Mixed Linear And Semidefinite Programming For Combinatorial And Quadratic Optimization

Present by Eric Sullivan and Lu Yu

1 Introduction to the problem

The original problem is:

$$\begin{array}{ll} \min & C \bullet (vv^T) + c^T x := v^T C v + c^T x \\ s.t. & A_i \bullet (vv^T) + a_i^T x := v^T A_i v + a_i^T x = b_i,\ i = 1, \ldots, m \\ & x \geq 0 \end{array}$$

where $C, A_i \in S^n, c, a_i \in R^n$.

We relax it to an SDP problem:

$$\begin{array}{ll} \min & C \bullet X \\ s.t. & A_i \bullet X = b_i,\ i = 1, \ldots, m \\ & X \in K \end{array}$$

where $K = K_1 \oplus K_2 \oplus, \ldots, \oplus K_r$, K_l is the cone of $n_l \times n_l$ symmetric positive semidefinite matrices, and $C, A_i \in S^{n \times n}$ are symmetric.

Its dual can be written as

$$\begin{array}{ll} \max & b^T y \\ s.t. & \sum_{i=1}^m y_i A_i + S = C, \\ & S \succeq 0 \end{array}$$

There are several methods to solve it: primal potential reduction algorithm; dual-scaling algorithm and primal-dual scaling algorithm.

For large-scale problem, the dual slack variable S tends to be very sparse and structured, the sparsity allows savings in both memory and computation time. The primal matrix X may be much less sparse. Then we mainly talk about the dual-scaling algorithm since it may fully use the sparseness and the structure of the data.

2 Dual scaling algorithm

Firstly we introduce some notations:

$$\mathcal{A}(X) = \begin{pmatrix} A_1 \bullet X \\ A_2 \bullet X \\ \vdots \\ A_m \bullet X \end{pmatrix}$$

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

We tend to reduce the duality gap by decreasing the dual potential function:

Let $\overline{z} = C \bullet X$ for some feasible X.
We define the dual potential function:

$$\psi(y, \overline{z}) = \rho \ln(\overline{z} - b^T y) - \ln((\det S)$$

The reduction of the first term decreases the duality gap, while the second term keeps S in the interior of the positive semidefinite matrix cone. And at every step we have to find a strictly feasible X.

The gradient of the potential function is:

$$\nabla\psi(y, \overline{z}) = -\frac{\rho}{\overline{z} - b^T y} b + \mathcal{A}(S^{-1}).$$

From the Lemma 1, and some computations, we also have:

$$\psi(y, \overline{z}^k) - \psi(y^k, \overline{z}^k)$$

$$\leq \nabla\psi(y^k, \overline{z}^k)^T (y - y^k) + \frac{\|(S^k)^{-.5}(\mathcal{A}^T(y - y^k))(S^k)^{-.5}\|_F}{2(1 - \|(S^k)^{-.5}(\mathcal{A}^T(y - y^k))(S^k)^{-.5}\|_\infty)} \quad (1)$$

Then we want to reduce the right-hand-side of the above inequality.

So beginning with a strictly feasible dual point (y^k, S^k) and upper bound, we solve the following problem:

$$\begin{array}{ll} \min & \nabla\psi(y^k, \overline{z}^k)^T(y - y^k) \\ s.t. & \|(S^k)^{-.5}(\mathcal{A}^T(y - y^k))(S^k)^{-.5}\|_F \leq \alpha \end{array}$$

where $\alpha \in (0, 1)$.

We want to minimize the first part of (1), and the constraint is to control the second part of (1) since for a given matrix, its Frobenius norm is always greater than or equal to its infinite norm.

Let $\hat{A}_i = (S^k)^{-.5}A_i(S^k)^{-.5}$. The first order K-K-T condition shows that the minimum point y^{k+1} should satisfy the followings:

$$M^k(y^{k+1} - y^k) + \beta\nabla\psi(y^k, \overline{z}^k) = 0$$

where $M^k = \begin{pmatrix} \hat{A}_1 \bullet \hat{A}_1 & \dots & \hat{A}_1 \bullet \hat{A}_m \\ \vdots & \ddots & \vdots \\ \hat{A}_m \bullet \hat{A}_1 & \dots & \hat{A}_m \bullet \hat{A}_m \end{pmatrix}$

for some positive scalar β.

We get the answer:

$$y^{k+1} = y^k + \frac{\alpha}{\sqrt{\nabla\psi(y^k, \overline{z}^k)^T(M^k)^{-1}\nabla\psi(y^k, \overline{z}^k)}}d(\overline{z}^k)_y$$

where $d(\overline{z}^k)_y = \dfrac{\rho}{\overline{z}^k - e^Ty^k}(M^k)^{-1}b - (M^k)^{-1}\mathcal{A}((S^k)^{-1})$.

Here are two ways to compute the M^k quickly when $A_i = a_ia_i^T$. We won't discuss about them too much, the details is given in the original paper.

We have finished finding y^{k+1} in the above part, in the following we will try to find the $\overline{z}$

We have to know $\overline{z}^{k+1} = C \bullet X(\overline{z}^k)$, that is, we have to find $X(\overline{z}^k)$.
To find a feasible point X, we solve

$$\begin{array}{ll} \min & \|(S^k)^{.5}X(S^k)^{.5} - \dfrac{\overline{z}^k - b^Ty^k}{\rho}I\| \\ s.t. & \mathcal{A}(X) = b \end{array}$$

The solution is

$$X(\overline{z}^k) = \frac{\overline{z}^k - b^T y^k}{\rho}(S^k)^{-1}(\mathcal{A}^T(d(\overline{z}^k)_y) + S^k)(S^k)^{-1}.$$

which costs lots of compututional work. But what we want is:

$$C \bullet X(\overline{z}^k). = b^T y^k + \frac{\overline{z}^k - b^T y^k}{\rho}(d(\overline{z}^k)_y^T \mathcal{A}((S^k)^{-1}) + n).$$

in which we've calculated $d(\overline{z}^k)_y^T$ and $\mathcal{A}((S^k)^{-1})$.

To determine if $X(\overline{z}^k)$ is good, which means $X(\overline{z}^k)$ is really positive definite, we have to check Lemma 2.

Define $P(\overline{z}^k) = \dfrac{\rho}{\overline{z}^k - b^T y^k}(S^k)^{.5}X(\overline{z}^k)(S^k)^{.5} - I.$

lemma 2 1. *Let* $\mu^k = \frac{\overline{z}^k - b^T y^k}{n}, \mu = \frac{C \bullet X(\overline{z}^k) - b^T y^k}{n}, \rho \geq n + \sqrt{n}, \alpha < 1. If$

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

then the following hold:
(1)$X(\overline{z}^k) \succ 0$;
(2)$\|(S^k)^{.5}X(\overline{z}^k)(S^k)^{.5} - \mu I\| \leq \alpha\mu$;
(3)$\mu \leq (1 - \frac{0.5\alpha}{\sqrt{n}})\mu^k$.

If the condition of lemma 2 hold, we choose $X^{k+1} = X(\overline{z}^k)$, if not, we stay on $X^{k+1} = X^k$.

Then it follows that

$$\nabla\psi(y^k, \overline{z}^k)^T d(\overline{z}^k)_y = -\|P(\overline{z}^k)\|^2,$$
$$\psi(y^k, \overline{z}^k)^T(y^{k+1} - y^k) = -\alpha\|P(\overline{z}^k)\|,$$
$$y^{k+1} = y^k + \frac{\alpha}{\|P(\overline{z}^{k+1})\|}d(\overline{z})_y.$$

Choose α such that $S^{k+1} = C - \mathcal{A}^T(y^{k+1}) \succ 0$. Finally we have

$$\psi(y^{k+1}, \overline{z}^k) - \psi(y^k, \overline{z}^k) \leq -\alpha\|P(\overline{z}^k)\| + \frac{\alpha}{2(1-\alpha)}.$$

which reduces our potential function.

And the followings is the detailed algorithm:

Dual Algorithm:
Starting:
Found an upper bound $\overline{z}^0$, a dual point (y^0, S^0) such that $S^0 = C - \mathcal{A}^T y^0 \succ 0$, set $k = 0, \rho > n + \sqrt{n}, \alpha \in (0, 1)$.
Checking for optimality:
While $\overline{z}^k - b^T y^k \geq \epsilon$, do the following:
Computing and Updating:
(1) Compute $\mathcal{A}((S^k)^{-1})$ and M^k.
(2) Compute the dual step direction $d(\overline{z}^k)_y$.
(3) Caculate $\|P(\overline{z}^k)\|$.
(4) **If** the conditions of lemma 2 is true,
$X^{k=1} = X(\overline{z}^k), \overline{z}^{k+1} = C \bullet X^{k+1}$.
else $X^{k+1} = X^k, \overline{z}^{k+1} = \overline{z}^k$,
$y^{k+1} = y^k + \dfrac{\alpha}{\|P(\overline{z}^{k+1})\|} d(\overline{z})_y, S^{k+1} = C - \mathcal{A}^T(y^{k+1})$.

And in the next section we will talk some applications that could be solved by this algorithm, but we manly focused on the introducing and modeling to these specific problems.

3 Applications

3.1 Combinatorial Optimization

3.1.1 The Max Cut Problem

Given a graph, $G(V, E)$, the max cut problem of the graph is to partition or cut V into two subsets, call them S and $V \backslash S$, in order to maximize the number of edges between the two subsets. The problem that follows is:

$$\begin{aligned} \max \quad & \text{trace}(Lxx^T) \\ s.t. \quad & x_i \in \{1, -1\} \end{aligned}$$

which can be relaxed to a semidefinite program by replacing xx^T with a positive semidefinite matrix, X. The constraint would then become Diag(X) $= e$.

3.1.2 Other Combinatorial Problems

The max cut problem forms a basis for a class of problems. Some other members of this class add new constraints. The unequal cut, equal cut, and s-t cut problems are such members of this class.

Unequal cut The unequal cut problem requires that the two sets of the partition be a particular size. This is achieved by defining a parameter, $\kappa = \|S\| - \|V \backslash S\|$. The additional constraint $\|ev^T\| = \kappa$ added to the original problem. This constraint becomes trace $(ee^T X) = \kappa^2$ in the relaxation.

Equal cut The equal cut problem is similar except that the two sets are required to be the same size. This complicates the problem since if trace $(ee^T X) = 0$ then X is singular. In order for interior point methods to operate a variable is added to this constraint and the objective function. This new variable is given a penalty coefficient in the objective function so the semidefinite problem is now a mixed linear and semidefinite problem:

$$\begin{aligned} \max \quad & \text{trace}(XL) + \Lambda x \\ s.t. \quad & \text{Diag}(X) = e \\ & \text{trace}(ee^T X) + x = 0 \\ & X \succeq 0 \\ & x \geq 0 \end{aligned}$$

s-t cut The s-t cut problem occurs when a node s and a node t must be placed in different subsets of V. In the original problem, the constraint $xe_{st} = 0$ is sufficient. For the relaxation, this becomes trace $(e_{st}e_{st}^T X) = 0$ but this constraint implies that X is only feasible on the boundary of the semidefinite cone. In order to create an interior a linear variable x is added similarly to the equal cut case.

3.1.3 General Form

All of these problems can be represented in a mixed linear and semidefinite problem formulation:

$$\begin{array}{cc} \max & \text{trace}(CX) + cx^T \\ s.t. & \text{trace}(a_i a_i^T X) + \hat{\alpha}_i x = b_i \\ & X \succeq 0 \\ & x \geq 0 \end{array}$$

3.2 Rounding

Once the semidefinite relaxation is solved for X, a problem specific randomized rounding procedure is used to create a feasible solution to our initial combinatorial problem.

First, generate a random vector u and factor $X = VV^T$. Then calculate $v = Vu$. For both max cut and s-t cut it suffices to round each component of v to positive or negative one depending upon its current sign. However, the unequal and equal cut problems require more care. With v as above the components should be sorted into ascending order. If the median $(v) > 0$ then $w_i = \begin{cases} 1 & \text{if } i > \frac{n-x}{2}, \\ -1 & \text{otherwise.} \end{cases}$ But if the median $(v) \leq 0$ then $w_i = \begin{cases} 1 & \text{if } i > \frac{n+x}{2}, \\ -1 & \text{otherwise.} \end{cases}$ For the equal cut problem, $\kappa = 0$ so these are the very same operation. Unsorting w will create a feasible solution to the original problem.

3.3 DSDP

The DSDP implementation solves the dual problem, as described in the algorithm section:

$$\begin{array}{cc} \max & b^T y \\ s.t. & \sum_{i=1}^m y_i(a_i a_i^T) + S = C \\ & \sum_{i=1}^m y_i \hat{a}_i + s = c \\ & S \succeq 0 \\ & s \geq 0 \end{array}$$

The matrices are reordered to take advantage of any sparsity they contain.

In order to avoid calculating Schur compliments, this solver calculates

$$M_{ij} = (a_i^T S_k^{-1} a_j)^2 + \hat{a}_i^T (diag(s^k))^{-2} \hat{a}_j$$

by solving systems of linear equations, $S^k w = a_i$, then multiplying $w^t a_j$. Once M is known dy_1 and dy_2 are easily obtained.

This implementation chooses the best ρ such that

$$\rho \in \{0.8(\rho^{k-1} - n) + n, \rho^{k-1}, 1.2(\rho^{k-1} - n) + n, 1.2n\}$$

so the chosen ρ has the most improved objective value amongst those ρ with positive semidefinite X. This is checked by factoring $A^t d(z_k)_y + S_k$ and checking for positive pivots. The program uses further specialized methods to factor the sparse and dense matrices that become S.

Finally, the DSDP implementation using the dual scaling algorithm is the first implementation that converges polynomially, use the sparsity of the dual slacks. Since, DSDP works on the general form

$$\begin{aligned} \max \quad & \text{trace}(CX) + cx^T \\ s.t. \quad & \text{trace}(a_i a_i^T X) + \hat{\alpha}_i x = b_i \\ & X \succeq 0 \\ & x \geq 0 \end{aligned}$$

it solves max cut, s-t cut, unequal cut and equal cut problems with unprecedented speed. Especially impressive is that it is the first to solve the max cut instances with $10,000$ vertices.

4 conclusions

This paper firstly introduced the problem forms, then because of the special structures and properties of the dual to these original problems, they talked about the dual scaling algorithm which is efficient for solving them. They went on and mentioned a group of cut-problems which has the similar forms. To solve them, the solver DSDP was developed on the base of dual scaling algorithm. And the computational results shows that the solver was able to solve the large scaled cut-problems polynomially.