Interior-Point Algorithms for Semidefinite Programming

Yinyu Ye
Department of Management Science and Engineering
Stanford University
Stanford, CA 94305, U.S.A.

http://www.stanford.edu/~yyye
Chapter 6
Primal-Dual Potential Functions for SDP

\[
\text{(SDP)} \quad \text{Minimize} \quad C \cdot X \quad \text{subject to} \quad AX = b, \ X \succeq 0.
\]

\[
\text{(SDD)} \quad \text{Maximize} \quad b^T y \quad \text{subject to} \quad A^T y + S = C, \ S \succeq 0.
\]

\[
AX = \begin{pmatrix}
A_1 \cdot X \\ \vdots \\ A_m \cdot X
\end{pmatrix} \quad \text{and} \quad A^T y = \sum_{i=1}^{m} y_i A_i.
\]

For any \((X, y, S)) \in \text{int} \ F\), the SDP primal-dual potential function is given as

\[
\psi_{n+\rho}(X, S) := (n + \rho) \log(X \cdot S) - \log(\det(X) \cdot \det(S))
\]

\[
\psi_{n+\rho}(X, S) = \rho \log(X \cdot S) + \psi_n(X, S) \geq \rho \log(X \cdot S) + n \log n.
\]

Then, for \(\rho > 0\), \(\psi_{n+\rho}(X, S) \rightarrow -\infty\) implies that \(X \cdot S \rightarrow 0\). More precisely, we have

\[
X \cdot S \leq \exp\left(\frac{\psi_{n+\rho}(X, S) - n \log n}{\rho}\right).
\]
Once we have a pair \((X, y, S) \in \text{int} \mathcal{F}\) with \(\mu = S \circ X/n\), we compute direction vectors/matrices \((D_X, d_y, D_S)\) from the system of linear equations:

\[
D^{-1} D_X D^{-1} + D_S = \frac{X \circ S}{n+\rho} X^{-1} - S, \\
AD_X = 0, \\
-A^T d_y - D_S = 0,
\]

where

\[
D = X^{.5}(X^{.5} S X^{.5})^{- .5} X^{.5}.
\]

\[
X^+ = X + \alpha D_X, \quad y^+ = y + \alpha d_y, \quad S^+ = S + \alpha D_S.
\]

**Theorem 1** Let \(\rho \geq \sqrt{n}\). Then there is a step-size rule such that

\[
\psi_{n+\rho}(X^+, S^+) - \psi_{n+\rho}(X, S) \leq -0.15.
\]
Given \((X^0, y^0, S^0) \in \text{int } \mathcal{F}\). Set \(\rho \geq \sqrt{n}\) and \(k := 0\).

While \(\mathcal{S}^k \cdot X^k \geq \epsilon\) do

1. Set \((X, S) = (X^k, S^k)\) and \(\gamma = n/(n + \rho)\) and compute \((D_X, d_y, D_S)\) from (1).

2. Let \(X^{k+1} = X^k + \bar{\alpha}D_X, y^{k+1} = y^k + \bar{\alpha}d_y,\) and \(S^{k+1} = S^k + \bar{\alpha}D_S\), where
   \[
   \bar{\alpha} = \arg\min_{\alpha \geq 0} \psi_{n+\rho}(X^k + \alpha D_X, S^k + \alpha D_S).
   \]

3. Let \(k := k + 1\) and return to Step 1.

The result relies on the Logarithmic Approximation Lemma (second-order scaled Lipschitz condition):

**Lemma 1** If \(D \in \mathcal{S}^n\) and \(\|D\| \leq \alpha < 1\), then,

\[
I \cdot D) \geq \log \det(I + D) \geq I \cdot D - \frac{1}{2(1 - \alpha)}\|D\|^2.
\]
A key question is how to exploit the **sparsity structure** by polynomial-time interior-point algorithms so that they can also solve large-scale problems in practice.

1. The computational cost of each iteration in the dual algorithm is less than the cost of the primal-dual iterations.

2. In most combinatorial applications, we need only a lower bound for the optimal objective value of (SDP).

3. For large scale problems, $S$ tends to be very sparse and structured since it is the **linear combination** of $C$ and the data matrices $A_i$’s. This sparsity allows considerable savings in both memory and computation time.
Dual Algorithm: an Alternating Descent Method

\[ \psi_{n+\rho}(X, S) = \rho \ln(X \bullet S) - \ln \det X - \ln \det S. \]

Let \( \bar{z} = C \bullet X \) for any primal feasible \( X \) and consider the dual potential function

\[ \psi(y, \bar{z}) = \rho \ln(\bar{z} - b^T y) - \ln \det S, \quad (S = C - A^T y). \]

Its gradient is

\[ \nabla \psi(y, \bar{z}) = -\frac{\rho}{\bar{z} - b^T y} b + A S^{-1}. \quad (2) \]

We minimize reduce the potential by updating \( y \) while keep \( X \) unchanged. We also have

**Lemma 2** The barrier function \( B(y) = -\ln \det S(y) \) is second-order scaled Lipschitz with

\[ \beta_\alpha = \frac{1}{2(1-\alpha)}. \]
A Quadratic Over-Estimator of Potential

For any given \((y, S) \in \text{int} \, \mathcal{F}_d\), one can show that the Hessian of \(B(y) = - \ln \det S(y)\) is

\[
\nabla^2 B(y) = \begin{pmatrix}
A_1 S^{-1} \bullet S^{-1} A_1 & \cdots & A_1 S^{-1} \bullet S^{-1} A_m \\
\vdots & \ddots & \vdots \\
A_m S^{-1} \bullet S^{-1} A_1 & \cdots & A_m S^{-1} \bullet S^{-1} A_m
\end{pmatrix}.
\]

Let direction vector/matrix \((d_y, D_S = A^T d_y)\) where \(d_y^T \nabla^2 B(y) d_y \leq \alpha^2\) for some constant \(\alpha \in (0, 1)\) yet to be determined. Then

\[
\psi(y + d_y, \bar{z}) - \psi(y, \bar{z}) \leq \nabla \psi(y, \bar{z})^T d_y + \frac{1}{2(1 - \alpha)} d_y^T \nabla^2 B(y) d_y. \tag{3}
\]

Thus, we solve the Ellipsoidal Constrained Problem:

\[
\text{Minimize} \quad \nabla \psi^T(y, \bar{z}) d_y \\
\text{subject to} \quad d_y^T \nabla^2 B(y) d_y \leq \alpha^2, \tag{4}
\]
Close-Form Solution and Potential Reduction

In what follows we let the current duality gap be $\Delta = \bar{z} - b^T y$. The optimal solution, $d^*_y$, of (4) is given by a close form

$$d^*_y = \frac{\alpha}{\sqrt{\nabla \psi^T(y, \bar{z}) (\nabla^2 B(y))^{-1} \nabla \psi(y, \bar{z})}} d(\bar{z})_y,$$

where

$$d(\bar{z})_y = -(\nabla^2 B(y))^{-1} \nabla \psi(y, \bar{z}) = \frac{\rho}{\Delta} (\nabla^2 B(y))^{-1} b - (\nabla^2 B(y))^{-1} AS^{-1}.$$  

We can derive

$$\nabla \psi^T(y, \bar{z}) d(\bar{z})_y = -\nabla \psi^T(y, \bar{z}) (\nabla^2 B(y))^{-1} \nabla \psi(y, \bar{z}) = -\|P(\bar{z})\|^2 \text{ where}$$

$$P(\bar{z}) = \frac{\rho}{\Delta} S^{0.5} X(\bar{z}) S^{0.5} - I,$$

and

$$X(\bar{z}) = \frac{\Delta}{\rho} S^{-1} \left(A^T d(\bar{z})_y + S\right) S^{-1}.$$  

$$\psi(y + d^*_y, \bar{z}) - \psi(y, \bar{z}) \leq -\alpha \|P(\bar{z})\| + \frac{\alpha^2}{2(1 - \alpha)}.$$  


Potential Primal Feasible Solution and its Objective Value

$X(\tilde{z})$ is actually the minimizer of the least squares problem

$$\text{Minimize } ||S^{-5}XS^{-5} - \frac{\Delta}{\rho} I||$$
subject to $AX = b$. 

(11)

so that it is the candidate of a new primal feasible solution. Furthermore, its objective value

$$C \cdot X(\tilde{z}) = b^T y + S \cdot X(\tilde{z})$$
$$= b^T y + S \cdot \left( \frac{\Delta}{\rho} S^{-1} (A^T (d(\tilde{z})_y) + S^k) S^{-1} \right)$$
$$= b^T y + \frac{\Delta}{\rho} I \cdot \left( S^{-1} A^T (d(\tilde{z})_y) + I \right)$$
$$= b^T y + \frac{\Delta}{\rho} \left( d(\tilde{z})_y^T (AS^{-1}) + n \right)$$

Since the vectors $AS^{-1}$ and $d(\tilde{z})_y$ were calculated, the cost of computing a primal objective value is the cost of a vector dot product!

But $X(\tilde{z})$ may not be PSD...
When $X(z)$ is PSD

We have the following lemma:

**Lemma 3** Let $\mu = \frac{\Delta}{n} = \frac{z-b^T y}{n}$, $\mu^+ = \frac{X(z) S}{n} = \frac{C \cdot X(z) - b^T y}{n}$, $\rho \geq n + \sqrt{n}$, and $\alpha < 1$. If

$$\|P(z)\| < \min \left( \alpha \sqrt{\frac{n}{n + \alpha^2}}, 1 - \alpha \right),$$

then the following three inequalities hold:

1. $X(z) \succ 0$;
2. $\|S^{1.5} X(z) S^{1.5} - \mu^+ I\| \leq \alpha \mu^+$;
3. $\mu^+ \leq (1 - \frac{\alpha}{2\sqrt{n}}) \mu$. 


Thus, if \( \|P(\bar{z})\| \geq \min \left( \alpha \sqrt{\frac{n}{n+\alpha^2}}, 1 - \alpha \right) \), we update \( y \) to \( (y^+ = y + d_y^*, S^+ = C - A^T y^+) \); otherwise, we let \( X^+ = X(\bar{z}) \) and \( \bar{z}^+ = C \bullet X^+ \). In such alternating moves, we have

**Theorem 2** *Either the primal-dual potential*

\[
\psi(X, S^+ ) \leq \psi(X, S) - \delta
\]

*or*

\[
\psi(X^+, S) \leq \psi(X, S) - \delta,
\]

*where \( \delta > 1/20 \).
Description of Dual SDP Potential Reduction Algorithm

Given an upper bound $\bar{z}_0$ and a dual interior point $(y_0, S^0) \in \text{int } \mathcal{F}_d$ set $k = 0$, $\rho > n + \sqrt{n}$, $\alpha \in (0, 1)$, and do the following:

while $\bar{z}_k - b^T y_k \geq \epsilon$ do

1. Compute $\nabla \psi(y_k, \bar{z}_k)$ and formulate the Hessian matrix $\nabla^2 B(y_k)$.

2. Solve (6) for the dual step direction $d(\bar{z}_k)_y$.

3. Calculate $\|P(\bar{z}_k)\|$ using (7).

4. If (12) is true, then $X^{k+1} = X(\bar{z}_k)$, $\bar{z}^{k+1} = C \bullet X^{k+1}$, and $(y^{k+1}, S^{k+1}) = (y_k, S_k)$;

   else $y^{k+1} = y_k + \frac{\alpha}{\|P(\bar{z}_k)\|} d(\bar{z}^{k+1})_y$, $S^{k+1} = C - A^T y^{k+1}$, $X^{k+1} = X^k$, and $\bar{z}^{k+1} = \bar{z}_k$.

   end if

5. $k := k + 1$.

end while
Corollary 1 Let $\rho = O(\sqrt{n})$. Then, the Algorithm terminates in at most $O(\sqrt{n} \log(C \cdot X^0 - b^T y^0)/\epsilon)$ iterations with

$C \cdot X^k - b^T y^k \leq \epsilon$.

- You do not need eigenvalue computation in evaluate $\|P(\tilde{z})\|$, but use $\|P(\tilde{z})\|^2 = \nabla \psi_T(y, \tilde{z}) d(\tilde{z}) y$.
- When $\tilde{z}$ is updated, it is easy to recompute $d(\tilde{z}) y$ and $\|P(\tilde{z})\|$.
- You may be more proactive in updating $X$.
- You may take a more aggressive step-size.
- Exploit data structure in constructing the Barrier Hessian $\nabla^2 B(y)$, where its $ij$th element is $A_i S^{-1} \cdot S^{-1} A_j$. 


When $A_i = a_i a_i^T$, the Hessian matrix can be rewritten in the form

$$\nabla^2 B(y) = \begin{pmatrix} (a_1^T S^{-1} a_1)^2 & \cdots & (a_1^T S^{-1} a_m)^2 \\ \vdots & \ddots & \vdots \\ (a_m^T S^{-1} a_1)^2 & \cdots & (a_m^T S^{-1} a_m)^2 \end{pmatrix}$$

(13)

and

$$\nabla B(y) = AS^{-1} = \begin{pmatrix} a_1^T S^{-1} a_1 \\ \vdots \\ a_m^T S^{-1} a_m \end{pmatrix}.$$ 

This matrix can be computed very quickly without computing nor saving $S^{-1}$ specifically but its factor.
Quick Computation with the Rank-One Structure at Step $k$

Let original data be given as $A^T = [a_1 \ a_2 \ ... \ a_m] \in \mathbb{R}^{n \times m}$.

- Compute matrices/vectors in $O(n^3 + n^2 m + nm^2)$ arithmetic operations (if the data are dense):

  $$A_s = AS^{-1/2} \in \mathbb{R}^{m \times n} \quad \text{and} \quad A_s A_s^T \in S^m$$

  $$\nabla B(y) = AS^{-1} = \text{diag}(A_s A_s^T) \in \mathbb{R}^m$$

  $$\nabla^2 B(y) = [A_s A_s^T]^2 \in S^m$$

- Compute the dual direction vector in $O(m^3)$ operations.

  $$d(\cdot)_y = - (\nabla^2 B(y))^{-1} \nabla \psi(\cdot).$$

- The norm of $P(\cdot)$ can be checked in $O(m^2)$ operations and the new upper bound can be updated in $O(m)$ operations.

- At the termination, primal solution $X(\cdot)$ can be computed in $O(n^3 + n^2 m)$ operations.
Software Implementation

**SEDUMI**: http://sedumi.mcmaster.ca/

**MOSEK**: http://www.mosek.com/products_mosek.html

**SDDPT3**: http://www.math.nus.edu.sg/~mattohkc/sdpt3.html


**CVX**: http://www.stanford.edu/~boyd/cvx

**DSDPMLB**: on piazza
Applications of SDP with Data Rank-One Structure Besides SNL

Homogeneous Quadratically Constrained Quadratic Programming (HQCQP):

\[ z^* := \text{Maximize} \quad x^T Q x \]
\[ \text{Subject to} \quad x^T A_i x = b_i, \forall i = 1, \ldots, m. \]

\[ z^* := \text{Maximize} \quad Q \cdot X \]
\[ \text{Subject to} \quad A_i \cdot X = b_i, \forall i = 1, \ldots, m, \]
\[ X \succeq 0, \text{rank}(X) = 1. \]

SDP Relaxation (SDPR) of the original problem:

\[ z_{sdp} := \text{Maximize} \quad Q \cdot X \]
\[ \text{Subject to} \quad A_i \cdot X = b_i, \forall i = 1, \ldots, m, \]
\[ X \succeq 0. \]
Binary Quadratic Maximization

\[(\text{BQP})\quad z^* := \text{Maximize} \quad x^T Q x \]

subject to \((x_j)^2 = 1, \ j = 1, \ldots, n.\)

**Max-Cut Problem:** An undirected graph \(G = (V, E)\) with non-negative weights \(w_{ij}\) for each edge in \(E\) (and \(w_{ij} = 0\) if \((i, j) \notin E\)), which is the problem of partitioning the nodes of \(V\) into two sets \(S\) and \(V \setminus S\) so that

\[w(S) := \sum_{i \in S, j \in V \setminus S} w_{ij}\]

is maximized. A problem of this type arises from many network planning, circuit design, and scheduling applications.

**Max Bisection Problem:** The number of nodes in each of the two sets is identical.
Figure 1: Illustration of the max-cut problem
SDP Relaxations:

Maximize \( w(x) := \sum_{i,j} w_{ij} (1 - x_i x_j) \)

Subject to \( (x_j)^2 = 1, \ j = 1, \ldots, n, \)

\[ (e^T x)^2 = 0, \] if Bisection.

SDP Relaxation:

Maximize \( Q \cdot X \)

Subject to \( e_j e_j^T \cdot X = 1, \ j = 1, \ldots, n, \)

\[ e e^T \cdot X = 0, \] if Bisection.

A major research is how to regulate the objective to find a lowest-rank solution.
The null-space reduction: find an extreme SDP solution in the null-space of the linear constraints.

The eigenvalue reduction: let $X^*$ be an SDP solution with rank $r$ and $X^* = \sum_{i=1}^{r} \lambda_i v_i v_i^T = V V^T$ where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Then, let $\hat{X} = \sum_{i=1}^{d} \lambda_i v_i v_i^T$.

Objective-regulated reduction: add a suitable regularative objective with a suitable weight (see HW4).

Binary randomized reduction: let random vector $u \in N(0, I)$ and $\hat{x} = \text{Sign}(Vu)$ where

$$\text{Sign}(x) = \begin{cases} 
1 & \text{if } x \geq 0 \\
-1 & \text{otherwise.}
\end{cases}$$

Continuous randomized reduction: let random matrix

$$R = \sum_{i}^{d} \xi_i \xi_i^T, \quad \xi_i \in N(0, \frac{1}{d}I); \quad \text{or} \quad \xi_i \in \text{Binary}(0, \frac{1}{d}I)$$

that is, each entry either $1$ or $-1$ in the latter case. Then assign $\hat{X} = VRV^T$. 

21
Interior-Point Algorithm for Linearly Constrained Optimization I

\[
\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad Ax = b, \quad x \geq 0, \\
& \quad Ax = b, \quad \text{OPT Conditions: } \quad s = \nabla f(x) - A^T y \geq 0, \\
& \quad x \geq 0, \\
& \quad Xs = 0.
\end{align*}
\]

Furthermore, let \( f(x) \) be convex and satisfy the second-order \( \beta_\alpha \)-Scaled Lipschitz condition: for any point \( x > 0 \)

\[
\|X (\nabla f(x + d) - \nabla f(x) - \nabla^2 f(x)d)\| \leq \beta_\alpha d^T \nabla^2 f(x)d, \text{ whenever } \|X^{-1}d\| \leq \alpha(1).
\]

Given a strictly feasible \( (x > 0, y, s = \nabla f(x) - A^T y > 0) \), compute direction vectors \( (d_x, d_y, d_s) \):

\[
\begin{align*}
Sd_x + Xd_s & = r := \frac{x^T s}{n+\rho}e - Xs, \\
Ad_x & = 0, \\
\nabla^2 f(x)d_x - A^T d_y - d_s & = 0.
\end{align*}
\]
Let

\[
\theta = \frac{\alpha \sqrt{\min(XSe)}}{\|(XS)^{-1/2}r\|}
\]

where \( \alpha \in (0, 1) \) is a constant depending on \( \beta_\alpha \), and

\[
x^+ = x + \theta d_x, \quad y^+ = y + \theta d_y, \quad \text{and} \quad s^+ = \nabla f(x^+) - A^T y^+. \]

Then, \((x^+ > 0, y, s^+ > 0)\) is strictly feasible and for a constant \( \delta \)

\[
\psi_{n+\rho}(x^+, s^+) - \psi_{n+\rho}(x, s) \leq -\delta.
\]