**Advanced Algorithms**

# Lecture 6: Randomness, Max Cut, and Non-linear Constraints

*Lecturer: Nathan Klein*

## 1 Max Cut

In the Max Cut problem, we are given a graph $G = (V, E)$ and we want to find a set $S \subseteq V$ so that $|\delta(S)|$ is maximized.

### 1.1 Expectation

Remember given a *discrete sample space* $\Omega$ with a *probability function* $\mathbb{P} : 2^{\Omega} \to [0, 1]$ (i.e. from subsets of $\Omega$ to $[0, 1]$), a *random variable* is a function $X : \Omega \to \mathbb{R}$ and its expectation $\mathbb{E}[X]$ is computed:

$$\mathbb{E}[X] = \sum_{\omega \in \Omega} \mathbb{P}[\omega] X(\omega)$$

Where $\text{supp}(X)$ is the set of numbers in $\mathbb{R}$ with $\mathbb{P}[X = x] > 0$, one can show:

$$\mathbb{E}[X] = \sum_{x \in \text{supp}(X)} x \cdot \mathbb{P}[X = x]$$

A key fact about expectation is as follows:

**Lemma 1.1** (Linearity of Expectation)**.** *For any collection of random variables $X_1, \ldots, X_n$,*

$$\mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{E}[X_i]$$

*Proof.*

$$\mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{\omega \in \Omega} \mathbb{P}[\omega] \sum_{i=1}^{n} X_i(\omega) = \sum_{i=1}^{n} \sum_{\omega \in \Omega} \mathbb{P}[\omega] X_i(\omega) = \sum_{i=1}^{n} \mathbb{E}[X_i] \qquad \square$$

### 1.2 Simple Approximations for Max Cut

My favorite $\frac{1}{2}$ approximation for Max Cut is the following randomized algorithm: construct $S$ by letting $v \in S$ with probability $\frac{1}{2}$ independently for every vertex $v$.

**Lemma 1.2.** *For every edge $e$, $\mathbb{P}[e \in \delta(S)] \geq \frac{1}{2}$.*

*Proof.*

$$\begin{aligned}\mathbb{P}[e \in \delta(S)] &= \mathbb{P}[u \in S, v \notin S] + \mathbb{P}[u \notin S, v \in S] \\ &= \mathbb{P}[u \in S]\, \mathbb{P}[v \notin S] + \mathbb{P}[u \notin S]\, \mathbb{P}[v \in S] = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}\end{aligned}$$

where we used that the events $v \in S$ and $u \in S$ are independent. $\square$

So, we can show:

**Lemma 1.3.** *Where $S$ is the random set produced by the algorithm and $OPT$ is the number of edges in the optimal cut,*

$$\mathbb{E}\left[\delta(S)\right] \geq \frac{1}{2}OPT$$

*Proof.* Let $X_e$ be a random variable indicating if $e \in \delta(S)$. Then $|\delta(S)| = \sum_{e \in E} X_e$. So:

$$\begin{aligned}
\mathbb{E}\left[|\delta(S)|\right] = \mathbb{E}\left[\sum_{e \in E} X_e\right] &= \sum_{e \in E} \mathbb{E}\left[X_e\right] && \text{By Linearity of Expectation} \\
&= \sum_{e \in E} \frac{1}{2} = \frac{1}{2}|E| && \text{By Lemma 1.2} \\
&\leq \frac{1}{2}OPT
\end{aligned}$$

since certainly OPT cuts at most every edge! □

This is what is called a *randomized* $\frac{1}{2}$ approximation, because it produces a cut of size at least $\frac{1}{2}OPT$ in expectation. In most settings, a randomized $\alpha$ approximation is considered just as good as a deterministic one because we can run it many times and with high probability one will be close to the guaranteed $\alpha$ factor.

### 1.3 LP for Max Cut

It's reasonable to now try to construct an LP which will improve upon this $\frac{1}{2}$ approximation. Here is the natural idea: if $x_v \in \{0,1\}$ encodes whether $v \in S$, you need to encode that $y_{\{u,v\}}$ (the indicator of whether $\{u,v\} \in \delta(S)$) has the property $y_{\{u,v\}} \leq |x_u - x_v|$. You can do so as follows:

$$\begin{aligned}
\max \quad & \sum_{\{u,v\} \in E} y_{\{u,v\}} \\
\text{s.t.} \quad & y_{\{u,v\}} \leq x_u + x_v && \forall e = \{u,v\} \in E \\
& y_{\{u,v\}} \leq 2 - x_u - x_v && \forall e = \{u,v\} \in E \\
& x_v \in \{0,1\} && \forall v \in V \\
& y_{\{u,v\}} \in \{0,1\} && \forall \{u,v\} \in E
\end{aligned} \tag{1}$$

Unfortunately, the relaxed LP has an integrality gap of 2. The reason is that setting $x_v = \frac{1}{2}$ for all $v \in V$ and $y_{\{u,v\}} = 1$ for all $e \in E$ is always feasible. So, the LP will *always* report a value of $|E|$, which is completely trivial!

### 1.4 Nonlinear Constraints

Instead of using $|x_u - x_v|$ to be the function which tells us whether $\{u,v\}$ is cut, we could try something else. Suppose we instead let $x_v = 1$ if $v \in S$ and $x_v = -1$ otherwise. Now, instead of the constraints above on $y_{\{u,v\}}$, let's replace it with a non-linear constraint: $y_{uv} = x_u x_v$, where for ease of notation I'll drop the brackets in $y_{\{u,v\}}$. This gives us the following program with non-linear constraints:

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

Here's another way to write the same program:

$$\begin{aligned} \max \quad & \sum_{\{u,v\}\in E} \frac{1}{2}(1-y_{uv}) \\ \text{s.t.} \quad & y_{uv} = x_u x_v && \forall u,v \in V \\ & y_{vv} = 1 && \forall v \in V \end{aligned}$$

This is equivalent since $y_{vv} = x_v^2 = 1$ if and only if $x_v \in \{-1,1\}$. What's interesting now is that there is only one problematic set of constraints: $y_{uv} = x_u x_v$, and we don't have to worry about the integer versus linear program aspect.

### 1.5 Solving this Program

This is simply not a linear program, so what are we supposed to do? **Let's just delete the constraint $y_{uv} = x_u x_v$, solve the LP, and hope we can implement a separation oracle.** Of course, we'll fail, because solving this program is equivalent to solving Max Cut and is therefore NP-Hard. But maybe we can get close. So here's the new LP:

$$\begin{aligned} \max \quad & \sum_{\{u,v\}\in E} \frac{1}{2}(1-y_{uv}) \\ \text{s.t.} \quad & y_{vv} = 1 && \forall v \in V \end{aligned}$$

This is a bit odd: now the $y_{uv}$ values are unconstrained. So the LP can return, for example, $y_{uv} = -1000$ for all $y_{uv}$ and still set $y_{vv} = 1$ for all $v \in V$. This will have a huge objective value.

But it's easy to refute these values, since *we* know that it must be the case that $y_{uv} = x_u x_v$ for some $x_u, x_v$ with $x_u^2 = x_v^2 = 1$. Remember, we're trying to be the separation oracle ourselves. How can we prove that $y_{uv} = -1000$ is not a valid solution using a linear constraint over the $y_{uv}$ variables?

You might say, well, it's obvious: it needs to be the case that $-1 \le y_{uv} \le 1$, since that's true of an integral solution. But since we want to automate this process with a separation oracle, let's try to figure out how you would prove that formally. If you play around a little bit, you might notice that you can use the inequality $(x_u + x_v)^2 \ge 0$, which is just a true mathematical fact. Writing it out, we get

$$0 \le (x_u + x_v)^2 = x_u^2 + x_v^2 + 2x_u x_v = 2 + 2y_{uv}$$

where we used that $y_{vv} = 1$ for all $v \in V$. Rewriting this, our first inequality says $y_{uv} \ge -1$. So if we throw in these constraints, at least our LP now can get at most value $|E|$.

So, we continue. Now maybe the LP returns a solution which sets $y_{uv} = -1$ for all edges in a triangle $u, v, w$. This is obviously wrong, so let's see if we can refute this in a similar fashion. The first thing we might try is the following:

$$(x_u + x_v + x_w)^2 \geq 0$$

Which, expanded and replacing squared terms with 1 is:

$$3 + 2x_u x_v + 2x_v x_w + 2x_u x_w \geq 0$$

Which is a refutation, giving us $y_{uv} + y_{vw} + y_{uw} \geq -\frac{3}{2}$. Note that this says the objective value of a triangle is at most $\frac{3}{2} - \frac{1}{2}(y_{uv} + y_{vw} + y_{uw}) \leq \frac{3}{2} + \frac{1}{2} \cdot \frac{3}{2} = \frac{9}{4}$.

Now notice that all of our separating hyperplanes so far (which are linear in $y_{uv}$) have originated from inequalities like $(\sum_{v \in V} c_v x_v)^2 \geq 0$ for some $c \in \mathbb{R}^n$. Can we capture *all* such inequalities and automate this? In the next class we will show the answer is yes.

Before we go, notice that $(\sum_{v \in V} c_v x_v)^2 \geq 0$ is equivalent to $0 \leq \sum_{u,v \in V} c_u c_v x_u x_v = \sum_{u,v \in V} c_u c_v y_{uv}$ (where pairs $u, v$ appear twice). But letting $Y = (y_{uv})_{u,v \in V}$, this is simply saying that $c^T Y c \geq 0$. This is exactly asking that $Y$ is a positive semi-definite matrix.

In the next class, we will show Goemans and Williamson [GW95] used a solution $y$ which obeys all such constraints to design an approximation algorithm for Max Cut with ratio about 0.878.

## References


[GW95] Michel X. Goemans and David P. Williamson. "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming". In: *J. ACM* 42.6 (Nov. 1995), pp. 1115–1145 (cit. on p. 4).