Lecture 8 (17.04.2013)

Scribe: Marcin Jedynak
Lecturer: Marek Cygan

Semidefinite Programming

1 Maximum Cut - Integer Program

We start with the following motivating example:

MAX-CUT
Input: A weighted, undirected graph $G = (V, E, w)$.
Output: A graph partition $(A, \overline{A})$, $A \subseteq V$ that maximizes $\sum_{uv \in E: u \in A, v \in B} w(uv)$.

The approximate solution of this problem can be found by choosing a uniformly random partition. Since every edge with probability $\frac{1}{2}$ has its ends in different sets, the expected value of the objective function is $\frac{\sum_{e \in E} w(e)}{2}$. Therefore this is a $\frac{1}{2}$-approximation algorithm.

Our goal now is to improve this approximation ratio. First we examine the following LP formulation:

$$\begin{aligned} maximize \sum_{uv \in E} z_{uv} w(uv) & \\ x_u + x_v \geq z_{uv} & \quad \forall uv \in E \\ (1 - x_u) + (1 - x_v) \geq z_{uv} & \quad \forall uv \in E \\ 0 \leq x_v \leq 1 & \quad \forall v \in V \\ 0 \leq z_{uv} \leq 1 & \quad \forall uv \in E \end{aligned}$$

The program above can be solved in polynomial time and then the result can be rounded. Unfortunately:

$$\begin{aligned} x_v = \frac{1}{2} & \quad \forall v \in V \\ z_{uv} = 1 & \quad \forall uv \in E \end{aligned}$$

is a feasible solution, which shows that this LP has integrality gap equal to two. Therefore we need to find another solution.

2 Maximum Cut - Semidefinite Program

Let's formulate the new IP as follows:

$$maximize \sum_{uv \in E} \frac{1}{2}(1 - y_u y_v) w(uv)$$
$$y_v^2 = 1 \quad \forall v \in V$$

In order to obtain a relaxation we will allow variables to be unit vectors in R^d for some d. The integer multiplication $y_u y_v$ will be treated as the dot product instead. We will show, that a program defined this way, the Semidefinite Program (SDP), can be solved in polynomial time. First we observe that this is actually the LP program. To see this let's replace every dot product $y_u y_v$ with a new variable a_{uv}. Now the program has the form:

$$maximize \sum_{uv \in E} \frac{1}{2}(1 - a_{uv}) w(uv)$$
$$a_{vv} = 1 \quad \forall v \in V$$

and is currently unbounded. The missing constraints should guarantee that there exists a set V of vectors in R^n such that $\forall u, v \; a_{uv} = y_u y_v$.

Let $A = (a_{uv})$ be the Gram matrix (matrix of inner products). Then A is obviously symmetric since the inner product is symmetric. Recall from the algebra course that:

Definition 1. A symmetric matrix $M \in R^{n \times n}$ is said to be positive semi-definite if

$$\forall x \in R^n \quad x M x^T \geq 0$$

Theorem 2. *Let M be a symmetric $n \times n$ matrix. The following are equivalent:*

1. *M is positive semi-definite.*
2. *All eigenvalues of M are non-negative.*
3. *There exists an $m \times n$ matrix V such that $M = VV^T$.*

Matrix A is positive semi-definite since we can take matrix formed from vectors y_v as the aforementioned matrix V. Hence the missing contraints have the form

$$x A x^T \geq 0 \quad \forall x \in R^n$$

which can be written as

$$\sum_{u,v \in V} x_u x_v a_{uv} \geq 0 \quad \forall x \in R^n$$

Although this set of constraints is infinite, this problem can still be solved in polynomial time as long as we are provided with the Separation Oracle.

The Separation Oracle can work as follows. We know that all eigenvalues of a positive semi-definite matrix are non-negative. Therefore they can be computed in polynomial time within desired accuracy ϵ. Solution is found if all of them are non-negative. Otherwise we have to fix the eigenvector corresponding to a negative eigenvalue. Hence:

Corollary 3. *Max-Cut SDP program can be solved in polynomial time.*

3 Random Rounding Algorithm

Now let's concentrate on transforming vectors from the SDP result to integer values. 0.878-approximation algorithm will be presented.

We choose a random vector r from the uniform distribution on the unit sphere. This vector uniquely determines the hyperplane through the origin that is perpendicular to it and separates vectors y_v. Hence we can define:

$$A = \{v : ry_v \geq 0\}$$
$$B = \{v : ry_v < 0\}$$

By spherical symmetry, every pair of vectors can be seen as vectors on the plane. Therefore picking a random hyperplane is equivalent to choosing a random diameter of the unit circle. Hence, from the equality:

$$y_u y_v = \cos(\angle y_u y_v)$$

the probability of cutting the egde uv is:

$$\mathbb{P}[(u,v) \textit{ is in cut}] = \frac{\arccos(y_u y_v)}{\pi}$$

Now we are interested in estimating the approximation ratio of the randomized rounding algorithm. It is known that:

$$\min_{(\angle y_u y_v) \in \{0,1\}} \frac{\frac{\angle y_u y_v}{\pi}}{\frac{1}{2}(1 - \cos(\angle y_u y_v))} \geq 0.87856\ldots$$

Therefore our algorithm will achieve ratio at least 0.878 at every egde. Hence:

Theorem 4 ([1])**.** *The randomized rounding algorithm has the approximation ratio* 0.878*.*

Remark 5. This algorithm can be derandomized.

Remark 6. Sometimes it is not sufficient to use R^2 relaxation (e.g. K_4).

Remark 7. If $P \neq NP$ there is no algorithm with an approximation ratio higher than $\frac{16}{17}$.

Remark 8. If the Unique Games Conjecture is true then the randomized rounding algorithm is optimal.

4 Semidefinite Programs - Formal Definition

Finally we present two formal, equivalent definitions of the SDP. As we have seen above, SDP variables can be vectors of R^d for some d. Hence the following definition appears:

$$\begin{aligned} maximize \sum_{i,j\in\{1,\ldots,n\}} & c_{ij} < x_i, x_j > \\ \sum_{i,j\in\{1,\ldots,n\}} & a_{ijk} < x_i, x_j > \leq b_k \quad \forall k \\ & x_i \in R^d \quad d \leq n \quad \forall i \in \{1, \ldots, n\} \end{aligned}$$

But we have dealt also with positive semi-definite matrices and we have observed, that elements of this matrices can be manipulated on separately. Therefore we can also formulate the following form of the SDP:

$$\begin{aligned} maximize \sum_{i,j\in\{1,\ldots,n\}} & c_{ij}x_{ij} \\ \sum_{i,j\in\{1,\ldots,n\}} & a_{ijk}x_{ij} \leq b_k \quad \forall k \\ X = (x_{ij})_{1\leq i,j\leq n} & \; is \; positive \; definite \end{aligned}$$

References

[1] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.