Applications of Performance Estimation

Pranav Reddy

November 23, 2023

UC San Diego

Overview

Refresh

Last time, we introduced a variety of conditions for interpolating a convex function through a set of points with given (sub)gradients.

Refresh

Last time, we introduced a variety of conditions for interpolating a convex function through a set of points with given (sub)gradients. Now, we can look at applying our knowledge to a few different algorithms.

Setting

Given a matrix $A \in \mathbb{R}^{n\times n}(\mathbb{C}^{n\times n})$, we say that A is symmetric (Hermitian) if $A = A^\top(A^H)$. Hermitian matrices have a very important characterization, known as the Spectral Theorem.

Theorem (Spectral Theorem)

Suppose we have a matrix $A \in \mathbb{C}^{n\times n}$. Then, A is symmetric if and only if it can be written as $A = PDP^H$, where P is unitary and D is a diagonal matrix with the eigenvalues of A on the diagonal. Moreover, the eigenvalues of A are real, and if A is real then P is real.

Proof of the Spectral Theorem

Proof.

One direction is easy to show. If $A = PDP^H$, then $A^H = (PDP^H)^H = PDP^H$, and if P is real then $P^H = P^\top$, so A is real.

Proof of the Spectral Theorem

Proof.

One direction is easy to show. If $A = PDP^H$, then $A^H = (PDP^H)^H = PDP^H$, and if P is real then $P^H = P^\top$, so A is real.

Now, suppose that A is Hermitian. Then, let λ be an eigenvalue of A and v be an eigenvector associated with λ. Then,

$$\lambda\langle v, v\rangle = \langle \lambda v, v\rangle = \langle Av, v\rangle = \langle v, A^H v\rangle = \langle v, Av\rangle = \langle v, \lambda v\rangle$$

$$= \overline{\lambda}\langle v, v\rangle.$$

Therefore, $\lambda = \overline{\lambda}$, so λ is real. □

Proof of the Spectral Theorem (Cont.)

Proof.

Now, we note that the eigenspace $\mathrm{span}\{v\} = V_\lambda$ is A-invariant. Therefore, its orthogonal complement, $V_\lambda^\perp$ is also A-invariant, since A is Hermitian. Viewing A as a linear operator, we can see that if we induct on the dimension of the ambient vector space, we have an orthonormal basis of eigenvectors of $V_\lambda^\perp$, and joining this with $\{v\}$ gives an orthonormal basis. □

Semidefinite Matrices

The spectral theorem leads us to the fundamental object of semidefinite programming: symmetric positive semidefinite matrices. We denote the vector space of real symmetric $n \times n$ matrices by $\mathbb{S}^n$.

Semidefinite Matrices

The spectral theorem leads us to the fundamental object of semidefinite programming: symmetric positive semidefinite matrices. We denote the vector space of real symmetric $n \times n$ matrices by $\mathbb{S}^n$.

Definition (Positive Semidefinite Matrix)

A matrix $A \in \mathbb{S}^n$ is **positive semidefinite** if for any nonzero vector $v \in \mathbb{R}^n$:

$$\langle Av, Av \rangle \geq 0.$$

If the inequality is strict, then A is **positive definite**. We denote the set of positive semidefinite matrices by $\mathbb{S}^n_+$, and the set of positive definite matrices by $\mathbb{S}^n_{++}$.

Semidefinite Program

Definition

A **semidefinite program** is a problem of the form

$$\begin{aligned} p^\star = \min_{X \in \mathbb{S}^n} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_k, X \rangle = b_k \quad k = 1, \ldots, m \\ & X \succeq 0. \end{aligned}$$

This is known as the **primal** problem.

Semidefinite Program

Definition

A **semidefinite program** is a problem of the form

$$\begin{aligned} p^\star = \min_{X \in \mathbb{S}^n} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_k, X \rangle = b_k \quad k = 1, \ldots, m \\ & X \succeq 0. \end{aligned}$$

This is known as the **primal** problem. The **dual** semidefinite program is

$$\begin{aligned} d^\star = \max_{y \in \mathbb{R}^m} \quad & \langle b, y \rangle \\ \text{subject to} \quad & C - \sum_{i=1}^{m} y_i A_i \succeq 0. \end{aligned}$$

Important Properties of Semidefinite Programs

We have the following:

$$\begin{aligned}
\langle C, X\rangle - \langle b, y\rangle &= \langle C, X\rangle - \sum_{i=1}^{m} b_i y_i \\
&= \langle C, X\rangle - \sum_{i=1}^{m} b_i y_i \\
&= \langle C, X\rangle - \sum_{i=1}^{m} y_i \langle A_i, X\rangle \\
&= \left\langle C - \sum_{i=1}^{m} y_i A_i, X \right\rangle \\
&\geq 0
\end{aligned}$$

Important Properties of Semidefinite Programs

We have the following:

$$\begin{aligned}
\langle C, X\rangle - \langle b, y\rangle &= \langle C, X\rangle - \sum_{i=1}^{m} b_i y_i \\
&= \langle C, X\rangle - \sum_{i=1}^{m} b_i y_i \\
&= \langle C, X\rangle - \sum_{i=1}^{m} y_i \langle A_i, X\rangle \\
&= \left\langle C - \sum_{i=1}^{m} y_i A_i, X \right\rangle \\
&\geq 0
\end{aligned}$$

So $p^\star \geq d^\star$.

Performance Estimation Problem

We are interested in the worst-case performance of first-order methods. Given a method $\mathcal{M}$, and performance measure $\mathcal{E}$, a class of functions $\mathcal{F}$, and some intial conditions $\mathcal{C}$, we are interested in the worst performance of $\mathcal{M}$:

$$\begin{aligned} \sup_{f \in \mathcal{F}} \quad & \mathcal{E}\left(\{x_i, f_i, g_i\}_{i=1,\dots,N,\star}\right) \\ \text{such that} \quad & f \in \mathcal{F} \\ & x_\star \text{ is optimal} \\ & \{x_i, f_i, g_i\}_{i=1,\dots,N,\star} \text{ are generated by } \mathcal{M} \\ & (x_0, f_0, g_0) \text{ satisfy the initial conditions } \mathcal{C} \end{aligned}$$

Performance Estimation Problem

We are interested in the worst-case performance of first-order methods. Given a method $\mathcal{M}$, and performance measure $\mathcal{E}$, a class of functions $\mathcal{F}$, and some intial conditions $\mathcal{C}$, we are interested in the worst performance of $\mathcal{M}$:

$$
\begin{aligned}
& \sup_{f \in \mathcal{F}} && \mathcal{E}\left(\{x_i, f_i, g_i\}_{i=1,\ldots,N,\star}\right) \\
& \text{such that} && f \in \mathcal{F} \\
& && x_\star \text{ is optimal} \\
& && \{x_i, f_i, g_i\}_{i=1,\ldots,N,\star} \text{ are generated by } \mathcal{M} \\
& && (x_0, f_0, g_0) \text{ satisfy the initial conditions } \mathcal{C}
\end{aligned}
$$

We will show that a large class of optimization problems can be cast in this form, including standard (sub)gradient descent, proximal point, and even some constrained optimization problems.

SDP Reformulation

Our goal is to show that the PEP, for many classes of functions and methods, can we written in the form

$$\begin{aligned} \sup_{F_N \in \mathbb{R}^{1\times(N+2)}, G_N \in \mathbb{S}^{2(N+1)}} \quad & c^\top F_N + \langle C, G_N \rangle \\ \text{subject to} \quad & a_i + b_i^\top F_N + \langle D_i, G_N \rangle \leq 0 \quad i = 1, \ldots k \\ & G_n \succeq 0 \end{aligned}$$

Fixed-Step Linear First-Order Method

Definition

A **fixed-step linear first-order method** (FSLFOM) is a method which produces iterates as the solution to

$$t_{i,i}x_i + h_{i,i}g_i = \sum_{j=0}^{i-1} \left(t_{i,j}x_j + h_{i,j}g_j\right),$$

where the step size coefficients $t_{i,j}$ and $h_{i,j}$ are fixed.

Fixed-Step Linear First-Order Method

Definition

A **fixed-step linear first-order method** (FSLFOM) is a method which produces iterates as the solution to

$$t_{i,i}x_i + h_{i,i}g_i = \sum_{j=0}^{i-1} \left(t_{i,j}x_j + h_{i,j}g_j\right),$$

where the step size coefficients $t_{i,j}$ and $h_{i,j}$ are fixed.

We note here that this class of methods is exactly those which can be written as

$$x_i = \arg\min_{x \in \mathbb{R}^n} \left\{ \frac{t_{i,i}}{2}\|x\|^2 + h_{i,i}F(x) - \left\langle \sum_{j=0}^{i-1} \left(t_{i,j}x_j + h_{i,j}\nabla F(x_j)\right), x \right\rangle \right\}$$

SDP Reformulation of FSLFOM

Our goal is to reformulate a given FSLFOM as the constraints of a semidefinite program.

SDP Reformulation of FSLFOM

Our goal is to reformulate a given FSLFOM as the constraints of a semidefinite program.

We define the matrices $P_N \in \mathbb{R}^{d \times 2(N+1)}$ and $F_N \in \mathbb{R}^{1 \times (N+2)}$ as

$$P_N = [x_0 \dots x_N \; x_\star \mid g_0 \dots g_N \; g_\star]$$

$$F_N = [f_0 \dots f_N \; f_\star].$$

SDP Reformulation of FSLFOM

Our goal is to reformulate a given FSLFOM as the constraints of a semidefinite program.
We define the matrices $P_N \in \mathbb{R}^{d\times 2(N+1)}$ and $F_N \in \mathbb{R}^{1\times(N+2)}$ as

$$P_N = [x_0 \dots x_N \; x_\star \mid g_0 \dots g_N \; g_\star]$$

$$F_N = [f_0 \dots f_N \; f_\star].$$

Using this, we define $G_N \in \mathbb{S}^{2(N+1)}$ by

$$G_n = P_N^\top P_N \succeq 0.$$

Note that rank $G_N \leq d$.

SDP Reformulation of FSLFOM (Cont.)

We can see that the definition of a FSLFOM is a system of linear equations that depends only on the coordinate and subgradients up to a given iterate. Therefore, we can write this as

$$P_n m_i = 0,$$

where m_i is some vector of coefficients and the coefficients corresponding to future coordinate and subgradient values are zero.

SDP Reformulation of FSLFOM (Cont.)

We can see that the definition of a FSLFOM is a system of linear equations that depends only on the coordinate and subgradients up to a given iterate. Therefore, we can write this as

$$P_n m_i = 0,$$

where m_i is some vector of coefficients and the coefficients corresponding to future coordinate and subgradient values are zero. Then,

$$P_n m_i = 0 \iff \|P_n m_i\|^2 = 0 \iff m_i^\top G_n m_i = 0,$$

so we have an equivalent semidefinite constraint.

SDP Reformulation of FSLFOM (Example)

For example, for simple gradient descent,

$$x_i = x_{i-1} - h\nabla F(x_{i-1}),$$

we can write this as

$$0 = x_{i-1} - h\nabla F(x_{i-1}) - x_i.$$

So m_i is simply the vector such that the ith and $(i-1)$th entry entry are 1, the $(N+i)$th entry is h, and every other entry is 0.

SDP Reformulation of FSLFOM (Example 2)

A more interesting example is the proximal point algorithm, which computes

$$x_i = \arg\min_x \left\{ h_i F(x) + \frac{1}{2}\|x - x_{i-1}\|^2 \right\}.$$

By first-order optimality conditions, this can be rewritten as

$$h_i \nabla F(x_i) + x_i - x_{i-1} = 0,$$

which is an implicit linear equation for x_i, so it can be fitted into the FSLFOM framework, and therefore can be represented by a semidefinite constraint.

PPA Classic Result

As an example, we will improve the classical result on convergence of the proximal point algorithm (PPA) using the PEP.

Theorem (Classical Result)

Let F be a convex function, and let $x_\star$ be a minimizer. If $\|x_0 - x_\star\| \leq R$, then after N steps of the PPA, we have

$$F(x_N) - F(x_\star) \leq \frac{R^2}{2\sum_{k=1}^{N} h_k}.$$

Tight Bound on PPA

Theorem (New Result)

In fact, we have

$$F(x_N) - F(x_\star) \leq \frac{R^2}{4\sum_{k=1}^{N} h_k},$$

and this bound is tight.

Proof of Tightness

Proof.

First, by consider the one-dimensional function

$$F(x) = \frac{R|x|}{2\sum_{k=1}^{N} h_k},$$

and the initial point $x = -R$.

Proof of Tightness

Proof.

First, by consider the one-dimensional function

$$F(x) = \frac{R|x|}{2\sum_{k=1}^{N} h_k},$$

and the initial point $x = -R$.
After N iterations of PPA, we have

$$x_N = x_0 + \sum_{k=1}^{N} h_k \frac{R}{2\sum_{k=1}^{N} h_k} = -\frac{R}{2}.$$

□

Proof of Tightness Continued

Proof.

We can see that

$$f(x_N) = \frac{R^2}{4\sum_{k=1}^{N} h_k},$$

and since $f(x_\star) = 0$, we have the desired equality. □

Proof of Upper Bound

First, we need to define our problem in terms of the PEP, so we must define $\mathcal{F}, \mathcal{E}, \mathcal{M}$, and $\mathcal{C}$:

- $\mathcal{E} = f(x_N) - f(x_\star)$
- $\mathcal{F} = \mathcal{F}_{0,\infty}$, closed proper convex functions
- $\mathcal{M} =$ proximal point algorithm
- $\mathcal{C} = \{\|x_0 - x_\star\| \leq R\}$

Additionally, we can assume without loss of generality that $F(x_\star) = 0$ and $x_\star = 0$.

Reformulating $\mathcal{E}$

We can see that $f(x_N) - f(x_\star)$ can be rewritten as a linear combination of the columms of the matrix F_N:

$$f(x_N) - f(x_\star) = (e_N - e_{N+1})^\top F_N = \langle e_N - e_{N+1}, F_N \rangle$$

and this is a semidefinite objective.

Reformulating $\mathcal{F}$ and $\mathcal{M}$

From the the previous presentation, we have that a set of point is $\mathcal{F}_{0,\infty}$-interpolable if and only if the $\binom{N}{2}$ linear inequalities

$$f_i - f_j - \langle g_j, x_i - x_j \rangle \geq 0 \quad \forall\ i, j \in \{1, \ldots, N, \star\}.$$

Reformulating $\mathcal{F}$ and $\mathcal{M}$

From the the previous presentation, we have that a set of point is $\mathcal{F}_{0,\infty}$-interpolable if and only if the $\binom{N}{2}$ linear inequalities

$$f_i - f_j - \langle g_j, x_i - x_j \rangle \geq 0 \quad \forall\ i, j \in \{1, \ldots, N, \star\}.$$

We will combine the conditions on $\mathcal{F}$ and $\mathcal{M}$ to simplify the PEP considerably. Using m_i and m_j such that $x_k = P_n m_k$, we have that

$$m_k = e_{N+1} - \sum_{i=1}^{k} h_i e_i,$$

with $m_0 = e_{N+1}$ and $m_\star = 0$.

Reformulating $\mathcal{F}$ and $\mathcal{M}$ (Cont.)

Define

$$A_{ij} = \frac{1}{2}\left(e_j(m_i - m_j)^\top + (m_i - m_j)e_j^\top\right)$$

where $e_\star = 0$. Then, the constraints given by $\mathcal{F}$ and $\mathcal{C}$ become

$$f_i - f_j + \langle A_{ij} G_N \rangle \leq 0 \quad \forall\ i, j \in \{1, \ldots, N, \star\}$$

Reformulating $\mathcal{F}$ and $\mathcal{M}$ (Cont.)

Define

$$A_{ij} = \frac{1}{2}\left(e_j(m_i - m_j)^\top + (m_i - m_j)e_j^\top\right)$$

where $e_\star = 0$. Then, the constraints given by $\mathcal{F}$ and $\mathcal{C}$ become

$$f_i - f_j + \langle A_{ij} G_N \rangle \leq 0 \quad \forall\ i, j \in \{1, \ldots, N, \star\}$$

This is all we need to fully reformulate the PEP to prove the better upper bound, since we have shown for the PPA:

$$x_k = x_{k-1} - h_k \nabla F(x_k).$$

Reformulated PEP

Using the previous inequalities, the PEP becomes

$$\begin{aligned}
\max_{F_N \in \mathbb{R}^{1\times(N+2)}, G_N \in \mathbb{S}^{2(N+1)}} \quad & \langle e_N - e_{N+1}, F_N \rangle \\
\text{subject to} \quad & \langle e_i - e_{j+1}, F_N \rangle + \langle A_{ij}, G_N \rangle \leq 0 \quad i, j = 1, \ldots, N, \star \\
& \text{rank } G_N \leq d \\
& \|x_0 - x_\star\| \leq R^2 \\
& G_n \succeq 0
\end{aligned}$$

Here, we make the additional assumption that $d \geq N + 2$ (although this is not necessary, see [1]).

Constructing an Upper Bound

We will take the dual of this problem to generate an upper bound on the PEP:

$$\begin{aligned} \min_{\lambda_{ij}\geq 0, \tau \geq 0} \quad & \tau R^2 \\ \text{subject to} \quad & e_N - \sum_i \sum_{j\neq i} (\lambda_{ij} - \lambda_{ji}) e_j = 0 \\ & \sum_i \sum_{j \neq i} \lambda_{ij} A_{ij} + \tau m_0 m_0^\top \succeq 0 \end{aligned}$$

Constructing an Upper Bound (Cont.)

We will choose the following multipliers to build a dual feasible solution:

$$\lambda_{i,i+1} = \frac{\sum_{k=1}^{i} h_k}{2\sum_{k=1}^{N} h_k - \sum_{k=1}^{i} h_k} \quad i = 1, \dots, N-1$$

$$\lambda_{\star,i} = \frac{2h_i \sum_{k=1}^{N} h_k}{(2\sum_{k=1}^{N} h_k - \sum_{k=1}^{i} h_k)(2\sum_{k=1}^{N} h_k - \sum_{k=1}^{i-1} h_k)} \quad i = 1, \dots, N$$

$$\tau = \frac{1}{4\sum_{k=1}^{N} h_k}$$

$$\lambda_{ij} = 0 \quad \text{otherwise}$$

Constructing an Upper Bound (Cont.)

We will choose the following multipliers to build a dual feasible solution:

$$\lambda_{i,i+1} = \frac{\sum_{k=1}^{i} h_k}{2\sum_{k=1}^{N} h_k - \sum_{k=1}^{i} h_k} \quad i = 1, \dots, N-1$$

$$\lambda_{\star,i} = \frac{2h_i \sum_{k=1}^{N} h_k}{(2\sum_{k=1}^{N} h_k - \sum_{k=1}^{i} h_k)(2\sum_{k=1}^{N} h_k - \sum_{k=1}^{i-1} h_k)} \quad i = 1, \dots, N$$

$$\tau = \frac{1}{4\sum_{k=1}^{N} h_k}$$

$$\lambda_{ij} = 0 \quad \text{otherwise}$$

Verifying the feasibility of this choice will not be done here. Therefore, $\frac{R^2}{4\sum_{k=1}^{N} h_k}$ is an upper bound on the primal problem, and therefore an upper bound on the performance of the PPA.

References

[1] Adrien B. Taylor, Julien M. Hendrickx, and François Glineur. Smooth strongly convex interpolation and exact worst-case performance of first-order methods, 2016.

[2] Adrien B. Taylor, Julien M. Hendrickx, and François Glineur. Exact worst-case performance of first-order methods for composite convex optimization. *SIAM Journal on Optimization*, 27(3):1283–1313, Jan 2017.