[CS1961: Lecture 8] Probabilistic Method, Approximation Algorithm

Instructor: Chihao Zhang;
Scribed by Shuze Chen, Ziqi Huang, Yikai Li, Yuchen He

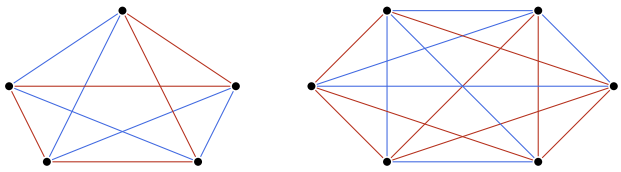
1 Probabilistic Method

The *probabilistic method* is a useful tool in combinatorics. To prove the existence of a certain object, we can construct a probability space and show that the probability of choosing the object is greater than 0 in a randomized trial. Another way to use the probabilistic method is by calculating the expected value of a random variable. We can argue that the probability of the random variable taking a value no larger or no less than the expectation is not 0.

1.1 Ramsey Number

Recall that the *Ramsey number* $R(k,k)$ is the minimum number of vertices of a complete graph such that there exists either a blue clique or a red clique of size k if we color each edge into red or blue in the graph. It is a typical problem in extremal combinatorics. More generally, we define $R(k,\ell)$ as the minimum n such that no matter how we color the edges of K_n, [1] it must contain a red K_k or blue K_ℓ as a subgraph.

[1] K_n is the complete graph with n vertices.



$R(3,3) > 5$ $\qquad$ $R(3,3) = 6$

It has been known that $R(5,5) \in [43,48]$ and $R(6,6) \in [102,165]$. However, it is hard to calculate the exact value of $R(k,\ell)$ even for small k and ℓ. We can get a lower bound $f(k)$ for $R(k,k)$ by constructing a coloring of $K_{f(k)}$ which guarantees no monochromatic K_k in $K_{f(k)}$.

Theorem 1 (Erdős) *Let $G = (V,E)$ be a complete graph with n vertices. If $\binom{n}{k}2^{1-\binom{k}{2}} < 1$, there exists a coloring way for G such that G does not contain a monochromatic subgraph K_k.*

Proof. We dye each edge of G into red or blue uniformly at random. Without loss of generality, let $V = [n]$. For $S \subseteq \binom{[n]}{k}$, let $X_S \triangleq \mathbf{1}[G[S] \text{ is monochromatic}]$. [2] Then

[2] Here $G[S]$ represent the subgraph of G induced by S.

$$\begin{aligned}\mathbf{Pr}\left[K_n \text{ contains a monochromatic } K_k\right] &= \mathbf{Pr}\left[\exists S \subseteq \binom{[n]}{k}, X_S = 1\right] \\ &\leqslant \sum_{S \in \binom{[n]}{k}} \mathbf{Pr}\left[X_S = 1\right] \\ &= \binom{n}{k} 2^{1-\binom{k}{2}} < 1,\end{aligned}$$

where the inequality follows from the union bound. □

The union bound says that given a group of events $A_1, A_2, \ldots, A_n$, $\mathbf{Pr}\left[\exists i\colon A_i \text{ happens}\right] \leqslant \sum_{i=1}^{n} \mathbf{Pr}\left[A_i \text{ happens}\right]$. The equality holds iff the events are mutually exclusive.

Note that we have $n = O\left(k \cdot 2^{\frac{k}{2}}\right)$ when $\binom{n}{k} 2^{1-\binom{k}{2}} < 1$. If $n = c \cdot k \cdot 2^{\frac{k}{2}}$ for small c, we can get a coloring avoiding monochromatic K_k in K_n with high probability by such a random dyeing way.

1.2 Tournament

Let $G = (V, E)$ be a directed graph with $V = [n]$. We call G a *tournament* if there is a directed edge between each pair of vertices in G. Each edge $(x, y) \in E$ depicts the contest result between players x and y. The property P_k states that given any k players, we can always find a stronger player who wins all of those k players in the tournament. In other words, P_k is the event that $\forall S \in \binom{[n]}{k},\ \exists x \in V \setminus S$ s.t. $\forall y \in S,\ (x, y) \in E$.

Theorem 2 (Erdős) *If $\binom{n}{k}(1 - 2^{-k})^{n-k} < 1$, there exists a tournament with n vertices satisfying P_k.*

Proof. Let G be a random graph where the edge between x and y is chosen from $\{(x, y), (y, x)\}$ u.a.r. for any $x, y \in V$. For $S \in \binom{[n]}{k}$, let $X_S \triangleq \mathbf{1}[\forall x \in V \setminus S, \exists y \in S, (x, y) \notin E]$. Then

$$\begin{aligned}\mathbf{Pr}\left[G \text{ does not satisfy } P_k\right] &= \mathbf{Pr}\left[\exists S \in \binom{[n]}{k}, X_S = 1\right] \\ &\leqslant \sum_{S \in \binom{[n]}{k}} \mathbf{Pr}\left[X_S = 1\right] \\ &= \binom{n}{k}\left(1 - 2^{-k}\right)^{n-k} < 1.\end{aligned}$$

□

This condition is satisfied for some $n = O\left(k^2 2^k\right)$.

1.3 Dominating Set

In this example, we demonstrate the technique of *alteration*. That is, we don't directly sample a desired object (which may appear with too small probability). Instead, we sample an object close to what we want, and do some small modification.

Let $G = (V, E)$. The vertices in V is said to be dominated by a set $S \subseteq V$ if $\forall v \in V,\ v \in S$ or $\exists u \in S, (u, v) \in E$. S is called a *dominating set.*

Theorem 3 *If the minimum degree of* G *is* δ*, then there exists a dominating set of size* $\frac{n(1+\log(\delta+1))}{\delta+1}$ *in* G*.*

Proof. We randomly pick a set $S \subseteq V$ by choosing v to be included in S with probability p independently for all $v \in V$. Let T be the set of vertices that are not dominated by S. Let $X = |S|$ and $Y = |T|$. Then $\mathbf{E}[X] = pn$ and

We use $deg(v)$ to denote the degree, i.e., the number of neighbors, of v.

$$\begin{aligned}\mathbf{E}[Y] &= \sum_{v \in V} \mathbf{E}[\mathbf{1}[v \in T]] = \sum_{v \in V} \mathbf{Pr}[v \in T] \\ &= \sum_{v \in V} (1-p) \cdot (1-p)^{\deg(v)} \leqslant n(1-p)^{\delta+1},\end{aligned}$$

where the first equality follows from the linearity of expectations.

The linearity of expectation states that if a random variable $X = \sum_{i=1}^{n} X_i$, then $\mathbf{E}[X] = \sum_{i=1}^{n} \mathbf{E}[X_i]$.

It is clear the union of S and T must be a dominating set. The expected size of $S \cup T$ can be bounded by

$$\mathbf{E}[|S \cup T|] = \mathbf{E}[X] + \mathbf{E}[Y] \leqslant np + n(1-p)^{\delta+1}. \tag{1}$$

Choosing $p = \frac{\log(\delta+1)}{\delta+1}$, which is the minimizer of Equation (1), we have $\mathbf{E}[|S \cup T|] \leqslant \frac{n(1+\log(\delta+1))}{\delta+1}$. Therefore, there exists a dominating set of size $\frac{n(1+\log(\delta+1))}{\delta+1}$ in G .

□

1.4 Independent Set

Let $G = (V, E)$ and $n = |V|$.. A set $S \subseteq V$ is called an *independent set* if $\forall u, v \in S, (u, v) \notin E$.

Theorem 4 *If* G *contains* $\frac{nd}{2}$ *edges for some* $d \geqslant 1$*, there exists an independent set of size at least* $\frac{n}{2d}$ *in* G*.*

Proof. We randomly pick a set $S \subseteq V$ by choosing v to be included in S with probability p independently for all $v \in V$. Let $X = |S|$ and Y be the number of edges in $G[S]$. Then $\mathbf{E}[X] = np$ and

$$\mathbf{E}[Y] = \sum_{e=(i,j) \in E} \mathbf{Pr}[i \in S \text{ and } j \in S] = \frac{p^2 nd}{2}.$$

Similarly, we construct an independent set by first choosing a random S and then deleting a vertex in each pair of adjacent vertices in S. The expected size of this independent set is no less than $\mathbf{E}[X - Y]$. Note that

$$\mathbf{E}[X - Y] = np - \frac{p^2 nd}{2}.$$

Setting p as the maximizer $\frac{1}{d}$, we have $\mathbf{E}[X - Y] = \frac{n}{2d}$. Therefore, we can find an independent set of size at least $\frac{n}{2d}$ in G.

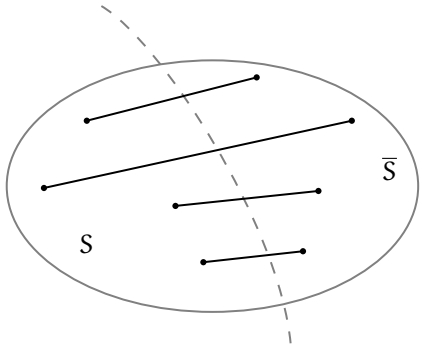
□

2 Approximation Algorithm

Although the probabilistic method is powerful to prove the existence of certain solutions, it cannot provide a concrete solution if no extra concentration condition is provided. However, for some problems, it is possible to design an effective algorithm by *derandomizing* the existence proof provided by the probabilistic method.

2.1 MaxCut

For a graph $G = (V, E)$ where $V = [n]$, the MaxCut problem asks for a partition $(S, \overline{S})$ that maximizes the number of crossed edges $\left|E\left(S, \overline{S}\right)\right|$. The Max Cut is **NP**-hard. However, we can give a lower bound of the maximum $\left|E\left(S, \overline{S}\right)\right|$ using the probabilistic method and find a approximate solution by derandomization.



We randomly pick a set $S \subseteq V$ by letting v to be included in S with probability $\frac{1}{2}$ independently for all $v \in V$, . Then

$$\mathbf{E}\left[\left|E\left(S, \overline{S}\right)\right|\right] = \sum_{e=(i,j)\in E} \mathbf{Pr}\left[e \in E\left(S, \overline{S}\right)\right] = \frac{1}{2}|E|.$$

This indicates that there exists a partition $(S, \overline{S})$ with $\left|E\left(S, \overline{S}\right)\right| \geqslant \frac{|E|}{2}$.

For $i \in V$, let $X_i = \mathbf{1}[i \in S]$. Let $f(X_1, X_2, \ldots, X_n) = \left|E\left(S, \overline{S}\right)\right|$. Then $\mathbf{E}[f(X_1, X_2, \ldots, X_n)] \geqslant \frac{|E|}{2}$. We can construct a concrete S satisfying $f(X_1, X_2, \ldots, X_n) \geqslant \frac{|E|}{2}$ in the following way. The method is called *derandomization by conditional probabilities*. Note that

$$\begin{aligned}\mathbf{E}[f(X_1, X_2, \ldots, X_n)] &= \mathbf{E}[\mathbf{E}[f(X_1, X_2, \ldots, X_n) \mid X_1]] \\ &= \mathbf{Pr}[X_1 = 1]\,\mathbf{E}[f(1, X_2, \ldots, X_n)] + \mathbf{Pr}[X_1 = 0]\,\mathbf{E}[f(0, X_2, \ldots, X_n)].\end{aligned}$$

We set X_1 to be $\arg\max_{x\in\{0,1\}} \mathbf{E}[f(x, X_2, \ldots, X_n)]$ and determine $X_2, \ldots, X_n$ similarly. The partition we construct in this way has a cut value no less than $\frac{OPT}{2}$ since $OPT \leqslant |E|$ and $\mathbf{E}[f(X_1, X_2, \ldots, X_n)] \geqslant \frac{|E|}{2}$.

This is a $\frac{1}{2}$-approximation algorithm in polynomial-time.

It is easy to determine $\mathbf{E}[f(x, X_2, \ldots, X_n)]$ in the way similar to calculating $\mathbf{E}[f(X_1, X_2, \ldots, X_n)]$.

2.2 MaxSAT

For a CNF $\phi = C_1 \wedge C_2 \wedge \cdots C_m$, the MaxSAT problem asks for the maximum number of clauses that can be satisified in ϕ. Let ℓ_j be the number of literals in clause C_j. We assign each variable x_i u.a.r. from $\{0,1\}$[3]. Let $X_j = \mathbf{1}[C_j \text{ is satisfied}]$ for $j \in [m]$ and $X = \sum_{j=1}^{m} X_j$ be the number of satisfied clauses. Then we have

$$\mathbf{E}[X] = \sum_{j=1}^{m} \mathbf{Pr}[X_j = 1] = \sum_{j=1}^{m} \left(1 - 2^{-\ell_j}\right) \geqslant \frac{m}{2} \geqslant \frac{\mathrm{OPT}}{2}.$$

[3] 0 stands for `false` and 1 stands for `true`.

It is easy to get a $\frac{1}{2}$-approximation algorithm using derandomization by conditional expectation as we did in Section 2.1. This approximation ratio can be further improved.

Note that if there exist some $j, j' \in [m]$ such that $C_j = x_i$ and $C_{j'} = \overline{x}_i$, C_j and $C_{j'}$ cannot be both satisfied in the same truth assignment. We can write ϕ as $\bigwedge_{j=1}^{t} (x_j \wedge \overline{x}_j) \bigwedge \phi'$ where ϕ' is a CNF without clause pairs like x_i and $\overline{x}_i$. Then $\mathrm{OPT} \leqslant m - t$. We rename x_i by letting

$$z_i = \begin{cases} \overline{x}_i, & \text{if } \overline{x}_i \text{ appears alone in a clause;} \\ x_i, & \text{o.w..} \end{cases}$$

Assign each $z_i = 1$ w.p. $p > \frac{1}{2}$ and $z_i = 0$ w.p. $1 - p$. Then

$$\mathbf{E}[X] = t + (m - 2t) \cdot \mathbf{Pr}[C_j \text{ is satisfied}].$$

Write C_j as $\left(\bigvee_{k \in P_j} z_{j,k}\right) \vee \left(\bigvee_{k \in N_j} \overline{z}_{j,k}\right)$ where P_j and N_j are the sets of subscripts k that $z_{j,k}$ appears positively and negatively in C_j respectively. When C_j is a singleton clause, we have $C_j = z_i$ for some i by definition. In this case, $\mathbf{Pr}[C_j \text{ is satisfied}] \geqslant p$. When C_j contains at least two literals, the hardest case is that $C_j = \overline{z}_i \vee \overline{z}_k$ for some i and k. In this case, $\mathbf{Pr}[C_j \text{ is satisfied}] \geqslant 1 - p^2$. Therefore, letting $p = 0.618$, we have

$$\mathbf{Pr}[C_j \text{ is satisfied}] \geqslant \min\left\{p, 1 - p^2\right\} \geqslant 0.618.$$

Then

$$\mathbf{E}[X] \geqslant t + (m - 2t) \cdot 0.618 \geqslant 0.618m \geqslant 0.618\mathrm{OPT}.$$

This algorithm improves the approximation ratio from $\frac{1}{2}$ to 0.618.