, with $\mathbb{I} : H \to H$ being the identity operator and, for some linear operator $A : H \to H$, we define $A \succeq 0 \iff \langle \phi, A\phi \rangle \geq 0, \forall \phi \in H$, which generates a cone.

We can then rewrite the problem by changing the equality $\mathcal{H}(\alpha)\psi = E_0\psi$ to the conic inequality $\mathcal{H}(\alpha) \succeq E_0\mathbb{I}$, which yields an optimization problem over a cone (it remains to be verified that $E_0 = \min_{\phi \in H} \langle \phi, \mathcal{H}(\alpha)\phi \rangle / \langle \phi, \phi \rangle$, but this is a standard result).

This is essentially an SDP over the space of linear operators mapping $H \to H$. Consider a finite discretization scheme $\partial_x^2 \mapsto L$, $V^i \mapsto D^i$, $\mathbb{I} \mapsto \bar{I}$ with $D^i, \bar{I}, L \in \mathbf{S}^m$ and $D^i, \bar{I}$ diagonal. We can rewrite the problem as a usual SDP:

$$\begin{aligned} \underset{\alpha, E_0}{\text{maximize}} \quad & E_0 \\ \text{subject to} \quad & \left(-\frac{\hbar^2}{2m} L + \sum_i \alpha_i D^i \right) \succeq E_0 \bar{I}, \\ & \sum_i \alpha_i = 1, \\ & \alpha_i \geq 0 \ \ \forall i, \end{aligned}$$

where we have plugged the above discretizations into the original definition of $\mathcal{H}$.

The particular structure of the matrices L, $\bar{I}$ and D^i generated by some discretization schemes makes this problem computationally feasible to optimize with the large domain sizes necessary for these PDEs to be accurate.

Note that we can view the first constraint to be a cone constraint on a linear operator which depends affinely on α, call this discretized operator $\mathscr{L} : \mathbf{R}^n \to \mathbf{S}^m$, while the rest of the constraints can be written as affine equalities with a nonnegativity constraint (roughly the same as the LP standard form). This results in the following problem:

$$\begin{aligned} \underset{\alpha, E_0}{\text{maximize}} \quad & E_0 \\ \text{subject to} \quad & \mathscr{L}(\alpha) \succeq E_0 \bar{I}, \\ & C\alpha = d, \\ & \alpha_i \geq 0 \ \ \forall i, \end{aligned} \tag{1}$$

for some $C \in \mathbf{R}^{r \times n}$ and $d \in \mathbf{R}^r$. This is the general problem we consider, with $\mathscr{L}(\alpha)$ having the structure of interest for every α.

1.1 Spectral Element Methods and Laplacian Construction

A *spectral element method* (SEM) [CHQZ06] is a finite-element method for solving PDEs which makes use of high-degree polynomials to approximate functions on a compact domain. The idea is to construct a quadrature rule such that all polynomials of lesser degree have

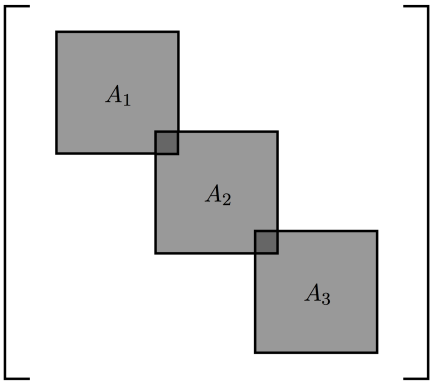

Figure 1: A simple example of the overlapping diagonal elements with three blocks. The darker box indicates a single overlapping element, *e.g.*, $(A_1)_{bb} = (A_2)_{11}$ if $A_1 \in \mathbf{R}^{b\times b}$.

exact integrals when sampled at a particular set of points and use these polynomials as a basis to approximate functions.

In general, these SEM-discretized problems—while sparse—are usually slow to solve in modern optimization packages for even a modest number of points in the discretization due to their structure. In particular, the matrix generated by the spectral elements method is essentially a block-diagonal matrix with small blocks, each of which overlap with the previous block by exactly one element (see Figure 1), so no completely trivial decomposition can be applied. [1] We construct a solver for problems similar to problem 1 where $\mathscr{L}(\alpha)$ has nice sparsity structure (*e.g.*, as given in Figure 1) over all α.

2 Approaches

Let $\mathscr{L} : \mathbf{R}^n \to \mathbf{S}^m$ be affine and have nice sparse structure (*e.g.*, as described in Figure 1), and let $x \in \mathbf{R}^n, t \in \mathbf{R}$, then the problem of interest can be written as the dual of a standard-form SDP,

$$\begin{array}{ll} \underset{x,\,t}{\text{minimize}} & -t \\ \text{subject to} & \mathscr{L}(x) \preceq tI, \\ & Cx = d, \\ & x \succeq 0. \end{array} \tag{2}$$

Here, $\mathscr{L}(x) = A_0 + \sum_i x_i A_i$, where all A_i have the given nice sparsity structure. The first inequality is with respect to the semidefinite cone, while the latter is with respect to the positive orthant.

Problem 2 can be solved in several ways. We present two possibilities for first-order solvers: alternating projections and a first-order interior-point methods.

[1] Further, it's important to note that this matrix sparsity pattern is a special case of chordal sparsity [VA+15], which, while fast to solve in general, does not take advantage of the further structure which is obvious from this problem.

2.1 Alternating Projections

As in [ZFP+17] define $E_i \in \mathbf{R}^{b \times m}$ to be the 'selection matrix' given by

$$E_i = \underbrace{\left[\underbrace{0_{b\times(b-1)} \quad \cdots \quad 0_{b\times(b-1)}}_{i-1} \quad I_b \quad 0_{b\times(b-1)} \quad \cdots \quad 0_{b\times(b-1)}\right]}_{B}.$$

where B is the number of sub-blocks of $\mathscr{L}$ and b is their dimension.

This allows us to rewrite (2) above to

$$\begin{aligned} \underset{x,\,t}{\text{minimize}} \quad & -t \\ \text{subject to} \quad & \mathscr{L}(x) - tI = \sum_i E_i Z_i E_i^T, \\ & Cx = d, \\ & x \succeq 0, \\ & Z_j \succeq 0,\ j \in \{1, 2, \ldots, B\}, \end{aligned}$$

Formulating the Lagrangian of this problem yield the following constraints for an optimal solution. From strong duality we have

$$d^T \eta - \operatorname{tr}(\Lambda A_0) - t = 0,$$

Primal feasibility implies,

$$\mathscr{L}(x) = \sum_j E_j Z_j E_j^T, \qquad x \succeq 0, \qquad Z_j \succeq 0 \ \ \forall j, \qquad Cx = d,$$

and finally from dual feasilibity we have

$$c_i + \operatorname{tr}(\Lambda A_i) + (C^T \eta)_i = \xi_i \ \ \forall i, \qquad -E_j^T \Lambda E_j \succeq 0 \ \ \forall j, \qquad \xi_i \geq 0 \ \ \forall i,$$

where η, Λ, ξ are dual variables corresponding to various constraints.

Note that these conditions are sufficient for optimality since this problem is always feasible with non-empty interior, therefore satisfying Slater's condition. Furthermore, all (in)equalities are projections either onto affine subspaces—and are therefore efficiently computable by caching the QR factorization of various matrices—or projections into the PSD or nonnegative orthant codes. In the latter case, the Z_j matrices (of size $b \times b$) are much smaller than the original $\mathscr{L}$ matrix (of size $m \times m$, with $m \gg b$) which makes these smaller projections much more efficient than projecting the complete matrix into the cone.

2.2 Interior-Point Methods

For interior-point methods, rewriting the problem using a barrier method is straightforward

$$\begin{aligned} \underset{x,\,t}{\text{minimize}} \quad & -t - \mu \left(\log\det(\mathscr{L}(x) - tI) + \sum_i \log(x_i) \right) \\ \text{subject to} \quad & Cx = d. \end{aligned}$$

As before, we assume that there exists some point x^0 with $Cx^0 = d$ and $x^0 \succ 0$, and, by picking $t^0 = \lambda_1(\mathscr{L}(x^0)) - \varepsilon$, for any $\varepsilon > 0$, then (t^0, x^0) lies in the interior of the domain (we compute such an x^0 in the pre-solve phase of the optimization). Usually, one would use a second-order method to optimize this problem for fixed μ, but computing the Hessian is expensive in this case. In order to avoid this, we use a projected gradient descent method to optimize the above function, subject to the given affine constraint.

In particular, since we are guaranteed that (t^0, x^0) lies strictly in the interior of the domain (and $Cx^0 = d$), it is always possible to pick a small enough step-size such that we never step outside of the valid domain of the barrier—therefore, all of our iterates remain strictly feasible with an appropriately small step size. As before, we can also cache the QR factorization of C, making our projections cost only a matrix-vector multiplication.

First, the gradient of the objective function (f, for notational convenience), with respect to each x_i is

$$\frac{\partial f}{\partial x_i} = -\mu \left(\operatorname{tr}\left((\mathscr{L}(x) - tI)^{-1} A_i \right) + \frac{1}{x_i} \right),$$

and with respect to t,

$$\frac{\partial f}{\partial t} = -1 + \mu \operatorname{tr}\left((\mathscr{L}(x) - tI)^{-1} \right).$$

Note that computing the values of both derivatives has the same cost as a matrix-matrix multiplication, once the sparse Cholesky factorization of $\mathscr{L}(x) - tI$ is computed, while computing $\log\det(\mathscr{L}(x) - tI)$ is essentially free after the factorization is known.

3 Results

3.1 Alternating Projections

We implement the method described in section 2.1 for the simple eigenvalue optimization problem described in equation 2 with $C = 0$ and $d = 0$. In particular, we construct an A_0 having the approximate-block-diagonal structure shown in Figure 1, where A_0 has B sub-blocks, each of dimension $b \times b$, and the nonzero elements of A are sampled $A_{ij} \overset{i.i.d.}{\sim} N(0, 1)$.

We conduct two tests, one in which we fix $B = 10$ and vary b, and the other in which we fix $b = 10$ and vary B. We compare time to convergence within 1% of the optimum for our algorithm and CVXPY [DB16] using the SCS solver [OCPB17]. The results of these tests are presented in Figure 2.

Note importantly that while SCS consistently outperforms our algorithm when holding $B = 10$ fixed and varying b, we begin to outperform SCS when holding $b = 10$ fixed and taking $B \geq 30$.

3.2 Interior-Point via Projected Gradient Descent

We implement the method described in section 2.2 for the general optimization problem described in equation 2. We construct A_0 as in section 3.1, and take $C = \mathbf{1}^T \in \mathbf{R}^{1\times n}$ and $d = 1 \in \mathbf{R}$.

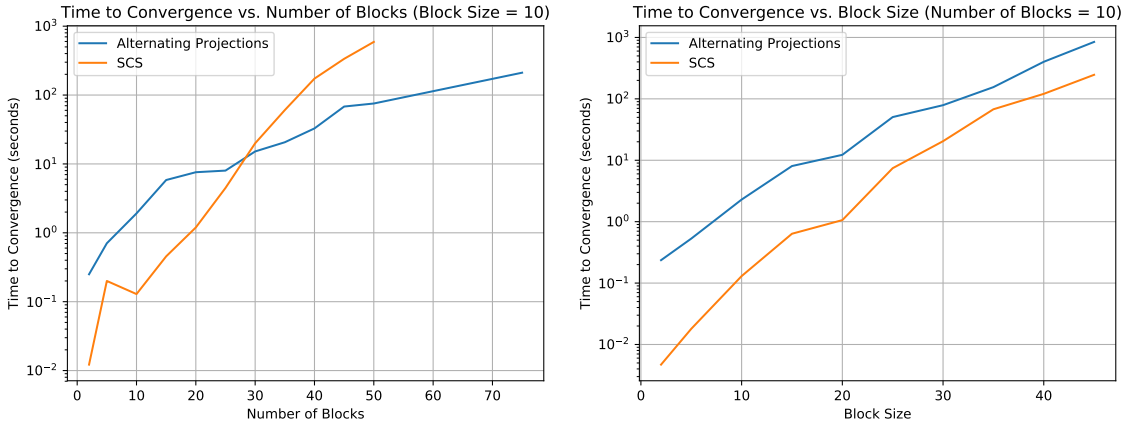

Figure 2: Comparison of Alternating Projections (2.1) to CVXPY (SCS)

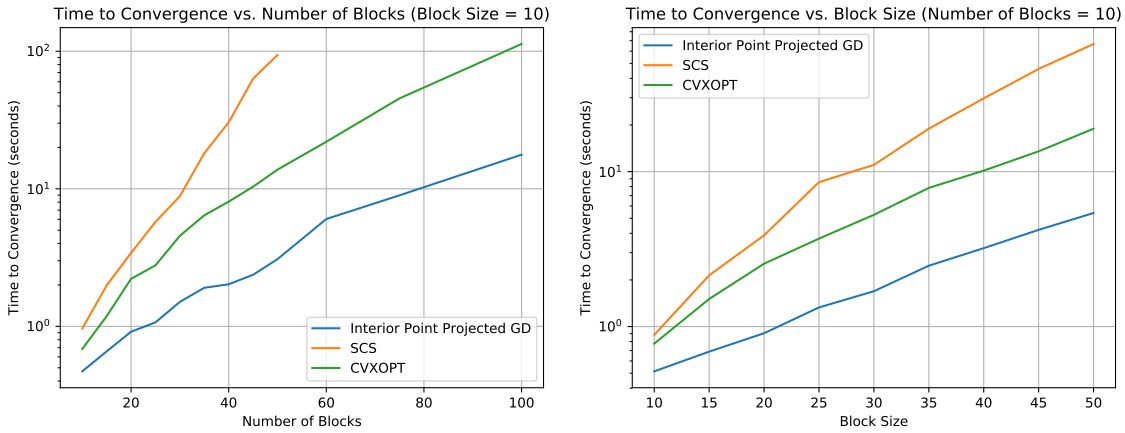

Figure 3: Comparison of Interior-Point Projected GD (2.2) to CVXPY (SCS and CVXOPT)

Our tests mimic those described in section 3.1, taking the block sized b fixed and varying the number of blocks B, and vice versa. Notably, while in section 3.1 we required convergence within a factor of 10^{-2} of the optimal value, here we require convergence to within a factor of 10^{-6} of optimal. The results of these tests are presented in Figure 3.

Note importantly that our methods consistently outperform both SCS and CVXOPT [ADV13] and that the gap in all cases either stays constant or increases.

4 Conclusion and Further Thoughts

In sections 2 and 3, we presented and evaluated two fast, first-order optimization methods for problems of the form presented in section 1. Most notably, our interior-point projected gradient descent method consistently outperforms both SCS and CVXOPT, suggesting that this solver could be used as a fast, specialized solver for problems of this form. We suspect that solving this sub-problem problem via some SLSQP-like method might yield better results (by approximating the Hessian with a BFGS-like update) than via GD—this is left as an open question for future research.

References

[ADV13] Martin S Andersen, Joachim Dahl, and Lieven Vandenberghe. Cvxopt: A python package for convex optimization, version 1.1. 6. *Available at cvxopt. org*, 54, 2013.

[CHQZ06] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods Fundamentals in Single Domains*. 2006.

[DB16] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[OCPB17] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. SCS: Splitting conic solver, version 2.0.2. `https://github.com/cvxgrp/scs`, November 2017.

[VA+15] Lieven Vandenberghe, Martin S Andersen, et al. Chordal graphs and semidefinite optimization. *Foundations and Trends® in Optimization*, 1(4):241–433, 2015.

[ZFP+17] Yang Zheng, Giovanni Fantuzzi, Antonis Papachristodoulou, Paul Goulart, and Andrew Wynn. Fast admm for semidefinite programs with chordal sparsity. In *American Control Conference (ACC), 2017*, pages 3335–3340. IEEE, 2017.

Appendix

Acknowledgements

The authors wish to thank John Duchi and Yair Carmon for their contributions to this project.

Source Code

The source code for this project, and instructions to run the tests, can be found at `https://github.com/guillean/LEigOpt`.

Note for EE364B Instructors

An instructor post on Piazza noted that "For 'new modeling approach' and 'theory' projects we will require only basic documentation/code readability, whereas for 'open source software'-type projects code and documentation quality will play a significant role in the project evaluation."

Our project, at first glance, may seem to fall in the intersection of these two projects (developing the theory for a new optimization technique which could, with time, become an open-source package for solving problems of this form). However, our intention with this project was to develop the theory behind this optimization technique. As such, our code was only intended as a simple proof of concept, rather than a full, open-source library. We request that the focus of the grading be on the theoretical results we present, rather than on the quality of our code (though, to be fair, the code isn't terrible...).