An Introduction to Semidefinite Programming

Brian Borchers
Department of Mathematics
New Mexico Tech
Socorro, NM 87801
borchers@nmt.edu

In fact, the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity.

R. Tyrrell Rockafellar (1993)

Why Convexity is Important

- An optimization problem is said to be convex if the set of feasible solutions is convex (every solution on the line segment between any two feasible solutions is also feasible) and the objective function to be minimized is convex.
- It's easy to show that any local minimum of a convex optimization problem is also a global minimum.
- Numerical methods for optimization are iterative methods that under appropriate conditions can be shown to converge to a locally optimal solution.
- If we have a convex minimization problem and an optimization algorithm that converges to a locally optimal solution, then we can be sure that we've found a globally optimal solution.

The SDP Problem

$$
\begin{array}{lrcl}
\max & \operatorname{tr}(CX) & & \\
(P) & A(X) & = & b \\
 & X & \succeq & 0
\end{array}
$$

where

$$
A(X) = \begin{bmatrix} \operatorname{tr}(A_1 X) \\ \operatorname{tr}(A_2 X) \\ \vdots \\ \operatorname{tr}(A_m X) \end{bmatrix}.
$$

The Dual Problem

$$
\begin{array}{lrcl}
\min & b^T y & & \\
(D) & A^T(y) - C & = & Z \\
 & Z & \succeq & 0
\end{array}
$$

where

$$
A^T(y) = \sum_{i=1}^{k} y_i A_i.
$$

Important Assumptions

- All of the matrices in the problem are symmetric matrices with real entries.
- It is assumed that the constraint matrices A_i are linearly independent.
- It is assumed that the problem has strictly feasible primal and dual solutions.
- Under these assumptions, (P) and (D) will have optimal values which are equal. Most algorithms for SDP provide both primal and dual optimal solutions.

Comparison with Linear Programming

- The objective function $\mathrm{tr}\,(CX)$ can be written as

  $$\mathrm{tr}\,(CX) = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} X_{ij}.$$

  Thus the objective is a linear function of the elements of X.

- By the same argument, the constraints $\mathrm{tr}(A_i X) = b_i$ are linear in the elements of the matrix X.

The Constraint $X \succeq 0$.

- The only thing nonlinear about our problem is the positive semidefinite constraint $X \succeq 0$.
- It turns out that the set of symmetric and positive semidefinite matrices is a closed convex cone.
- Thus our SDP is a convex optimization problem.

A Hierarchy Of Convex Optimization Problems

$$LP \subset CQP \subset SOCP \subset SDP \subset CP.$$

Polynomial time interior point methods for LP have been generalized to solve problems in this hierarchy up to the level of SDP.

Many other convex optimization problems can be formulated as structured convex optimization problems that fit into this classification scheme at some level.

Some nonconvex optimization problems have convex relaxations that can be fit into this scheme. The relaxations can be used to compute bounds on the nonconvex optimization problem.

An Example

Consider the minimization problem

$$\begin{aligned} \min \quad & \frac{(c^T x)^2}{d^T x} \\ Ax &\geq b \\ x &\geq 0 \end{aligned}$$

where we will assume that $d \geq 0$.

An Example

First, we use a standard trick to move the nonlinear objective function into the constraints.

$$
\begin{array}{lrcl}
\min & t & & \\
& \frac{(c^T x)^2}{d^T x} & \leq & t \\
& Ax & \geq & b \\
& x & \geq & 0
\end{array}
$$

Next, consider the matrix

$$
\begin{bmatrix} t & c^T x \\ c^T x & d^T x \end{bmatrix}
$$

An Example

- This two by two matrix is positive semidefinite iff its principal minors are nonnegative.
- In this case, we need $t \geq 0$, $d^T x \geq 0$ and $t(d^T x) - (c^T x)^2 \geq 0$.
- Since our original objective function is nonnegative, the constraint $t \geq 0$ will cause no problems.
- Since $d \geq 0$ and $x \geq 0$, $d^T x \geq 0$.
- Furthermore, the inequality $\frac{(c^T x)^2}{d^T x} \leq t$ can be rewritten as $t(d^T x) - (c^T x)^2 \geq 0$.
- Thus our matrix is positive semidefinite iff x and t satisfy the inequality.

An Example

- The constraint $Ax \geq b$ is equivalent to $Ax - b \geq 0$. Since a diagonal matrix is positive semidefinite iff its diagonal elements are nonnegative, this is equivalent to the constraint $\operatorname{diag}(Ax - b) \succeq 0$.
- The constraint $x \geq 0$ is equivalent to the constraint $\operatorname{diag}(x) \succeq 0$.
- All of these constraints can be combined into a single constraint as

$$\begin{bmatrix} \operatorname{diag}(Ax - b) & 0 & 0 & 0 \\ 0 & \operatorname{diag}(x) & 0 & 0 \\ 0 & 0 & t & c^T x \\ 0 & 0 & c^T x & d^T x \end{bmatrix} \succeq 0.$$

An Example

This constraint can be written as

$$A_0 + x_1 A_1 + \ldots + x_n A_n + t A_{n+1} \succeq 0$$

where

$$A_0 = \begin{bmatrix} \mathrm{diag}(-b) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

An Example

For $i = 1, 2, \ldots n$,

$$A_i = \begin{bmatrix} \mathrm{diag}(A(\cdot, i)) & 0 & 0 & 0 \\ 0 & \mathrm{diag}(e_i) & 0 & 0 \\ 0 & 0 & 0 & c_i \\ 0 & 0 & c_i & d_i \end{bmatrix}$$

and,

$$A_{n+1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

An Example

Our problem can now be written in the standard dual form as

$$\begin{array}{lrcl} \min & t & & \\ & A_0 + x_1 A_1 + \ldots + x_n A_n + t A_{n+1} & = & Z \\ & Z & \succeq & 0. \end{array}$$

The Schur Complement Theorem

A symmetric matrix

$$D = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$$

is positive semidefinite (PSD) iff the matrices C and $A - BC^\dagger B^T$ are PSD and $B(I - CC^\dagger) = 0$. Here $C^\dagger$ is the Moore-Penrose pseudoinverse of C.

In the special case where C is invertible, D is PSD iff C and $A - BC^{-1}B^T$ are PSD.

The Schur Complement Theorem

Note also that a symmetric permutation of a matrix $B = P^{-1}AP$ is a similarity transformation, so that B is PSD iff A is PSD. Thus we can permute the columns and rows of the matrix and retain PSD. In particular, this means that D is PSD iff

$$E = \begin{bmatrix} C & B^T \\ B & A \end{bmatrix}$$

is PSD.

Convex Quadratic Constraints

The convex quadratic constraint

$$(Ax - b)^T (Ax - b) - (c^T x + d) \leq 0$$

can be written as

$$\begin{bmatrix} c^T x + d & (Ax - b)^T \\ (Ax - b) & I \end{bmatrix} \succeq 0$$

or

$$\begin{bmatrix} I & Ax - b \\ (Ax - b)^T & c^T x + d \end{bmatrix} \succeq 0.$$

Constraints On Eigenvalues

Recall that if X has eigenvalues λ_1, λ_2, ..., λ_n, then $X - tI$ has eigenvalues $\lambda_1 - t$, $\lambda_2 - t$, ..., $\lambda_n - t$.

If we want the eigenvalues of $A(x) = x_1 A_1 + \ldots + x_m A_m$ to be greater than or equal to t, then we can use the constraint

$$A(x) - tI \succeq 0$$

Similarly, if we want the eigenvalues of $A(x)$ to be less than or equal to t, then we can use the constraint

$$-(A(x) - tI) \succeq 0.$$

If we want to minimize the maximum eigenvalue of $A(x)$,

$$\begin{array}{lrcl} \min & & t & \\ & tI - A(x) & \succeq & 0. \end{array}$$

Constraints On Eigenvalues

Consider the problem of maximizing

$$\|A(x)\|_2^2 = \lambda_{\max}(A(x)^T A(x)).$$

Using the Schur complement theorem and the above eigenvalue inequalities, this can be written as

$$\begin{array}{ll} \min & t \\ & \begin{bmatrix} tI & A(x) \\ A(x)^T & tI \end{bmatrix} \succeq 0 \end{array}$$

Constraints On Rank

In some cases, we would like to enforce the constraint

$$X = xx^T.$$

That is, X is required to be a rank one matrix. Unfortunately, this constraint is nonconvex and cannot be represented in SDP.

However, it is possible to relax the constraint to

$$X - xx^T \succeq 0$$

or

$$\begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0$$

or

$$X \succeq 0$$

The MAX–CUT Problem

Suppose that we are given a graph with n nodes and edge weights $w_{i,j}$. We would like to partition the nodes of the graph into two sets so that the weight of the edges between the two sets is maximized.

Let x_i be +1 if node i is in the first set, and -1 if node i is in the second set. We can then formulate the MAX–CUT problem as a ±1 quadratic programming problem.

$$\max \quad \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{w_{i,j}(1-x_i x_j)}{4}$$

$$x_i \quad = \quad \pm 1 \quad i = 1, 2, \ldots, n$$

The MAX–CUT Problem

We can write the constraints $x_i = \pm 1$ as

$$X = xx^T$$

and

$$X_{i,i} = 1 \quad i = 1, 2, \ldots, n$$

We can then relax $X = xx^T$ to $X \succeq 0$.

The MAX–CUT Problem

The resulting SDP relaxation of MAX–CUT is

$$
(MCR) \quad \begin{array}{rrcl} \max & \operatorname{tr}((-W/4)X) + c & & \\ & X_{i,i} & = & 1 \;\; i = 1, 2, \ldots, n \\ & X & \succeq & 0 \end{array}
$$

where

$$c = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j}/4$$

The MAX–CUT Problem

This SDP relaxation of MAX–CUT was introduced by Goemans and Williamson in 1995. They showed that:

- The optimal value of (MCR) is no more than 1.14 times the optimal value of the MAX–CUT problem.
- A randomized rounding heuristic can be used to generate a cut with expected value no less than 0.878 times the value of (MCR).
- A deterministic derandomized algorithm can be used to generate a cut with value no less than 0.878 times the value of (MCR).

Algorithms for SDP

The most commonly used algorithms for SDP are primal-dual interior point methods. These methods essentially apply Newton's method to a slightly perturbed version of the Karush-Kuhn-Tucker optimality conditions for the primal and dual SDP problems

Unfortunately, the primal-dual interior point method requires the solution of a dense linear system of m equations in m unknowns. For problems with tens of thousands of constraints, the storage required by the method makes the primal-dual method impractical.

There is considerable interest in first order methods that do not require $O(m^2)$ storage. Augmented Lagrangian methods offer some promise but are not yet as robust as the primal-dual interior point method.

Software for SDP

A number of software packages for solving SDP problems are available:

- CSDP. Borchers.
- DSDP. Benson.
- SeDuMi. Sturm.
- SDPA. Fujisawa, Kojima, Nakata, Yamashita.
- SDPSOL. Boyd.
- SDPT3. Todd, Toh, and Tutuncu.
- PENNON. Kocvara.

See http://plato.la.asu.edu/topics/problems/nlores.html for links to these packages.

Conclusions

- Semidefinite programming is a relatively new area of research, having been developed within the past 15 years.
- Interest in semidefinite programming has grown out of work with interior point methods for LP, eigenvalue optimization problems, and work in control theory.
- At this point, software for solving SDP's (and related problems) is readily available. Problems with hundreds or thousands of constraints can be readily solved, but larger problems are still difficult.
- Applications of SDP have arisen in many different areas. We can expect that more applications will appear over the next few years.