Lesson 10: The probabilistic method part. I

Theme: Some examples of the probabilistic method.

1 The basic counting argument

Let $K_n$ be a complete (undirected) graph with $n$ vertices without self-loop.

Proposition 10.1 For every $n$ and $k \leq n$, the following holds. If $\binom{n}{k}2^{-\binom{k}{2}+1} < 1$, then it is possible to colour the edges of $K_n$ (with either red or blue) so that it has no monochromatic $K_k$ subgraph.

Proof. Given a complete graph $K_n$, we randomly colour each edge independently with either red or blue (with equal probability). Note that there are exactly $\binom{n}{k}$ different $k$-cliques. Let $m = \binom{n}{k}$. We fix an ordering of all of these $k$-cliques: $C_1, \ldots, C_m$ and let $E_i$ denote the event that clique $C_i$ is monochromatic. Then, $\Pr[E_i] = 2^{-\binom{k}{2}+1}$.

The probability that there is a monochromatic $k$-clique is:

$$\Pr[E_1 \cup \cdots \cup E_m] \leq \sum_{i=1}^{m} \Pr[E_i] \leq m \cdot 2^{-\binom{k}{2}+1} < 1$$

Hence, the probability that none of the cliques $C_1, \ldots, C_m$ are monochromatic is not zero, i.e., there is a colouring of the edges of $K_n$ so that there is no monochromatic $k$-clique. ■

The proof above can be converted into the following Las Vegas type of algorithm.

**Algorithm 1**

Input: A complete graph $K_n$ and an integer $k$ where $\binom{n}{k}2^{-\binom{k}{2}+1} < 1$.

Task: Output a colouring of the edges of $K_n$ in which there is monochromatic $k$-clique.

1. Let $\xi$ be a random colouring of the edges in $K_n$.
2. while there is a monochromatic $k$-clique with colouring $\xi$
3. Choose another random colouring $\xi$
4. Output $\xi$

In principle, Algorithm 1 may not terminate, but the expected number of steps is finite. Let $p = \Pr[E_1 \cup \cdots \cup E_m]$ and let $N$ be the random variable for the number of iterations (of the while loop). Then, the expectation of $N$ is $1/(1-p)$. Note that if $k$ is fixed, $1/(1-p) = \text{poly}(n)$.

2 The expectation argument

In the following example we will use the fact that if $X$ is a random variable, and $\mu$ is its expectation, then $\Pr[X \geq \mu] > 0$ and $\Pr[X \leq \mu] > 0$.

Let $G = (V, E)$ be an undirected graph. A cut of $G$ is a pair $C = (A, B)$ where $A \cup B$ is partition of $V$. Its value is the number of edges of $E$ that cross from $A$ to $B$.

Proposition 10.2 Let $G$ be an undirected graph with $m$ edges. Then, it has a cut with value at least $m/2$.
Proof. Let \( G = (V, E) \) be an undirected graph with \( m \) edges. We construct a cut \( C = A \cup B \) by randomly assigning each vertex \( u \in V \) to either \( A \) or \( B \) (with equal probability).

Let \( e_1, \ldots, e_m \) be the edges in \( G \). For each \( i = 1, \ldots, m \), let \( X_i \) denote the random variable:

\[
X_i \overset{\text{def}}{=} \begin{cases} 
1, & \text{if the two endpoints of } e_i \text{ are in different sets} \\
0, & \text{otherwise}
\end{cases}
\]

Let \( X = \sum_{i=1}^m X_i \), i.e., \( X \) is the random variable for the value of the cut \( C = (A, B) \). Note that \( \Pr[X_i = 1] = 1/2 \). Hence, \( \Exp[X] = m/2 \). Therefore, \( G \) has a cut with value at least \( m/2 \).

Similar to Algorithm 1 above, we can design a Las Vegas algorithm for finding a cut with value \( m/2 \), where \( m \) is the number of edges in the input graph. To bound its expected run time, let \( p = \Pr[C \text{ has value } m/2] \). Now, since \( \Exp[X] = \Exp[\text{value of } C = m/2] \), we can calculate that \( p \geq 1/\left(\frac{m}{2} + 1\right) \). Thus, the expected run time of our Las Vegas algorithm is \( \leq 1/p = \frac{m}{2} + 1 \).

Below we will show how it can be derandomized.

We need a few notations. Let \( G = (V, E) \) be an undirected graph and \( P, Q \) be two disjoint subsets of \( V \). Similar to above, to get a cut \( C = (A, B) \), we assign each vertex \( u \in V \) to either \( A \) or \( B \) as follows.

- Every vertex \( u \in P \) is assigned to \( A \).
- Every vertex \( u \in Q \) is assigned to \( B \).
- Every vertex \( u \notin P \cup Q \) is randomly assigned to either \( A \) or \( B \) (with equal probability).

Let \( N(P, Q) \) denote the random variable for the value of the cut \( C = (A, B) \) where \( P \subseteq A \) and \( Q \subseteq B \). Note that \( \Exp[N(P, Q)] \) is exactly the value of \( (P, Q) \) plus half the number of edges in \( E \setminus (P \cup Q) \times (P \cup Q) \), i.e., the number of edges whose both endpoints are not in \( P \cup Q \). Consider the following deterministic algorithm.

Algorithm 2

Input: A graph \( G = (V, E) \).

Task: Output a cut \( C = (A, B) \) with value at least \( m/2 \), where \( m \) is the number of edges.

1. Let \( v_1, \ldots, v_n \) be the vertices in \( G \).
2. \( P := \emptyset \) and \( Q := \emptyset \).
3. for \( i = 1, \ldots, n \) do
4. if \( \Exp[N(P \cup \{x_i\}, Q)] > \Exp[N(P, Q \cup \{x_i\})] \) then
5. \( P := P \cup \{x_i\} \) and \( Q := Q \cup \{x_i\} \).
6. else
7. \( P := P \) and \( Q := Q \setminus \{x_i\} \).
8. Output the cut \( C = (P, Q) \).

That Algorithm 2 output a cut \( C = (P, Q) \) with value at least \( m/2 \) follows from the following observations.

- \( \Exp[N(\emptyset, \emptyset)] \geq m/2 \) (by Proposition\[10.2\]).
- Let \( (P_0, Q_0), \ldots, (P_n, Q_n) \) denote the sets \( (P, Q) \) after the \( i \)th iteration. Then, for every \( i = 0, \ldots, n - 1 \):
  \[
  \Exp[N(P_i, Q_i)] \leq \Exp[N(P_{i+1}, Q_{i+1})]
  \]
- \( \Exp[N(P_n, Q_n)] \) is the value of the cut \( C = (P, Q) \).

Checking whether \( \Exp[N(P \cup \{x_i\}, Q)] > \Exp[N(P, Q \cup \{x_i\})] \) can be done by comparing the number of neighbours of \( x_i \) that are in \( P \) and \( Q \).
3 Sample and modify

Proposition 10.3 Let G be a graph with n vertices and m edges where \( m = \frac{dn}{2} \), for some d. Then, G has an independent set with at least \( \frac{n}{2d} \) vertices.

Proof. Let G be a graph as stated. Consider the following algorithm.

- Delete every vertex (together with its incident edges) independently with probability \( 1 - \frac{1}{d} \).
- For each remaining edge, remove it and one of its incident vertices.

Obviously, the remaining set of vertices is independent set. Let \( X \) denote the number of vertices that survive the first step and \( Y \) denote the number of edges that survive the first step. Note that each vertex survives with probability \( \frac{1}{d} \) and an edge survives with probability \( \frac{1}{d^2} \). Thus,

\[
\text{Exp}[X] = \frac{n}{d} \quad \text{and} \quad \text{Exp}[Y] = \frac{dn}{2} \cdot \frac{1}{d^2} = \frac{n}{2d}
\]

The number of vertices removed in the second step is at most \( Y \). So the number of remaining vertices after the second step is at least \( X - Y \). Since \( \text{Exp}[X - Y] = \frac{n}{2d} \), the expected number of vertices output the algorithm is at least \( \frac{n}{2d} \). Hence, there is an independent set with at least \( \frac{n}{2d} \) vertices.

Proposition 10.4 For every integer \( k \geq 3 \), there is an undirected graph with n vertices, at least \( \frac{1}{4}n^{1+(1/k)} \) and girth at least \( k \).

Proof. Let \( G_{n,p} \) be the random (undirected) graph with n vertices where between every pair of vertices the probability that there is an edge between them is \( p \). Consider the following algorithm.

- Sample \( G \in G_{n,p} \) with \( p = \frac{n^{(1/k) - 1}}{k - 1} \).
- For every cycle of length \( \leq k - 1 \), delete one of its edges.

Let \( X \) be the number of the edges in \( G \) after the first step and let \( Y \) be the number of the cycles with length \( \leq k - 1 \). There are at most \( \binom{n}{i} \frac{(i-1)!}{2} \) cycles of length \( i \). We have:

\[
\text{Exp}[X] = p \cdot \binom{n}{2} = \frac{1}{2} \left( 1 - \frac{1}{n} \right) n^{1+(1/k)}
\]

\[
\text{Exp}[Y] = \sum_{i=3}^{k-1} \binom{n}{i} \frac{(i-1)!}{2} \leq \sum_{i=3}^{k-1} n^i p^i = \sum_{i=3}^{k-1} n^{i/k} < kn^{(k-1)/k}
\]

Thus, \( \text{Exp}[X - Y] \geq \frac{1}{4}n^{1+(1/k)} \).

Note that the number of edges after the second step is at least \( X - Y \). Thus, there is a graph with \( n \) vertices, at least \( \frac{1}{4}n^{1+(1/k)} \) edges and girth at least \( k \). \( \blacksquare \)

*The girth of a graph is the length of its shortest cycle.*
Appendix

A Pair-wise independent collection of hash functions

Definition 10.5 For $n, k \geq 1$, let $H_{n,k}$ be a collection of functions from $\{0, 1\}^n$ to $\{0, 1\}^k$. We say that $H_{n,k}$ is pair-wise independent, if for every $x, x' \in \{0, 1\}^n$ where $x \neq x'$ and for every $y, y' \in \{0, 1\}^k$, the following holds.

$$\Pr_{h \in H_{n,k}}[ h(x) = y \land h(x') = y'] = 2^{-2k}$$

In the following we show that $H_{n,k}$ exists. First, we show that $H_{n,n}$ exists. For every $n \geq 1$, for every $a, b \in \mathbb{GF}(2^n)$, define a function $h_{a,b}$ from $\{0, 1\}^n$ to $\{0, 1\}^n$ as follows.

$$h_{a,b}(x) \overset{\text{def}}{=} xa + b$$

Theorem 10.6 The collection $H_{n,n} \overset{\text{def}}{=} \{h_{a,b} : a, b \in \mathbb{GF}(2^n)\}$ is pair-wise independent.

We have another candidate for pair-wise independent collection. For every $n \geq 1$, for every $A \in \{0, 1\}^{n \times n}$ and $b \in \{0, 1\}^{n \times 1}$, define a function $h_{A,b}$ from $\{0, 1\}^n$ to $\{0, 1\}^n$ as follows.

$$h_{A,b}(x) \overset{\text{def}}{=} Ax + b$$

Theorem 10.7 The collection $H_{n,n} \overset{\text{def}}{=} \{h_{A,b} : A \in \{0, 1\}^{n \times n} \text{ and } b \in \{0, 1\}^{n \times 1}\}$ is pair-wise independent.

Remark 10.8 Note that the existence of $H_{n,n}$ implies the existence of $H_{n,k}$. If $n < k$, then we can use $H_{k,k}$ and extend $n$ bit inputs to $k$ by padding with zeros. If $n > k$, then we can use $H_{n,n}$ and reduce $n$ bit outputs to $k$ by truncating the last $(n - k)$ bits.

Lemma 10.9 (Valiant and Vazirani, 1986) Let $H_{n,k}$ be a pair-wise independent hash function collection. Let $S \subseteq \{0, 1\}^n$ such that $2^{k-2} \leq |S| \leq 2^{k-1}$. Then, the following holds.

$$\Pr_{h \in H_{n,k}}[\text{there is a unique } x \in S \text{ such that } h(x) = 0^k] \geq 1/8$$

---

$^1$GF($2^n$) denotes a finite field with $2^n$ elements, where each element can be encoded as a 0-1 string of length $n$.

$^2$\{0, 1\}$^{n \times n}$ denotes the set of 0-1 matrices with $n$ rows and $n$ columns and \{0, 1\}$^{n \times 1}$ denotes the set of 0-1 column vectors of $n$ rows. Here the addition $+$ and multiplication $\cdot$ are defined over $\mathbb{Z}_2$. 