COMPSCI 590.1: Graph Algorithms	April 7, 2015

Lecture 24

Lecturer: Debmalya Panigrahi — *Scribe: Xi He*

1 Overview

In the past, we learned how to solve the minimum cut problem, but the *maximum cut problem* is also interesting and important. This first part of the lecture covered the maximum cut problem, and then the second part introduced *metric embedding* with its application on *group steiner tree problem*.

2 The Maximum Cut Problem

Definition 1. *$C = (S, \bar{S})$ is a partition of V of a graph $G = (V, E)$ into two disjoint subsets S and $\bar{S}$. The cut-set of a cut $C = (S, \bar{S})$ is the set $\{(u, v) \in E | u \in S, v \in \bar{S}\}$ of edges that have one endpoint in S and the other endpoint in $\bar{S}$.*

In this note, the cut C is referred as the cut-set and the size of the cut $|C|$ as the size of the cut-set. For a graph, a maximum cut is a cut whose size is at least the size of any other cut. The problem of finding a maximum cut in a graph is known as the *maximum cut problem*. The problem is NP-hard. Simple 0.5-approximation algorithms existed long time ago, but no improvement was made till 1990s by Goemans and Williamson [GW95] using semidefinite programming and randomized rounding that achieves an approximation ratio 0.878. We describe one 0.5-approximation algorithm in Section 2.1 and then Goemans and Williamson's method in Section 2.2.

2.1 Simple Approximation Algorithm

In this section, we present a simple deterministic polynomial-time 0.5-approximation algorithm based on local optimum solution described below.

Definition 2. *Given $G = (V, E)$, for all $v \in V$, a cut is local optimum for the maximum cut problem, if for all $v \in V$, the number of neighbors on the side of v is less or equal to the number of neighbors on the other side of the cut.*

We first start with an arbitrary partition of the vertices of the given graph $G = (V, E)$. A search of local optimum is carried out by finding a vertex which has more neighbors on its side than the other side of the cut. If such vertex exists, move this vertex to the other side of the cut, which will improve the size of the current cut. This search continues till no such vertex can be found.

Theorem 1. *Given $G = (V, E)$, the approximation factor of a local optimum solution for the maximum cut problem is $\frac{1}{2}$.*

Proof. The size of the local optimum cut is bounded by half of the total number of edges as shown below.

$$\begin{aligned}
\text{\# of edges in the cut} &= \frac{1}{2}\sum_{v\in V} d'_v, \text{ where } d'_v = \text{\# of edges in the cut incident to } v \\
&\geq \frac{1}{2}\sum_{v\in V}\frac{1}{2}d_v = \frac{1}{4}\sum_{v\in V} d_v = \frac{1}{2}|E|
\end{aligned}$$

As the total number of edges is the upper bound for the maximum cut, the approximation factor is $\frac{1}{2}$. □

2.2 Improved Approximation Algorithm

The maximum cut problem can be written as a linear program in the following way: Given a graph $G = (V,E)$, where $V = \{1,\ldots,n\}$, let x_{ij} be the indicator variable for each edge $(i,j) \in E$ to be chosen in the cut.

$$\begin{aligned}
\text{Maximize} \quad & \textstyle\sum_{(i,j)\in E} x_{ij} \\
\text{subject to} \quad & \textstyle\sum_{e\in T} x_e \leq 2 \quad \forall \text{ triangle } T & (1)\\
& x_{ij} \leq x_{ik} + x_{jk} \quad \forall \text{ triangle } (i,j,k) & (2)\\
& 0 \leq x_{ij} \leq 1 \quad \forall (i,j) \in E
\end{aligned}$$

The objective function is to maximize the cut, i.e. the sum of indicator variables. Equation (1) and (2) are a set of constraints based triangles K_3. A triangle has three edges and at most 2 edges can appear in the cut, which gives Equation (1). In addition, if two edges of a triangle do not exist in the cut, the third edge should not appear in the cut as well. This observation is part of the triangle inequality constraints, as summarized in Equation (2). It is not clear how to solve this linear program, but Goemans and Williamson [GW95] first formulated the problem into a quadratic problem, and then converted the quadratic problem into a semi-definite problem. The quadratic problem is described as follows. Given a graph $G = (V,E)$, where $V = \{1,\ldots,n\}$, let x_i be an variable associated to each vertex $i \in V$, the solution to the maximum cut problem is given by the following integer quadratic program:

$$\begin{aligned}
\text{Maximize} \quad & \textstyle\sum_{(i,j)\in E} \frac{1-x_i x_j}{2} \\
\text{subject to} \quad & x_i \in \{-1,+1\} \quad \forall i \in V & (3)
\end{aligned}$$

The solution of the above program gives a set $S = \{i|x_i = +1\}$ and $\bar{S} = \{i|x_i = -1\}$, which corresponds to a cut of size $\sum_{(i,j)\in E} \frac{1-x_i x_j}{2}$.

Solving this integer quadratic program is NP-complete, Goemans and Williamson relax the problem to higher dimension. Equation (3) restricts the variable x_i to be a 1-dimensional vector of unit norm. The relaxations is defined by allowing x_i to be a multidimensional vector v_i of unit Euclidean norm. Since the linear space spanned by the vectors v_i has dimension at most n, we can assume that these vectors belong to n-dimensional unit sphere S_n. The resulting relaxation gives this semidefinite program:

$$\begin{aligned}
\text{Maximize} \quad & \textstyle\sum_{(i,j)\in E} \frac{1-v_i\cdot v_j}{2} \\
\text{subject to} \quad & ||v_i|| = v_i \cdot v_i = 1 \quad \forall i \in V, & (4)
\end{aligned}$$

where $v_i \cdot v_j$ represents the inner product of v_i and v_j.

This semidefinite program is no longer NP hard, but can be solved to arbitrary precision in polynomial time. First, setting $v_i \cdot v_j = d_{ij}$ gives an linear objective function:

$$\begin{array}{rl} \text{Maximize} & \sum_{(i,j)\in E} \frac{1-d_{ij}}{2} \\ \text{subject to} & d_{ii} = 1 \quad \forall i \in V \\ & D = V^T V, \end{array} \tag{5}$$

where $V = (v_1, \ldots, v_n)$. Though Equation (5) is not linear constraint, D is a positive semi-definite matrix and hence $x^T D x \geq 0$, for all x. Therefore, Equation (5) is equivalent to a set of infinite number of linear constraints. We have seen in the previous lectures, regardless of exponential or infinite number of linear constraints, as long as we can find a separate oracle, we can solve this linear program in polynomial time.

Given the solution of this semidefinite program $(v_1, \ldots, v_n)$, a simple randomized algorithm is used for the rounding step in the maximum cut problem: choose a random hyperplane through the origin (let r be a vector uniformly distributed on the unit sphere S_n), and partition the vertices into those vectors that lie 'above' the plane ($S = \{i | v_i \cdot r \geq 0\}$) and those that lie 'below' it ($\bar{S} = \{i | v_i \cdot r < 0\}$).

Let W denote the size of the cut produced in this way, and $E[W]$ be the expected size of the cut which is characterized in the theorem below.

Theorem 2.

$$E[W] = \frac{1}{\pi} \sum_{(i,j)\in E} \arccos(v_i \cdot v_j).$$

Proof. Given a vector r drawn uniformly from the unit sphere S_n, we get by the linearity of expectation that

$$\begin{aligned} E[W] &= \sum_{(i,j)\in E} Pr[sgn(v_i \cdot r) \neq sgn(v_j \cdot r)] \\ &= \sum_{(i,j)\in E} \frac{\arccos(v_i \cdot v_j)}{\pi} \\ &= \frac{1}{\pi} \sum_{(i,j)\in E} \arccos(v_i \cdot v_j), \end{aligned}$$

where $sgn(x) = 1$ if $x \geq 0$, and -1 otherwise. □

Theorem 3. *Given the solution of the semidefinite program is $(v_1, \ldots, v_n)$ with its cut size $\sum_{(i,j)\in E} \frac{1-v_i \cdot v_j}{2}$, the ratio between $E[W]$ and this size is greater than 0.878.*

Proof. By Theorem 2, the ratio can be written as

$$\begin{array}{cc} & \frac{1}{\pi} \sum_{(i,j)\in E} \arccos(v_i \cdot v_j) / \sum_{(i,j)\in E} \frac{1-v_i \cdot v_j}{2} \\ \geq & \frac{2}{\pi} \min_{(i,j)\in E} \frac{\arccos(v_i \cdot v_j)}{1-v_i \cdot v_j} \\ = & \frac{2}{\pi} \min_{(i,j)\in E} \frac{\theta_{ij}}{1-\cos\theta_{ij}}, \text{ where } \theta_{ij} = \arccos(v_i, v_j) \\ \geq & \frac{2}{\pi} \min_{0\leq\theta\leq\pi} \frac{\theta}{1-\cos\theta} \\ > & 0.878 \end{array}$$

The last inequality can be obtained using simple calculus, one can see that the ratio achieves its value for $\theta = 2.331122$, which is the nonzero root of $\cos\theta + \theta \sin\theta = 1$. □

Though the discussion here is on unweighted graphs, the weighted problem can be approached in the same way. Similarly, semidefinite program can be useful to solve other NP-hard problems, one example is sparse cut problem, where the size of the smaller side of the cut is minimized.

3 Metric Embedding

In this section, we introduce metric embedding concept to map vertices in one metric space to anther metric space, with minimum distortion, and then solve group steiner tree problem with this metric embedding techniques. First, α-probabilistically approximate metric space of a metric space M defined in [GK00] is described below.

Definition 3. *A set of metric spaces $\mathbb{S}$ over V is said to α-probabilistically approximate a metric space M over V, if (1) for all $x,y \in V$ and $S \in \mathscr{S}$, $d_S(x,y) \geq d_M(x,y)$, and (2) there exists a probability distribution D over metric spaces in $\mathscr{S}$ such that for all $x,y \in V$, $E[d_D(x,y)] \leq \alpha d_M(x,y)$.*

Here we just consider one metric space M_2 over V to α-approximate V in the metric space M_1, such that

$$\alpha \cdot d_{M_1}(u,v) \geq d_{M_2}(u,v) \geq d_{M_1}(u,v), \forall u,v \in V.$$

Example 1. If embedding a complete graph K_n into a line graph L_n, the approximation factor α is the n, i.e.

$$n \cdot d_{K_n}(u,v) \geq d_{L_n}(u,v) \geq d_{K_n}(u,v), \forall u,v \in V.$$

Example 2. If embedding a general graph $G = (V,E)$ into a tree, the approximation factor α is $O(\log n)$, by Bartal [Bar96]. This technique is useful to solve the group steiner tree problem [GK00]. This problem can be stated formally as follows: we are given a graph $G = (V,E)$ with the cost function $c : E \to \mathbb{R}^+$ and subsets of vertices $g_1, \ldots g_k \subseteq V$. We call $g_1, \ldots, g_k$ groups. Given a root $r \in V$, the objective is to find the minimum cost subtree T of G that connects r to each of the set g_i. The following linear programming is the relaxation of the group steiner tree problem:

$$\begin{array}{ll} \text{Minimize} & \sum_{e \in E} c_e x_e \\ \text{subject to} & \sum_{e \in (S,\bar{S})} x_e \geq 1, \forall S \ \exists i, \text{ s.t. } g_i \subseteq S, \ \ r \in \bar{S} \\ & x_e \geq 0 \end{array}$$

For a general graph G, rounding for this linear programming is not clear, but for a tree graph $T'' = (V,E)$ is possible. Let the optimal solution for T'' to the linear program be T'. The rounding step starts from edges incident on root r. For each edge e, if its parent edge of e, denoted by f is included, then include e with probability $\frac{x_e}{x_f}$; and with probability 0 otherwise. If e is incident on r, include it with probability x_e. The expected cost of the tree T picked by this random experiments is equal to the cost of the optimal solution T' to the linear program. A general graph can be embedded into a tree space with distortion of $O(\log n)$ and solve the group steiner tree problem in the tree space. The next lecture will continue this topic.

4 Summary

This lecture covered the maximum cut problem which can be solved with semi-definite programming and group steiner tree with metric embedding technique.

References

[Bar96] Yair Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. *ACM Symposium on Theory of Computing*, pages 184–193, 1996.

[GK00] Naveen Garg and Goran Konjevod. A polylogarithmic approximation algorithm for the group steiner tree problem. In *Journal of Algorithm*, number 37, 2000.

[GW95] Michel X. Goemans and David P. Williamson. *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming*. Journal of the ACM, 1995.