CS880: Approximations Algorithms

Scribe: Siddharth Barman **Lecturer:** Shuchi Chawla
Topic: SDP: Max-cut, Max-2-SAT **Date:** 03/27/07

In this lecture we give SDP (semi definite programming) based algorithms for the Max-cut and Max-2-SAT problem.

A semi definite program is a set of linear inequalities along with the constraint that the matrix of variables is positive semi definite. The general form of a SDP is as follows:

$$\begin{aligned} & \min AC \\ \text{subject to} \quad & A \cdot B_k \geq b_k \quad \forall k \\ & A \succeq 0 \end{aligned}$$

Where $A \succeq 0$ indicates that the variable matrix A is positive semi definite. Also note that the first constraint, $A \cdot B_k \geq b_k$, is a constraint over the linear combination of elements A_{ij} of matrix A i.e. $\sum_{ij} A_{ij}(B_k)_{ij} \geq b_k$.

An important observation is that a semi definite program is equivalent to a *vector program* where matrix A is composed of inner products of variable vectors. That is the above mentioned SDP is equivalent to the following vector program

$$\begin{aligned} & \min \text{linear funct. of } \langle v_i, v_j \rangle \\ \text{subject to} \quad & \text{linear constraints } over \langle v_i, v_j \rangle \\ & v_i \in \Re^n \text{ and} \\ & A_{ij} = v_i \cdot v_j \end{aligned}$$

We shall formulate vector programs for the Max-cut and Max-2-SAT problem, which essentially are semi definite programs, the general strategy then is to solve the SDP to obtain vector solutions for the variables. We then "extract" an integral solution from these vectors which preserve the objective function value to a good extent.

20.1 Max-cut

Problem Statement: Given a graph G with weights c_e on edges, find a cut $C \subseteq E$ of maximum weight which separates the vertices of G in two connected components. Here the weight of the cut C is defined as $\sum_{e \in C} c_e$.

First we mention that the direct linear programming based approach that we have applied for other cut problems does not go through for Max-cut. An LP relaxation for Max-cut is as follows:

$$\begin{aligned} & \max \sum c_e d_e \\ \text{subject to} \quad & \text{d is a metric} \\ & d(u,v) \leq 1 \quad \forall u, v \end{aligned}$$

But this program can be maximized by setting $d(u,v) = 1\ \forall u \neq v$ (which is a legitimate metric); resulting in objective function value equal to $\sum_{e \in E} c_e$. But this implies that all the edges are in the cut and the requirement that the graph is separated into only two components S and $V \setminus S$ is not met.

The important insight here is to restrict the form of the metric d to ensure that a sound cut is obtained. Specifically we wish to determine a metric which places the vertices of the graph either at 1 or at -1. That is for all vertices v we must have $x_v^2 = 1$. Thus d_e is either 0 or 2 and the objective function becomes $\max \sum \frac{c_e d_e}{2}$.

For the above mentioned metric we have $d_{uv}^2 = (x_u - x_v)^2 = x_u^2 + x_v^2 - 2x_u x_v$. But x_u^2 and x_v^2 are equal to 1 hence $d_{uv}^2 = 2(1 - x_u x_v)$. Also note that $d_{uv} \in \{0, 2\}$ thus $d_{uv} = d_{uv}^2/2$. Hence the Max-cut problem can be concisely stated as the following *quadratic program*:

$$\begin{aligned} & \max \textstyle\sum_{(u,v) \in E} \frac{c_e}{2}(1 - x_u x_v) \\ \text{s.t.} \quad & x_u^2 = 1\ \forall u \\ & x_u \in \Re \end{aligned}$$

We relax this to a vector program. In particular,

$$\begin{aligned} & \max \textstyle\sum_{(u,v) \in E} \frac{c_e}{2}(1 - x_u \cdot x_v) \\ \text{s.t.} \quad & x_u \cdot x_u = 1\ \forall u \\ & x_u \in \Re^n \end{aligned} \tag{20.1.1}$$

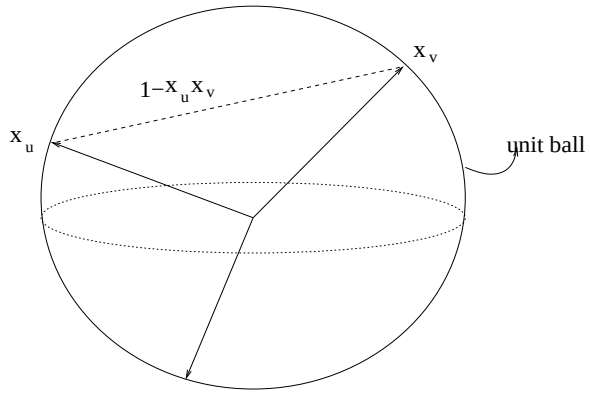

Figure 20.1.1: Unit ball in $\Re^n$

After solving this relaxed formulation we get vectors x_u on the unit ball (see Figure 20.1.1) and we wish to determine a cut in $\Re^n$ such that long edges are very likely in it. The idea is to consider a random hyperplane in $\Re^n$ passing through the origin; long edges which contribute more to the SDP solution have high probability of being cut by such a plane. Our cut is the set of edges that cross this hyperplane. Next we prove that this in fact achieves an approximation factor of 0.878.

Formally the Max-cut algorithm is as follows

1. Solve SDP 20.1.1 to obtain vectors $\{x_u\}$
2. Pick a random unit vector α in $\Re^n$
3. Output $\{v \mid x_v^T \alpha > 0\}$

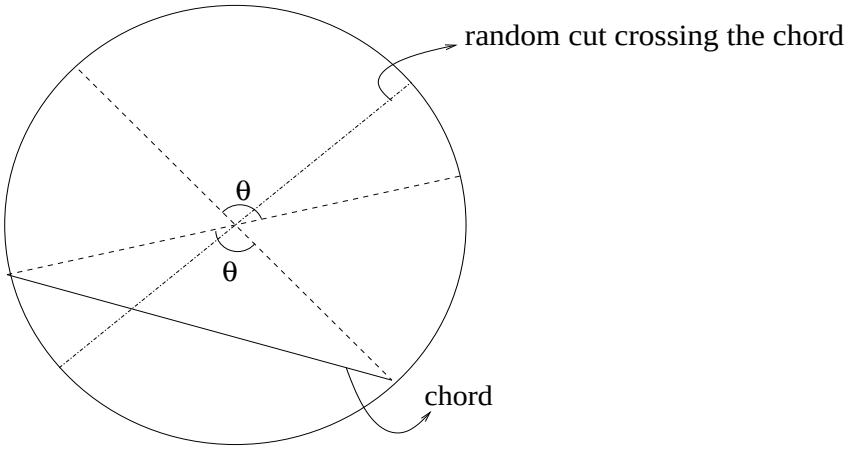

Figure 20.1.2: Chord with random cut

First we consider vectors in $\Re^2$ and then generalize to cuts in $\Re^n$. Consider a chord in a unit circle that subtends an angle of θ at the origin (see Figure 20.1.2). Note that in the present context selecting an angle between 0 and 2π, uniformly at random is equivalent to selecting a cut uniformly at random.

The probability that the chord (representing an edge) is cut is θ/π. As any angle selected in the marked region of Figure 20.1.2 would define a cut that intersects the chord and the total angle contained by the marked region is 2θ the probability is $2\theta/2\pi = \theta/\pi$.

As vectors x_u lie on the unit ball the square of the length of the chord/edge is $2(1-\cos\theta)$. Hence its contribution to SDP is $2c_e(1-\cos\theta)$. The expected contribution of the edge to our solution is $\frac{\theta}{\pi}c_e$. The maximal difference between the two could be $\gamma = \max_\theta \frac{\pi(1-\cos\theta)}{2\theta} = \frac{1}{0.878} \approx 1.12$. That is the contribution of any edge to the SDP can not be more than γ times its expected contribution to our solution.

The following lemma ensures that the above mentioned Max-cut algorithm achieves a 0.878 approximation.

Lemma 20.1.1 *For every edge $e = (u, v)$ in the graph, let θ_e be the angle between x_u and x_v, then the probability that we cut edge e is θ_e/π.*

Proof: Consider the plane defined by vectors x_u, x_v and the origin. The intersection of the hyperplane defined by random vector α and this plane is a line passing through the origin. Say this line is α'. Note that α' is picked from a spherically symmetric distribution hence as above the probability that (u, v) is cut is θ_e/π. ■

Note that step 2 of the algorithm is equivalent to picking a point uniformly at random from the surface of a unit sphere. This is accomplished by picking a point from an n dimensional Gaussian and then renormalizing it to be of unit length.

Feige and Schechtman [2] have shown that the integrality gap of this SDP is in fact 1/0.878. Also hardness of 0.878 for Max-cut has been shown under the unique games conjecture [3].

Integral solutions of the Max-cut problem satisfy an interesting property. In particular any integral solution sets $x_u \in \{-1, 1\}$ hence it must satisfy $d_{uv}^2 = 2d_{uv}$. This relation implies a triangle inequality for distance squares, that is $d_{uv}^2 \leq d_{uw}^2 + d_{wv}^2$. Such a relation is not necessary for all metrics but it holds for integral solutions. With this constraint we can formulate another stronger SDP for Max-cut:

$$\begin{aligned} & \max \textstyle\sum_{(u,v)\in E} \frac{c_e}{2}(1 - x_u x_v) \\ \text{s.t.} \quad & x_u^T x_u = 1 \ \forall u \\ & x_u \in \Re^n \\ & 1 - x_u x_v \leq 1 - x_u x_w + 1 - x_w x_v \ \ \forall u, v, w \end{aligned} \tag{20.1.2}$$

The last constraint can be simplified to $x_u x_w + x_w x_v \leq 1 + x_u x_v$. The solutions to such an SDP are special metrics called l_2^2 metrics:

Definition 20.1.2 *Metrics for which not only the distances but the square root of distances satisfy the triangle inequality are called* ***squared euclidian metrics*** *(negative type) and are denoted as* l_2^2.

Note that l_2^2 metric embed with distortion $O(\sqrt{\log n} \log \log n)$ in l_1 [1], which implies better approximation factor for some problems with such metric requirement. Though for Max-cut formulation 20.1.2 does not reduce the integrality gap but it does for some other problems, in particular for sparsest cut integrality gap is reduced to $\tilde{O}(\sqrt{\log n})$ [1].

20.2 Max-2-SAT

Problem Statement: Given a 2-CNF boolean formula we need to determine an assignment which maximizes the number of satisfied clauses. In the weighted version of the problem each clause is associated with a weight v_i and we need to determine an assignment which maximizes the sum of the values v_i for the satisfied clauses.

We formulate a SDP by considering vector v_x for every variable x, also $v_{\overline{x}}$ (for $\overline{x}$) is set to $-v_x$. For reference a true vector v_T is also defined, thus informally $v_x v_T$ is the extent to which x is true.

Now, clause $x \vee y$ contributes a value of 1 to the objective function if either x or y is true else it contributes a value of 0. For $v_x \in \{-1, 1\}$ this can be equivalently expressed as $\frac{3+v_x v_T + v_y v_T - v_x v_y}{4}$. As the expression is set to one when either $v_x v_T$ or $v_y v_T$ is set to one, and it is set to zero when v_x and v_T are of different signs. We relax v_x to vectors in $\Re^n$ to obtain the SDP:

$$\begin{aligned} & \max \textstyle\sum_{clauses \quad x \vee y} \frac{3+v_x v_T+v_y v_T-v_x v_y}{4} \\ \text{s.t.} \quad & v_x v_x = 1 \ \forall x \\ & v_x \in \Re^n \\ & v_T \text{ is the truth vector} \end{aligned} \tag{20.2.3}$$

The algorithm $\mathcal{A}$, for Max-2-SAT is as follows:

1. Pick a random hyperplane
2. Set everything on the side of v_T to true and rest to false

The analysis of Max-2-SAT follows closely from Max-cut. The objective function in this case can be rewritten as $\frac{1}{4}\sum((1+v_x v_T)+(1+v_y v_T)+(1-v_x v_y))$, thus the expression consists of terms of the form $1+v_i v_j$ and $1-v_i v_j$. Next we analyze the two terms separately.

As in the case of Max-cut the probability that the edge between i and j is cut is θ/π. The SDP contribution of the term is directly $1-v_i v_j$. As before with probability θ/π the plane separates the two vectors in which case the contribution of the term is 2 else the contribution is 0. Hence the expected contribution is $\frac{2\theta}{\pi}+0$. As before the ratio of the expected contribution to the SDP contribution is no less than $\gamma = \min \frac{2\theta}{\pi(1-\cos\theta)} = 0.878$.

Note that $v_i v_j = \cos\theta$, where θ is the angle between v_i and v_j. Hence for the term $1+v_i v_j$ the SDP contribution is $1+\cos\theta$. The expected contribution to the integral solution on the other hand is $0 \times \frac{\theta}{\pi} + 2\left(1-\frac{\theta}{\pi}\right)$. Hence the ratio in this case is $\frac{2(\pi-\theta)}{\pi(1+\cos\theta)}$. By setting $\theta' = \pi - \theta$ the ratio reduces to $\frac{2\theta'}{\pi(1-\cos\theta')}$, as before this is no less than 0.878. Hence the total expected contribution is no less than 0.878. So we have the following theorem:

Theorem 20.2.1 *The above mentioned algorithm $\mathcal{A}$ achieves an approximation factor of* 0.878 *for Max-2-SAT.*

References

[1] S. Arora, J.R. Lee, A. Naor Euclidean distortion and the sparsest cut. *In Proceedings on 37th Annual ACM Symposium on Theory of Computing (STOC)* (2005), pp: 553-562.

[2] U. Feige, G. Schechtman: On the integrality ratio of semidefinite relaxations of MAX CUT. *In Proceedings on 33rd Annual ACM Symposium on Theory of Computing (STOC)* (2001), pp: 433-442.

[3] S. Khot, G. Kindler, E. Mossel, R. O'Donnell Optimal Inapproximability Results for Max-Cut and Other 2-Variable CSPs? *In 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (2004), pp: 146-154.