Combinatorics of Computation Spring 2025

The Probabilistic Method

Prof. Jara Uitto *Antonio Cruciani*

Overview

The probabilistic method is a tool for proving the existence of objects. The underlying principle is simple: If the probability of selecting an object with the required properties from a well-defined sample space is positive, then the sample space must contain such an object, and therefore such an object exists. Despite of this idea being simple, its application often involves sophisticated combinatorial arguments. Furthermore, from an algorithmic point of view, we are interested in explicit constructions of objects and not just the proof of their existence. In many cases the proofs of existence can be in fact converted into efficient randomized algorithms and sometimes into efficient deterministic algorithms: this process is called derandomization. We will see examples of both randomized and deterministic construction algorithms arising from the probabilistic method.

1 The Basic Counting Argument

1.1 Edge Coloring

As the first example, consider the problem of coloring the edges of a complete graph with two colors such that there are no large cliques in which all the edges have the same color. We will denote by K_n a complete graph on n vertices and will be interested in $K_k \subseteq K_n$, i.e., complete graphs on $k \leq n$ vertices contained in K_n.

Theorem 1. *If $\binom{n}{k} < 2^{\binom{k}{2}-1}$, then it is possible to color the edges of K_n with two colors such that there exists no monochromatic $K_k \subseteq K_n$.*

Proof. First, note that there are $2^{\binom{k}{2}}$ possible colorings of a clique of size k, because it contains exactly $\binom{k}{2}$ edges. Furthermore, there are $\binom{n}{k}$ different k-vertex cliques in K_n, which we number from 1 to $\binom{n}{k}$.

Now, let us denote by A_i the event that the i-th clique of k vertices is monochromatic. Then, $\Pr(A_i) = \frac{2}{2^{\binom{k}{2}}} = 2^{1-\binom{k}{2}}$, because out of all possible colorings of the edges of K_k, only two are monochromatic. By using the union bound, we get

$$\Pr\left(A_1 \vee \cdots \vee A_{\binom{n}{k}}\right) \leq \binom{n}{k} \cdot 2^{1-\binom{k}{2}} < 1,$$

from which we conclude that

$$\Pr\left(\neg A_1 \wedge \cdots \wedge \neg A_{\binom{n}{k}}\right) = 1 - \Pr\left(A_1 \vee \cdots \vee A_{\binom{n}{k}}\right) > 0.$$

□

2 The Expectation Argument

Another approach, sometimes easier, for proving that an object with certain properties exists is to use an averaging argument. More specifically, in a discrete probability space a random variable must with positive probability assume at least one value that is no greater than its expectation and at least one value that is not smaller than its expectation

2.1 Finding a Large Cut

Let us consider the problem of finding a large cut in an undirected graph, where a cut is a partition of the vertices into two disjoint sets and its size is equal to the number of edges whose endpoints are on different sides of the partition. The problem of finding a maximum cut is NP-hard. Using the probabilistic method, it is easy to show that the value of the maximum cut must be at least $\frac{1}{2}$ the number of edges.

Theorem 2. *In an undirected graph G with m edges, there is always a cut of size at least $\frac{m}{2}$.*

Proof. We construct a random cut by assigning the vertices randomly to the two sides of the cut. For every edge e, we define a binary random variable X_e and set

$$X_e = \begin{cases} 1 & \text{If } e \text{ is in the cut} \\ 0 & \text{Otherwise} \end{cases}$$

Then the probability $\Pr(e \in C(A,B)) = \Pr(X_e = 1) = 2/4 = 1/2$. That is because, the endpoints of $e = (u, v)$ can be both in A or in B, and $u \in A, v \in B$ or $v \in A, u \in B$.

Then, $\mathbf{E}[X_e] = \frac{1}{2}$. Moreover, let us define $X = \sum_{e \in E} X_e$ and notice that these X_e are not independent. However, we are interested in X's expected value and the linearity of expectation does not "care" about dependencies. Thus

$$\mathbf{E}[X] = \sum_{e \in E} \mathbf{E}[X_e] = \frac{m}{2}.$$

We conclude that there must exist at least one cut (A, B) of size not smaller than $\frac{m}{2}$. □

Let us now see how we can transform the proof argument into an efficient algorithm. We will first show how to obtain a Las Vegas algorithm.

It is easy to randomly choose a partition as described in the proof. For a partition (A, B) of the vertices, let $C(A, B)$ be the capacity of the cut (meaning, the number of edges with one endpoint on both sides of the cut). Then, let $p = \Pr\left(C(A,B) \geq \frac{m}{2}\right)$. Then,

$$\frac{m}{2} = \mathbf{E}[C(A,B)] = \sum_{i<m/2} i \cdot \Pr\left(C(A,B) = i\right) + \sum_{i \geq m/2} i \cdot \Pr\left(C(A,B) = i\right) \leq (1-p)\left(\frac{m}{2} - 1\right) + p \cdot m,$$

as the capacity of any cut is upper bounded by m. But this implies that $p \geq \frac{1}{m/2+1}$.

The algorithm: Generate a random cut and determine its capacity. Repeat until a cut of capacity at least $m/2$ has been found. The expected number of repetitions is $\frac{1}{p} \leq \frac{m}{2} + 1$.

3 Derandomization Using Conditional Expectations

3.1 Finding a Large Cut

We will now try to derive a deterministic algorithm for determining a cut of size at least $m/2$. First, we number the vertices v_1 to v_n and denote by S_i the set where v_i is placed, meaning A or B. Imagine we have deterministically placed vertices $v_1, \ldots, v_k$. Then, let $\mathbf{E}[C(A,B) \mid S_1, \ldots, S_k]$ denote the conditional expectation of the cut capacity given that we placed vertex v_i on side $S_i \in \{A, B\}$ for $1 \leq i \leq k$. We will show inductively how to place the next vertex such that

$$\mathbf{E}[C(A,B) \mid S_1, \ldots, S_k] \leq \mathbf{E}[C(A,B) \mid S_1, \ldots, S_k, S_{k+1}],$$

from which it follows that

$$\mathbf{E}[C(A,B)] \leq \mathbf{E}[C(A,B) \mid S_1, \ldots, S_n].$$

Notice that the last inequality states exactly what we wanted, since it represents the value of the cut determined by our placement algorithm.

Now, for $k = 1$ it holds that $\mathbf{E}[C(A,B)] = \mathbf{E}[C(A,B) \mid S_1]$ because by symmetry it does not matter where the first vertex is placed. For the induction step, consider placing v_{k+1} randomly. Then,

$$\mathbf{E}[C(A,B) \mid S_1, \ldots, S_k] = \frac{1}{2}\mathbf{E}[C(A,B) \mid S_1, \ldots, S_k, A] + \frac{1}{2}\mathbf{E}[C(A,B) \mid S_1, \ldots, S_k, B].$$

What we can do is compute both quantities on the right and place v_{k+1} in the set that yields the larger expectation. If we do so, our placement will satisfy

$$\mathbf{E}[C(A,B) \mid S_1, \ldots, S_k] \leq \mathbf{E}[C(A,B) \mid S_1, \ldots, S_k, S_{k+1}].$$

Computing the expectations can be done in linear time: for edges having both endpoints among $v_1, \ldots, v_{k+1}$ we know their contribution and all other edges contribute with probability $\frac{1}{2}$. Since the contribution of these other edges is the same for both placements, we simply need to place v_{k+1} such that we cut at least half of the edges connecting it to vertices $v_1, \ldots, v_k$.

The algorithm then has the following simple form: Place the first vertex arbitrarily. Place each successive vertex on the side with fewer neighbors, breaking ties arbitrarily.

This simple greedy algorithm guarantees a cut with at least $m/2$ edges. To verify this, denote by d'_{k+1} the number of edges connecting v_{k+1} to $\{v_1, \ldots, v_k\}$. We place v_1 arbitrarily and v_{k+1}, $k \geq 1$ such that at least $d'_k/2$ edges are cut. Then,

$$\sum_{k=1}^{n} \frac{d'_k}{2} = \frac{1}{2}\sum_{k=1}^{n} d'_k = \frac{m}{2}.$$

References

[1] Mitzenmacher, Michael and Upfal, Eli, *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis, Cambridge university press, 2017*,