

Probabilistic Method

Sam Heil

29 March 2019

Main idea: For some existence problems in graph theory, explicitly constructing an object is difficult, so it is best to select things randomly and show that it must exist in some case.

There are essentially two (pretty trivial) ideas that make most of these proofs work:

1. Linearity of Expectation: $\mathbb{E}[\sum_{n=1}^N X_n] = \sum_{n=1}^N \mathbb{E}[X_n]$.
2. “Existence from Expectation”: If $\mathbb{E}_x[f(x)] = C$, then there are x_1, x_2 with $f(x_1) \leq C, f(x_2) \geq C$.

These methods are very problem-specific, so it's best to see them by starting with some problems.

Problem 1 (Vertex Cover). *Given a graph G on n vertices with vertex set V , what is the smallest k such that there is a vertex set $U \subset V$ with $|U| = k$ such that every vertex $v \in V \setminus U$ is adjacent to some vertex $u \in U$?*

This problem is NP-complete in general. However, if we include a lower bound on the degrees of the vertices, we can get an upper bound on k using probability!

Theorem 1. *Let G be a graph on n vertices, such that each vertex has degree at least d . Let V be the set of vertices of G . Then, there is some subset $U \subset V$ such that every vertex $v \in V \setminus U$ is adjacent to some $u \in U$, for which*

$$|U| \leq \frac{n(1 + \log(d + 1))}{d + 1}.$$

Proof. It's really hard to construct U explicitly; in fact, we just saw that a similar problem is NP-complete.

Because of this, we'll just choose U randomly! Let p be some probability (to be named later), and for each $v \in V$, say that $v \in U_0$ with probability p . (Note: U_0 is approximately going to be U , but we'll have to add a few elements separately.)

Now, let v_i be any vertex, chosen a priori. The probability that we get a counterexample from vertex v_i for a specific U_0 generated according to the random process above is then equal to $(1 - p)^{1 + \deg(v_i)} \leq (1 - p)^{d+1}$, since we need that $v_i \notin U$ and also that every neighbor of v_i is not in U_0 .

Now, let C be the set of counterexamples with the set U_0 . We have that

□