CMPUT 675: Approximation Algorithms **Fall 2014**

Lecture 21 (Oct. 24): Max Cut SDP Gap and Max 2-SAT

Lecturer: Zachary Friggstad *Scribe: Chris Martin*

21.1 Near-Tight Analysis of the Max Cut SDP

Recall the MAX CUT problem. Given an undirected graph $G = (V, E)$ with edge weights $w(e), e \in E$, find $S \subseteq V$ that maximizes $\sum_{e\in\delta(S)} w(e)$. The following strict quadratic program was presented for this problem in the previous lecture. For each $i \in V$, we use the interpretation that $v_i = 1$ means $i \in S$ and $v_i = -1$ means $i \notin S$.

$$\begin{aligned}\text{maximize:} \quad & \sum_{(i,j)\in E} w(i,j) \cdot \frac{1 - v_i v_j}{2} && (\textbf{MAXCUT-QP}) \\ \text{subject to:} \quad & v_i v_i = 1, \quad \forall i \in V && (21.1)\end{aligned}$$

We relaxed this to an SDP by replacing each v_i with a vector $\mathbf{v}_i \in \mathbb{R}^n$ and replacing the multiplication of variables with the dot product $\circ$.

$$\begin{aligned}\text{maximize:} \quad & \sum_{(i,j)\in E} w(i,j) \cdot \frac{1 - \mathbf{v}_i \circ \mathbf{v}_j}{2} && (\textbf{MAXCUT-SDP}) \\ \text{subject to:} \quad & \mathbf{v}_i \circ \mathbf{v}_i = 1, \quad \forall i \in V && (21.2)\end{aligned}$$

We saw a simple randomized rounding algorithm that rounds an optimal solution to (**MAXCUT-SDP**) to an integer solution with expected value at least $\alpha \cdot OPT_{\text{SDP}}$ where

$$\alpha := \min_{0\le\theta\le2\pi} \frac{2\theta}{\pi(1-\cos(\theta)} \ge 0.87856.$$

There is a simple example for MAX CUT that shows our analysis is *nearly* tight.

Consider $G = C_5$, the cycle on 5 nodes $\{0, 1, 2, 3, 4\}$, with all edge weights being 1. The optimum solution is 4 with an optimum cut being, say, $S = \{1, 3\}$ with $\delta(S) = \{(0, 1), (1, 2), (2, 3), (3, 4)\}$ (we can do no better because C_5 is not bipartite). Next we define a feasible SDP solution with value greater than 4.

Let $\theta = 4\pi/5$. For $0 \le i \le 4$, let $\overline{\mathbf{v}}_i = (\cos(i\theta), \sin(i\theta), 0, 0, 0)$. Note that $\overline{\mathbf{v}}_i \circ \overline{\mathbf{v}}_i = 1$ for all i, since $\cos^2(i\theta) + \sin^2(i\theta) = 1$. So, this is a feasible solution to **MAXCUT-SDP**. The angle between any two nodes corresponding to an edge $(i, i+1)$ is θ, so $\overline{\mathbf{v}}_i \circ \overline{\mathbf{v}}_{i+1} = ||\overline{\mathbf{v}}_i|| \cdot ||\overline{\mathbf{v}}_{i+1}|| \cdot \cos(\theta) = \cos(\theta)$. Thus,

$$\begin{aligned}OPT_{SDP} &\ge \sum_{i=0}^{4} \frac{1-\cos(\theta)}{2} \\ &= \frac{5}{2} \cdot \left(1 - \cos\left(\frac{4\pi}{5}\right)\right) \\ &\ge 4.52254.\end{aligned}$$

Therefore,

$$\frac{OPT}{OPT_{SDP}} \leq \frac{4}{\frac{5}{2} \cdot \left(1 - \cos\left(\frac{4\pi}{5}\right)\right)} \leq 0.88445$$

which is very close to α. So, we know that our analysis of the integrality gap is nearly tight.

In fact, our initial integrality gap analysis is tight. There are more sophisticated examples showing that for any constant $c > \alpha$ that the integrality gap is at most c [FS02].

21.2 Max 2-SAT

The Max 2-SAT problem is as follows. Given 2-CNF clauses $C_1, \cdots, C_m$ over variables $x_1, \cdots x_n$, and weights for each clause $w(C)$, choose a boolean assignment of the variables that maximizes the total weight of satisfied clauses. An interesting aspect of this problem is that there is a polynomial-time algorithm that determines if all clauses can be satisfied [APT79]. However, if an instance is not satisfiable then the algorithm in [APT79] does not give us much of an idea on the maximum number that can be satisfied; this remains NP-hard.

Converting this problem into a strict quadratic program takes a bit of insight. For example, we cannot simply associate a variable v_i for each x_i and say $v_i = 1$ corresponds to $x_i =$ True. This is because a quadratic term of the form $v_i \cdot v_j$ cannot distinguish between the value $v_i \cdot v_j$ or $(-v_i) \cdot (-v_j)$. This was not a problem with Max Cut because, intuitively, negating all variables simply swapped the solution S to $V - S$ (which cut the same edges).

We will add a special variable which we call v_0 and define a truth assignment relative to v_0. That is, in any setting of the values $v_0, v_1, \ldots, v_n \in \{-1, +1\}$ we will set $x_i =$ True iff $v_0 = v_i$. Thus, negating all v_i variables does not change the associated truth assignment.

With the understanding that each v_i will take $\{-1, +1\}$ values, we can model the clause $(x_1 \vee \overline{x_2})$ with the following expression:

$$1 - \frac{1 - v_0 v_1}{2} \cdot \frac{1 + v_0 v_2}{2} \tag{21.3}$$

Note that this is 1 iff $(v_0 = v_1 \vee v_0 \neq v_2)$, as desired. Expanding this gives the following:

$$= 1 - \frac{1}{4}(1 - v_0 v_1 + v_0 v_2 - v_0^2 v_1 v_2) \tag{21.4}$$

$$= 1 - \frac{1}{4}(1 - v_0 v_1 + v_0 v_2 - v_1 v_2) \tag{21.5}$$

$$= \frac{1}{4}(1 - v_0 v_1) + \frac{1}{4}(1 + v_0 v_2) + \frac{1}{4}(1 - v_1 v_2). \tag{21.6}$$

Note that (21.5) holds because $v_0 \in \{1, -1\}$. Summing (21.6) over all clauses (some of the $\pm$ signs in front of the quadratic terms may differ, depending on whether the literals in the clause appear positively or negatively) with weights multiplied through gives the weight of satisfied clauses being expressed by

$$\sum_{0 \leq i < j \leq n} a_{ij}(1 + v_i v_j) + b_{ij}(1 - v_i v_j)$$

where the values a_{ij}, b_{ij} are constants ≥ 0. This will become our objective function for the quadratic program.

$$\text{maximize:} \quad \sum_{0 \leq i < j \leq n} a_{ij}(1 + v_i v_j) + b_{ij}(1 - v_i v_j) \qquad \textbf{(MAX-2SAT-QP)}$$

$$\text{subject to:} \quad v_i v_i = 1, \quad 0 \leq i \leq n \tag{21.7}$$

As in the SDP for MAX CUT, we will replace the variables v_i with vectors $\mathbf{v}_i \in \mathbb{R}^{n+1}$ and replace the products with vector dot products to get the following SDP.

$$\text{maximize:} \quad \sum_{0 \le i < j \le n} a_{ij}(1 + \mathbf{v}_i \circ \mathbf{v}_j) + b_{ij}(1 - \mathbf{v}_i \circ \mathbf{v}_j) \qquad (\textbf{MAX-2SAT-SDP})$$

$$\text{subject to:} \quad \mathbf{v}_i \circ \mathbf{v}_i = 1, \quad 0 \le i \le n \qquad (21.8)$$

We round **MAX-2SAT-SDP** using an algorithm similar to the one presented in the previous lecture for MAX CUT.

1. Solve **MAX-2SAT-SDP** to get vectors $\mathbf{v}_i^*$, $0 \le i \le n$.
2. Let $\mathbf{r} \in \mathbb{R}^{n+1}$ be a random unit vector.
3. Define values $y_i, 0 \le i \le n$ by
$$y_i = \begin{cases} 1 & \text{if } \mathbf{v}_i^* \circ \mathbf{r} \ge 0 \\ -1 & \text{if } \mathbf{v}_i^* \circ \mathbf{r} < 0 \end{cases}$$
4. For each $1 \le i \le n$, set
$$x_i := \begin{cases} \text{TRUE} & \text{if } y_0 \cdot y_i = 1 \\ \text{FALSE} & \text{if } y_0 \cdot y_i = -1 \end{cases}$$

Claim 1 *The expected weight of satisfied clauses is $\alpha \cdot OPT_{SDP}$.*

Proof. A clause of the form, say, $x_i \vee \overline{x_j}$ is satisfied if and only if $\frac{1}{4}(1+y_0 \cdot y_i) + \frac{1}{4}(1-y_0 \cdot y_j) + \frac{1}{4}(1-y_i \cdot y_j) = 1$ (otherwise the expression is 0) where the y_i are the values constructed in the algorithm. Thus, by how we collected the quadratic terms we have that the value of the truth assignment is exactly

$$\sum_{0 \le i < j \le n} a_{ij}(1 + y_i \cdot y_j) + b_{ij}(1 - y_i \cdot y_j).$$

Let θ_{ij} be the angle between $\mathbf{v}_i^*$ and $\mathbf{v}_j^*$. From the MAX CUT analysis, we know $\Pr[y_i \cdot y_j = -1] = \frac{\theta_{ij}}{\pi}$. This also shows $\Pr[y_i \cdot y_j = 1] = 1 - \frac{\theta_{ij}}{\pi}$. Therefore,

$$\mathrm{E}[y_i \cdot y_j] = \Pr[y_i \cdot y_j = 1] \cdot 1 + \Pr[y_i \cdot y_j = -1] \cdot (-1) = \left(1 - \frac{\theta_{ij}}{\pi}\right) \cdot 1 + \left(\frac{\theta_{ij}}{\pi}\right) \cdot (-1) = 1 - \frac{2\theta_{ij}}{\pi}.$$

By linearity of expectation, the expected weight of satisfied clauses is exactly

$$\begin{aligned}
& \mathrm{E}\left[\sum_{0 \le i < j \le n} a_{ij} \cdot (1 + y_i \cdot y_j) + b_{ij} \cdot (1 - y_i \cdot y_j)\right] \\
= & \sum_{0 \le i < j \le n} a_{ij}(1 + \mathrm{E}[y_i \cdot y_j]) + b_{ij}(1 - \mathrm{E}[y_i \cdot y_j]) \\
= & \sum_{0 \le i < j \le n} a_{ij} \cdot \left(2 - \frac{2\theta_{ij}}{\pi}\right) + b_{ij} \cdot \frac{2\theta_{ij}}{\pi} \\
= & \sum_{0 \le i < j \le n} 2 \cdot a_{ij} \cdot \left(1 - \frac{\theta_{ij}}{\pi}\right) + 2 \cdot b_{ij} \cdot \frac{\theta_{ij}}{\pi} \qquad (21.9)
\end{aligned}$$

Recall that for any $\theta \in [0, \pi]$ we have

$$\frac{\theta}{\pi} \geq \alpha \cdot \frac{1 - \cos(\theta)}{2}$$

where $\alpha > 0.87856$ is the constant in the Max Cut analysis. One can also show that for $\theta \in [0, \pi]$ we also have

$$1 - \frac{\theta}{\pi} \geq \alpha \cdot \frac{1 + \cos(\theta)}{2}.$$

Using these bounds, the fact that $\cos(\theta_{ij}) = \mathbf{v}_i^* \circ \mathbf{v}_j^*$, and that all a_{ij} and b_{ij} coefficients are nonnegative we conclude by bounding expression (21.9) as follows:

$$\sum_{0 \leq i < j \leq n} 2 \cdot a_{ij} \cdot \left(1 - \frac{\theta_{ij}}{\pi}\right) + 2 \cdot b_{ij} \cdot \frac{\theta_{ij}}{\pi} \geq \sum_{0 \leq i < j \leq n} a_{ij} \cdot \alpha \cdot (1 + \mathbf{v}_i^* \circ \mathbf{v}_j^*) + b_{ij} \cdot \alpha \cdot (1 - \mathbf{v}_i^* \circ \mathbf{v}_j^*) = \alpha \cdot OPT_{\text{SDP}}.$$

■

21.3 Discussion

One can better with Max-2SAT. There is a 0.94-approximation via SDP techniques [LLZ02] and this is nearly tight. Under the Unique Games Conjecture, there is no 0.943-approximation [K+07]. Assuming only P $\neq$ NP, there is no $21/22 \approx 0.954$ approximation for Max-2SAT [H01].

In general, SDPs perform very well for Constraint Satisfiaction Problems where each clause is over only two variables (i.e. a 2-CSP) and where the variables take values from a constant-size domain. For example, Max-2SAT and Max Cut are two such problems. Another is the the 3-colouring variant where we must colour each vertex one of three colours and the goal is to maximize the number of edges that whose endpoints are coloured differently. This is a 2-CSP with domain size 3. A random colouring satisfies each constraint with probability 2/3, leading to a randomized 2/3-approximation. However, SDP-based techniques approximate this problem within a constant better than 2/3. In general, any 2-CSP over a constant-size domain can be approximated better than the random assignment using SDPs [H08].

Finally, under the Unique Games Conjecture (UGC) we have a complete characterization of constraint satisfaction problems of constant constraint size over constant domains (including problems like Max-3SAT). Namely, Raghavendra describes an SDP relaxation for such CSPs and proves, in some sense, that the if integrality gap for this relaxation is γ then it is UGC-hard to approximate the problem within any constant better than γ [R08]. He also describes an algorithm that rounds the SDP and obtains an approximation guarantee essentially equal to the integrality gap γ.

References

APT79 B. Aspvall, M. Plass and R. Tarjan, A linear-time algorithm for testing the truth of certain quantified boolean formulas, *Information Processing Letters 8(3)*, 121–123, 1979.

FS02 U. Feige and G. Schechtman, On the optimality of the random hyperplane rounding technique for MAX-CUT, *Random Structures and Algorithms*, 20:403–440, 2002.

H01 J. Håstad, Some optimal inapproximability results, *Journal of the ACM*, 48:798–869, 2001.

H08 J. Håstad, Every 2-CSP allows nontrivial approximation, *Computational Complexity*, 17(4):549–566, 2008.

K+07 S. Khot, G. Kindler, E. Mossel, and R. O'Donnell, Optimal inapproximability results for MAX-CUT and Other 2-Variable CSPs?, *SIAM Journal on Computing*, 37(1):319–357, 2007.

LLZ02 M. Lewin, D. Livnat, and U. Zwick, Improved rounding techniques for the MAX-2SAT and MAX-DICUT problems. In Proceedings of IPCO, 2002.

R08 P. Raghavendra, Optimal algorithms and inapproximability results for every CSP?, In Proceedings of STOC, 2008.