6.854J / 18.415J Advanced Algorithms
Fall 2008

18.415/6.854 Advanced Algorithms	November 21, 2008

Lecture 19

Lecturer: Michel X. Goemans

1 Introduction

In this lecture, we revisit MAXCUT and describe a *randomized* γ ($\approx$ .87856)-approximation algorithm. We also explore SPARSEST-CUT, an NP-hard problem for which no constant factor approximation is known. We begin to describe an $O(\log k)$ approximation using *multicommodity flows*; here k is the number of commodities. To define the relationship between the optimal values of SPARSEST-CUT and multicommodity flow, we introduce metrics and finite metric spaces.

2 Revisiting MAXCUT

Recall the MAXCUT problem: given a graph $G = (V, E)$ and weights $w\colon E \to \mathbb{R}_+$ (we could assume that G is the complete graph and weights are 0 for the original non-edges), maximize $w(S : \bar{S})$ $(= \sum_{\substack{i\in S\\ j\in \bar{S}}} w_{ij})$ in $S \subset V$. MAXCUT can be formulated as the integer program

$$\max \sum_{(i,j)\in E} w_{ij}(1 - x_i x_j)/2$$

subject to

$$x_i \in \{\pm 1\}, \forall i.$$

The prior lecture described a 1/2-approximation algorithm and an upper bound on the solution to the above optimization, via reduction to a semidefinite program.

2.1 SDP Relaxation of MAXCUT

In the SDP relaxation, we replaced the x_i with unit vectors in the sphere $S^{n-1} := \{x \in \mathbb{R}^n \colon \|x\| = 1\}$. Thus, the goal of the relaxed MAXCUT was to find

$$\max \sum_{(i,j)\in E} w_{ij}(1 - v_i^T v_j)/2$$

subject to

$$v_i \in S^{n-1}, \forall i.$$

Though it is not immediately clear that this represents a semidefinite program, it can be reformulated as follows:

$$\max \sum_{(i,j)} w_{ij}(1 - Y_{ij})/2,$$

subject to

$$Y_{ii} = 1, \forall i$$
$$Y \succeq 0.$$

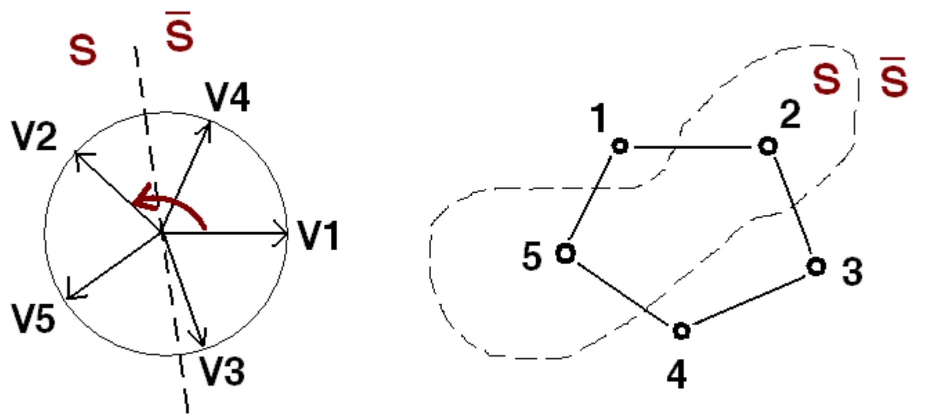

Figure 1: For the 5-cycle, the optimum vectors end up being in a lower-dimensional space (of dimension 2), see left figure. The angle between any two consecutive vectors is $4\pi/5$ and total SDP value is $5(1-\cos(4\pi/5))/2 = 4.52\cdots$. Taking a random hyperplane through the origin gives the cut $(S:\bar{S})$, see the right figure.

Given a solution to the SDP in the form of unit vectors v_i, we would like to find a feasible S giving as large a cut as possible. The ideal is to have vertices i and j separated by the cut when $(1-v_i^T v_j)/2$ is large, i.e., v_i and v_j are far apart on the sphere. Here is a way to do this. Choosing a hyperplane through the origin divides the vectors into two groups, and we let S be the intersection of one halfspace with the set of vectors. The sets of vectors on each side of the hyperplane correspond to S and $\bar{S}$. As an example, we illustrate the vectors for a cycle of length 5 in Figure 1.

Which hyperplane should we choose? Well, the optimum vectors are definitely not unique; any rotation of them (orthonormal transformation) will also provide an optimum solution since the objective function depends only on the inner products $(v_i^T v_j)$. Therefore we should not have a preferred direction for the hyperplane.

3 MAXCUT γ-Approximation Algorithm

This discussion provides the intuition behind the following randomized algorithm, due to Goemans and Williamson ([1]):

1. Choose a unit vector $r \in S^{n-1}$ uniformly.
2. Let $S = \{i \in V : r^T v_i \geq 0\}$.

Remark 1 *In the case $n = 2$, it is easy to pick a uniform r, by taking $\theta \in [0, 2\pi)$ uniformly, whence $r = (\cos\theta, \sin\theta)^T$. For a general n, we should find $r \in S^{n-1}$ by selecting each component independently from a Gaussian distribution, and then normalize to $\|r\| = 1$.*

Theorem 1 *The Goemans-Williamson algorithm is a randomized γ-approximation algorithm for MAXCUT, where* $\gamma = \min_{-1\leq x\leq 1} \frac{2\cos^{-1} x}{\pi(1-x)} (\approx .87856)$.

Proof: "OPT" and "SDP" will denote the optimal solution to the MAXCUT instance and its SDP relaxation. We show $E[w(S:\bar{S})] \geq \gamma \cdot \text{SDP} \geq \gamma \cdot \text{OPT}$.

By linearity of expectations, we have:

$$\begin{aligned} E[w(S:\bar{S})] &= E[\sum_{(i,j)} w_{ij}\{1\, if\ (i,j) \in (S:\bar{S});\ 0\, otherwise\}] \\ &= \sum_{(i,j)} w_{ij} Pr[(i,j) \in (S:\bar{S})]. \end{aligned}$$

If we were in dimension 2 then v_i and v_j are separated by the line orthogonal to r if and only if this line falls between v_i and v_j and this occurs with a probability $\angle(v_i, v_j)/\pi$ (where $\angle(v_i, v_j)$ denotes the angle between v_i and v_j). The same is also true for higher dimensions. Indeed, let p denote the projection of r onto the 2-dimensional space F spanned by v_i and v_j. We have

$$r^T v_i = p^T v_i$$
$$r^T v_j = p^T v_j$$

implying that v_i and v_j are separated for the partition defined by r if and only if they are separated for the partition defined by p. But $p/||p||$ is uniform over the unit circle in F. Therefore,

$$Pr[(i,j) \in (S:\bar{S})] = \angle(v_i, v_j)/\pi$$

and, using the fact that v_i and v_j are unit vectors (and thus $v_i^T v_j = \cos\angle(v_i, v_j)$):

$$Pr[(i,j) \in (S:\bar{S})] = \cos^{-1}(v_i^T v_j)\pi.$$

So, we get a closed-form formula for the expected weight of the cut produced:

$$E[w(S:\bar{S})] = \sum_{(i,j)} w_{ij} \cos^{-1}(v_i^T v_j)/\pi.$$

On the other hand, we know that

$$\text{SDP} = \sum_{(i,j)} w_{ij}(1 - v_i^T v_j)/2.$$

Since w_{ij} is non-negative, $E[w(S:\bar{S})]/\text{SDP} \geq$ the smallest ratio over all (v_i, v_j):

$$\begin{aligned} E[w(S:\bar{S})]/\text{SDP} &\geq \min_{-1 \leq x \leq 1} (\cos^{-1}(x)/\pi)/[(1-x)/2] \\ &=: \gamma(\approx 0.87856). \end{aligned}$$

□

Several remarks are in order.

Remark 2 *The analysis is tight in the sense that, for any $\varepsilon > 0$, there exist instances such that* OPT/SDP $\leq \gamma + \varepsilon$.*[2]*

Remark 3 *It is possible to derandomize Goemans-Williamson (and achieve a performance guarantee of γ); still, in practice, the fact that one can output many cuts is useful as one can then exploit the variance of the weight of the cut.*

Remark 4 *No approximation algorithm achieving better than γ is currently known.*

Remark 5 *Approximating MAXCUT within* $16/17$ *($\approx$ .94117) $+\varepsilon$ for any $\varepsilon > 0$ is NP-hard[3]. Approximating MAXCUT within $\gamma + \varepsilon$ for any $\varepsilon > 0$ is UGC-hard; that is, an efficient algorithm doing such would imply the falsity of the Unique Games Conjecture.*

Remark 6 *It can be shown that the SDP relaxation above always has an optimal solution in dimension r where $\frac{r(r+1}{2} \leq n$ (i.e. $r \leq 2\sqrt{n}$).*

4 SPARSEST-CUT and Multicommodity-Cut

We now consider the problem of identifying a sparse cut in a graph: one which is as small as possible, *relative to* the number of edges which could exist between the sets of vertices. The latter quantity is maximized by balancing the vertices across the partition. Hence, we seek $S \subset V$ minimizing $w(S : \bar{S})/|S \times \bar{S}|$. A generalization of SPARSEST-CUT is the multicommodity cut problem, in which we have, in addition to a capacitated $G = (V, E)$, some k commodities, each associated with a "demand" f_i and a source and sink $s_i, t_i \in V$. (The idea is that we want to ship f_i units of commodity i from s_i to t_i.) We seek the value of a cut $(S : \bar{S})$ with minimum capacity relative to the demand across it, i.e.,

$$\min_{S:\bar{S}} \frac{u(S : \bar{S})}{[\sum_{i:(s_i,t_i)\in(S:\bar{S})} f_i]}.$$

We will write β for the objective in this expression, and denote its optimum by β^*.

We recover SPARSEST-CUT by taking $u = w$ and creating a commodity of demand 1 for each pair of vertices. As another special case, when $k = 1$, we are minimizing $u(S : \bar{S})$ over cuts separating s and t, so we have the min s–t cut problem (in an undirected graph).

4.1 Concurrent multicommodity flow

Let us now discuss a problem which is in a sense dual to the multicommodity cut. In concurrent multicommodity flow, we are given $G = (V, E)$ with k commodities and capacity constraints on each edge $\in E$, and seek the maximum α such that we can send αf_i units of flow across the graph from s_i to t_i for all i simultaneously, without violating the capacity constraints on each edge. Let α^* denote the optimal value. It is easy to see how to do multicommodity flow by linear programming.

The multicommodity cut and flow problems are related by $\alpha^* \leq \beta^*$. Indeed, if we can send αf_i from s_i to t_i for all i, $u(S : \bar{S})$ must be at least αf_i for each (s_i, t_i) in the cut, so

$$\beta = \frac{u(S : \bar{S})}{[\sum_{i:(s_i,t_i)\in(S:\bar{S})} f_i]} \geq \alpha$$

for all feasible β and α. This is a "weak duality"-type condition.

If $k = 1$, we have equality, by the max $s - t$ flow min $s - t$ cut theorem (one can show that the theorem for directed graphs implies it also for undirected graphs). It is non-obvious that we have $\alpha^* = \beta^*$ for $k = 2$ as well. In general, however, we do not have equality. In figure 2, we show an example of a graph with a relatively small number of commodities (4) for which α^* is strictly less than β^*.

In this graph, all capacities have value = 1. For this graph, $\beta^* = 1$. Consider the multicommodity cut given by the dashed line. For this cut, and any similar cuts, the sum of the capacities across the cut is $u(S : \bar{S}) = 3$ and the amount of demand that needs to go through it is $\sum_{i:(s_i,t_i)\in(S:\bar{S})} f_i = 3$ also. If we choose a cut for which the capacties sum to 2 instead, the sum of the demands will also be 2. Therefore, $\beta^* = 1$.

What is α^* though? There are $k = 4$ commodities in this graph, and yet a maximum of 3 units of flow can be pushed across a cut at one time. Since s_2 and t_2 are on the same side of the cut, you might think that α^* might be able to reach 1. However, since each s_i is at least two edges away from its t_i and there are 4 commodities, if $\alpha^* = 1$ then the sum of the flow on all the edges of the graph would have to be $(4)(2)(1) = 8$. Yet there only 6 edges, each with capacity 1. This shows that $\alpha^* \leq 3/4$.

So what IS the relationship between α^* and β^* in general?

Theorem 2

$$\frac{\beta^*}{\alpha^*} = O(\log k).$$

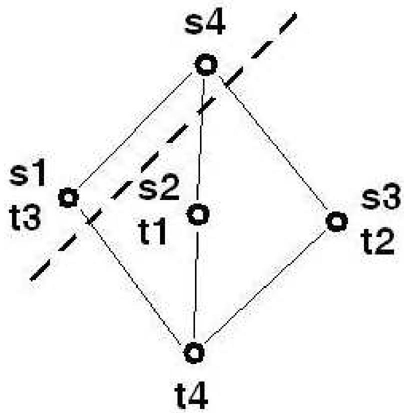

Figure 2: An Example Graph where $\alpha^* < \beta^*$.

Remark 7 *Computing β^* is NP-hard. However—as we will see in the upcoming lecture — we can get a $O(\log k)$ approximation using the LP we have for α^*, and a tighter $O(\sqrt{\log k})$ approximation using an SDP.*

To prove the above result, we introduce *metric spaces*.

5 Finite Metric Spaces

Definition 1 *Let X be an arbitrary set, and d a function $X \times X \to \mathbb{R}$. (X, d) is a* metric space *if the following properties hold for all $x, y, z \in X$:*

1. $d(x, y) \geq 0$ *(Nonnegativity)*
2. $d(x, y) = d(y, x)$ *(Reflexivity)*
3. $d(x, y) + d(y, z) \geq d(x, z)$ *(Triangle Inequality)*

For simplicity, we will deal only with finite metric spaces (i.e. $|X|$ is finite).

Definition 2 *Let X, Y be sets with associated metrics d, ℓ. For $c \geq 1$, we say that (X, d)* embeds into (Y, ℓ) with distortion c *if there is a mapping $\phi\colon X \to Y$ such that for any $x, y \in X$, $d(x, y) \leq \ell(\phi(x), \phi(y)) \leq c d(x, y)$. If $c = 1$, the embedding is called* isometric.

This distortion measure is useful when we can transform a problem defined on one metric into another metric that is easier to deal with. This is precisely what we will do in the context of multicommodity cuts and flows.

The most familiar metric spaces are n-dimensional Euclidean spaces, where $d(x, y) := \|x - y\|_2 = \sqrt{\sum_i (x_i - y_i)^2}$. Generalizing gives the family of ℓ_p^n spaces, where we work over the set $\mathbb{R}^n$ and $d(x, y) := \|x - y\|_p = (\sum_i |x_i - y_i|^p)^{1/p}$. One can show that in the limit as $p \to \infty$, this expression tends to $\max_i |x_i - y_i|$. This space is denoted ℓ_∞^n.

Suppose (X, d) is isometrically embeddable into ℓ_1 (that is, ℓ_1^n for some n). Is d isometrically embeddable into ℓ_2 as well? Not necessarily. Here we claim that ℓ_2-embeddable metrics are only a subset of ℓ_1-embeddable metrics, which in turn are a subset of ℓ_∞ metrics. In fact, we put forth the following lemma:

Lemma 3 *Any finite metric space (V, d) is isometrically embeddable in $\ell_\infty^{|V|}$.*

Proof: For notational purposes, let $V = \{1, 2, \ldots, n\}$. The mapping $\phi \colon V \to \mathbb{R}^{|V|}$ is given by

$$\phi(v) = (d(1, v), d(2, v), \ldots, d(n, v)).$$

Using properties of metrics, we have

$$\begin{aligned} d(u,v) = &= |d(u,u) - d(u,v)| \\ &\leq \max_{i \in V} |d(i,u) - d(i,v)| \\ &= \|\phi(u) - \phi(v)\|_\infty \\ &= \ell_\infty(\phi(u), \phi(v)). \end{aligned}$$

On the other hand, the triangle inequality gives

$$\begin{aligned} (\phi(u) - \phi(v))_i = d(i,u) - d(i,v) \leq d(u,v) \\ (\phi(v) - \phi(u))_i = d(i,v) - d(i,u) \leq d(u,v) \end{aligned}$$

for all i, so $\ell_\infty(\phi(u), \phi(v)) = \max_{i \in V} |(\phi(u) - \phi(v))_i| \leq d(u,v)$.

□

Remark 8 *The ℓ_2-embeddable finite metrics are ℓ_1-embeddable.*

The proof for this will be revisited in the next lecture. For now we return to the Multicommodity-Cut problem, and how metrics can help us get an approximation algorithm for it.

6 Back to multicommodity cut

In the notation of metric spaces, we have the following. ("$M \leq M'$" means "M is isometrically embeddable in M'")

Theorem 4

$$\begin{aligned} \alpha^* &= \min_{\ell \colon (V,\ell) \leq \ell_\infty} \frac{\sum_{e=(i,j) \in E} u(e)\ell(i,j)}{\sum_{i=1}^k f_i \ell(s_i, t_i)} \\ \beta^* &= \min_{\ell \colon (V,\ell) \leq \ell_1} \frac{\sum_{e=(i,j) \in E} u(e)\ell(i,j)}{\sum_{i=1}^k f_i \ell(s_i, t_i)} \end{aligned}$$

(Note that the only difference between these two expressions is the class of metrics in which we permit (V, ℓ) to reside. Thus, since α^* minimizes over a larger space, we have $\alpha^* \leq \beta^*$ immediately—as we expect.) In the following lecture, we show an algorithm to compute β^* approximately, making use of the above.

References

[1] M.X. Goemans and D.P. Williamson, Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming, J. ACM, 42, 1115–1145, 1995.

[2] U. Feige and G. Schechtman, On the optimality of the random hyperplane rounding technique for MAX CUT, Algorithms, 2000.

[3] J. Håstad, Some optimal inapproximability results, J. ACM, 48, 798–869, 2001.