PRINCETON U. SP'02 COS 598B: ALGORITHMS AND COMPLEXITY

Lecture 8: Approximating Min UnCut and Min-2CNF Deletion

Lecturer: *Sanjeev Arora* Scribe:*Konstantin Makarychev*

In this lecture we will present $\sqrt{\log n}$-approximation algorithms for MIN UNCUT and MIN-2CNF DELETION [1].

1 Approximating Min UnCut

We first formulate the MIN UNCUT problem.

DEFINITION 1 (MIN UNCUT PROBLEM) *Given a graph $G = (V, E)$, find a cut that minimizes the number of uncut edges i.e. the number of edges within each part.*

REMARK 1 The MIN UNCUT problem is a complement to the MAXCUT problem: The sum of the number of cut edges and uncut edges is equal to the total number of edges in the graph.

We will reduce MIN UNCUT to an alternate problem which will be convenient for our purposes. Let us assume that the vertices of the graph G are numbers $1, \ldots, n$. Construct a new graph G' on the set of vertices $\{-n, \ldots, -1\} \cup \{1, \ldots, n\}$. We connect two vertices i and j with an edge in G' *iff* i and $-j$ or $-i$ and j are connected with an edge in G. We want to find a symmetric cut $(S', T' = -S')$ in the graph G' which minimizes the number of cut edges, where $-S' \equiv \{-i : i \in S'\}$. Every cut (S, T) in the graph G corresponds to the cut $(S \cup (-T), (-S) \cup T)$ in G'. Indeed if an edge (i, j) is uncut in G, say $i, j \in S$, then the corresponding edges $(i, -j)$ and $(-i, j)$ are cut in G': $i, j \in S$ and $-i, -j \in T$. If (i, j) is cut in G, the edges $(i, -j)$ and $(-i, j)$ are uncut in G'. On the other hand, given a cut (S', T') in G' the corresponding cut (S, T) in G is defined as follows $S = \{i \in S' : i \geq 0\}$; $T = \{i \in T' : i \geq 0\}$. Thus MIN UNCUT is equivalent to the following problem:

DEFINITION 2 *Given a graph $G = (V, E)$, where $V = \{-n, \ldots, -1\} \cup \{-1, \ldots, -n\}$ find a cut $(S, T = -S)$ that minimizes the number of cut edges i.e. the number of edges going from the part S to T.*

1.1 SDP relaxation

Write an SDP (vector program) relaxation for the new problem:

$$\min \frac{1}{4} \sum_{(i,j) \in E} |v_i - v_j|^2 \tag{1}$$

$$\text{s.t.} \quad |v_i|^2 = 1 \qquad \forall i \in V \tag{2}$$

$$|v_i - v_j|^2 \leq |v_i - v_k|^2 + |v_k - v_j|^2 \qquad \forall i, j, k \in V \tag{3}$$

$$v_i = -v_{-i} \qquad \forall i \in V \tag{4}$$

This SDP is indeed a relaxation. Every cut $(S, T = -S)$ corresponds to a feasible set of vectors:

$$v_i = \begin{cases} v_0 & \text{, if } i \in S; \\ -v_0 & \text{, if } i \in T; \end{cases}$$

where v_0 is a fixed unit vector. The objective function is equal to the number of cut edges.

Define the volume of a set $M \subset V$ to be

$$\mathrm{vol}(M) = \sum_{\substack{(i,j)\in E \\ i,j\in M}} |v_i - v_j|^2.$$

In other words, the volume of a set is equal the contribution of the set to the SDP value multiplied by four. Similarly the volume of an edge (i, j) is $|v_i - v_j|^2$.

1.2 Applying the ARV separation theorem

We now sketch the algorithm for partitioning the graph. First we solve the SDP relaxation and get a set of unit vectors (unit-ℓ_2^2 representation). Then we apply the ARV separation theorem [2] and get two $\Delta = \Omega(1/\sqrt{\log n})$-separated sets S^* and T^* w.r.t the squared Euclidean distance (ℓ_2^2). Let us show why we can use the separation theorem. Our SDP constraints contain all the constraints of ARV, except for the spreading constraint; which follows from existence of antipodal vectors:

Lemma 1

Every symmetric set of unit vectors is $1/3$*-spread (for* $n \geq 9$*), that is:*

$$\frac{1}{n^2}\sum_{i,j}(v_i - v_j)^2 \geq \frac{1}{3}$$

Proof:

$$\begin{aligned} &\sum_{i<j} |v_i - v_j|^2 \\ &\quad = \sum_{i>0}\sum_{j>i} \big(|v_i - v_j|^2 + |v_i - v_{-j}|^2 \\ &\qquad\qquad + |v_{-i} - v_j|^2 + |v_{-i} - v_{-j}|^2\big) \\ &\quad = \sum_{i>0}\sum_{j>i} 8 = 4(n-1)n \geq 4 \cdot \frac{1}{3} \cdot \frac{2}{3} \cdot 4n^2. \end{aligned}$$

□

Another observation is that if the set of vectors given to the separation algorithm is symmetric, the sets S^* and T^* are also symmetric *i.e.* $S^* = -T^*$. Indeed, the ARV algorithms has two phases:

- In the first phase, we pick a random unit vector u and set

$$S_I = \{v_i : \langle v_i, u\rangle \geq c/\sqrt{n}\};$$
$$T_I = \{v_i : \langle v_i, u\rangle \leq -c/\sqrt{n}\};$$

  where c is some constant. Clearly, $S_I = -T_I$.

- In the second phase, we remove a maximum matching M of vectors (v_i, v_j) s.t. the ℓ_2^2 distance between v_i and v_j is bigger than Δ. Generally speaking, this matching does not have to be symmetric , but the algorithm can always pick a symmetric matching since if $(v_i - v_j)^2 \geq \Delta$, then $(v_{-i} - v_{-j})^2 = ((-v_i) - (-v_j))^2 \geq \Delta$. Therefore, the second step of the algorithm (after some tweaking) also produces symmetric sets S^* and T^*.

1.3 Growing Balls

Let us now consider the t-neighborhoods of S^* and T^* for $0 \leq t < \Delta/2$:

$$N_t(S^*) = \left\{ v_i : d(v_i, S^*) \equiv \min_{u \in S^*} (v_i - u)^2 \leq t \right\};$$
$$N_t(T^*) = \left\{ v_i : d(v_i, T^*) \equiv \min_{u \in T^*} (v_i - u)^2 \leq t \right\}.$$

Similarly to what we saw in ARV, for some t_0 the number of outgoing edges from $N_{t_0}(S^*)$ plus the number of incoming edges to $N_{t_0}(T^*)$ is at most $(4/\Delta)\,\mathrm{vol}(V)$. We set $S_1 = N_{t_0}(S^*)$ and $T_1 = N_{t_0}(T^*)$. Note that the sets S_1 and T_1 are symmetric and disjoint (since S^* and T^* are Δ-separated and $t < \Delta/2$).

1.4 Recursion

We apply the same procedure to the remaining part $R_1 = V \setminus (S_1 \cup T_1)$ and get sets S_2 and $T_2 = -S_2$ *etc*. Finally we set $S = \cup_i S_i$, $T = \cup_i T_i$ and return the cut (S, T). Since all sets S_i and T_i are symmetric, the cut (S, T) is also symmetric. The size of the cut is less than or equal to the sum of the number of outgoing edges from S_1, S_2, *etc* plus the number of incoming edges to T_1, T_2, *etc.* Which is bounded by

$$O(\sqrt{\log n}) \cdot (\mathrm{vol}(V) + \mathrm{vol}(R_1) + \mathrm{vol}(R_2) + \ldots).$$

In order this sum to be $O(\sqrt{\log n}\,\mathrm{vol}(G))$, it suffices that the volumes of R_i decrease geometrically. In other words, the sets S_i and T_i should contain a constant fraction of the volume of R_{i-1} at each iteration of the algorithm.

To guarantee this we assign to each vertex weight proportional to the volume of the outgoing edges from this vertex. Then the volume of every set of vertices is approximately (up to a factor of 2) is equal to the weight of the set. In the algorithm described above, we shall use the weighted version of the separation algorithm (instead of the unweighted version). The weighted version finds symmetric Δ-separated sets S^* and T^* that contain a constant fraction of the weight (this approach is due to [4]). Therefore, the weights of R_i decrease geometrically and thus the volumes of R_i decrease also geometrically.

We now describe the weighted separation algorithm. The algorithm duplicates each vector the number of times proportional to its weight and then starts the ARV algorithm on the duplicate vectors. The ARV algorithm returns two sets S_{dup} and T_{dup}. If any duplicate of a vector v_i belongs to S_{dup} $[T_{dup}]$, we add v_i to S $[T]$. The total number of duplicate vectors is proportional to the total weight of the graph; the weight of S is proportional to the number of vectors in S; thus the weight of S is at least a constant fraction of the total weight of the graph.

2 Approximating Min-2CNF Deletion

Definition 3 (Min-2CNF Deletion Problem) *Consider boolean variables* $b_1, \ldots, b_n$ *and a set of constraints of the form* $b_i \vee b_j$, $\bar{b_i} \vee b_j$ *and* $\bar{b_i} \vee \bar{b_j}$. *The goal is to minimize the number of unsatisfied constraints.*

We first note that this problem can be reformulated in a similar form to Min Uncut. For each variable b_i we introduce a new variable b_{-i} and set $b_{-i} = \bar{b_i}$. Then we replace each constraint $b_i \vee b_j$ with two equivalent constraints $b_{-i} \to b_j$ and $b_{-j} \to b_i$. We now want to minimize the number of unsatisfied constraints of the new form. We consider the graph $G = (V, E)$, where $V = \{-n, \ldots, -1\} \cup \{1, \ldots, n\}$ and $(i, j) \in E$ *iff* there is a constraint $b_i \to b_j$.

We claim that Min-2CNF Deletion is equivalent to the problem of finding a minimal symmetric directed cut $(S, T = -S)$ in G. The symmetric cut gives us an assignment of truth values to variables in the original instance – one part corresponds to the variables set to true, and the other corresponds to those set to false. Note that the cut edges in the new problem correspond to constraints that are unsatisfied in the original instance.

We get the following equivalent definition.

Definition 4 *Given a* directed *graph* $G = (V, E)$, *where* $V = \{-n, \ldots, -1\} \cup \{-1, \ldots, -n\}$ *find a cut* $(S, T = -S)$ *that minimizes the number of edges going from* S *to* T.

We write an SDP relaxation for Min-2CNF Deletion:

$$\min \frac{1}{8} \sum_{(i,j)\in E} |v_i - v_j|^2 + |v_j - v_0|^2 - |v_i - v_0|^2 \tag{5}$$

$$\text{s.t.} \qquad |v_i|^2 = 1 \qquad \forall i \in V \tag{6}$$

$$|v_i - v_j|^2 \le |v_i - v_k|^2 + |v_k - v_j|^2 \qquad \forall i, j, k \in V \tag{7}$$

$$v_i = -v_{-i} \qquad \forall i \in V \tag{8}$$

where v_0 corresponds to the part S (true); and $-v_0$ corresponds to the part T (false). Note that this is indeed a valid relaxation. For every edge $i \to j$, we have the term $\frac{1}{8}(|v_i - v_j|^2 - |v_0 - v_i|^2 + |v_0 - v_j|^2)$ in the objective function. If $v_i = v_j$ or if $v_i = -v_0, v_j = v_0$ (*i.e.* the edge does not go from S to T), the value of this expression is 0. On the other hand, the value is 1 if $v_i = v_0, v_j = -v_0$ (*i.e.* the edge goes from S to T).

Instead of the ℓ_2^2 metric, we use the following distance function:

$$d(v_i, v_j) = \frac{1}{8} \left[|v_i - v_j|^2 + |v_j - v_0|^2 - |v_i - v_0|^2 \right].$$

This distance function

- is positive: $d(v_i, v_j) \ge 0$, this follows from the triangle inequality for ℓ_2^2.
- satisfies the triangle inequality $(\forall i, j, k \in V)$

$$\begin{aligned} d(v_i, v_k) + d(v_k, v_j) &= |v_i - v_k|^2 - |v_0 - v_i|^2 \\ &\quad + |v_0 - v_k|^2 + |v_k - v_j|^2 - |v_0 - v_k|^2 + |v_0 - v_j|^2 \\ &= |v_i - v_k|^2 + |v_k - v_j|^2 - |v_0 - v_i|^2 + |v_0 - v_j|^2 \\ &\ge |v_i - v_j|^2 - |v_0 - v_i|^2 + |v_0 - v_j|^2 = d(v_i, v_j) \end{aligned}$$

- is not symmetric: generally speaking $d(v_i, v_j) \neq d(v_j, v_i)$;
- if $d(v_i, v_j) = 0$, v_i does not necessary equal to v_j;

We call this distance function a directed semimetric.

2.1 Separation theorem for the directed semimetric

Replacing ℓ_2^2 metric in the Min UnCut approximation algorithm with the distance function d we get an approximation algorithm for Min 2 CNF Deletion.

The only missing step is a separation theorem for the distance d, which we describe now. First we find two symmetric sets S and T that are Δ-separated with respect to ℓ_2^2. Then we set

$$S^+ = \{v_i \in S : \langle v_i, v_0 \rangle \geq 0\}; \qquad T^+ = \{v_i \in T : \langle v_i, v_0 \rangle \geq 0\};$$
$$S^- = \{v_i \in S : \langle v_i, v_0 \rangle \leq 0\}; \qquad T^- = \{v_i \in T : \langle v_i, v_0 \rangle \leq 0\};$$

The sets S^+ and T^-; T^+ and S^- are $\Delta/8$-separated: If $v_i \in S^+$, $v_j \in T^-$, then

$$\begin{aligned} d(v_i, v_j) &= \frac{1}{8}\left[(v_i - v_j)^2 + (v_j - v_0)^2 - (v_i - v_0)^2\right] \\ &= \frac{1}{8}\left[(v_i - v_j)^2 - 2\langle v_j, v_0 \rangle + 2\langle v_i, v_0 \rangle\right] \\ &\geq \frac{(v_i - v_j)^2}{8} \geq \frac{\Delta}{8} \end{aligned}$$

Thus S^+ and T^- are $\Delta/8$-separated w.r.t the directed distance d. Similarly T^+ and S^- are $\Delta/8$-separated w.r.t the directed distance. Since $S^+ \cup S^- = S$, S^+ or S^- contains at least a half of all points. If S^+ contains a half of vertices of S, the algorithm returns the sets (S^+, T^-); otherwise (T^+, S^-).

References

[1] A. Agarwal, M. Charikar, K. Makarychev, Y. Makarychev, $O(\sqrt{\log n})$ approximation algorithms for Min UnCut, Min-2CNF Deletion, and directed cut problems. To appear in STOC 2005.

[2] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 222–231, 2004.

[3] B. Bollobás. *Combinatorics: Set Systems, Hypergraphs, Families of Vectors and Probabilistic Combinatorics*, pages 122–130. 1986.

[4] S. Chawla, A. Gupta, and H. Räcke. Approximations for generalized sparsest cut and embeddings of l_2 into l_1. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2005.

[5] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC)*, pages 698–707, 1993.

[6] G. Karakostas. A better approximation ratio for the vertex cover problem. In *Electronic Colloquium on Computational Complexity Report TR04-084*, 2004.

[7] J. R. Lee. On distance scales, embeddings, and efficient relaxations of the cut cone. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2005.