Lecture 4: A second helping of second moments CSE 525, Fall 2016
Instructor: James R. Lee

1 Percolation on a tree

In the last lecture, we used the fact that for a real-valued random variable X, we have

$$\mathbb{P}(X = 0) \leqslant \frac{\mathrm{Var}(X)}{(\mathbb{E}\, X)^2}\,.$$

Let us see that second moments can also control the probability that a non-negative random variable is non-zero. (A generalization of this inequality often goes by the name "Payley-Zygmund inequality.")

Lemma 1.1. *If X is a non-negative random variable, then*

$$\mathbb{P}(X > 0) \geqslant \frac{(\mathbb{E}\, X)^2}{\mathbb{E}[X^2]}\,.$$

Proof. The proof relies on the Cauchy-Schwarz inequality: For any two real-valued random variables Y and Z,

$$\mathbb{E}\,|YZ| \leqslant \sqrt{\mathbb{E}[Y^2]}\sqrt{\mathbb{E}[Z^2]}\,.$$

We will use $Y = X$ and $Z = \mathbf{1}_{\{X>0\}}$, i.e. Z is the indicator of the event that $X > 0$:

$$\mathbb{E}\, X = \mathbb{E}\left[X\mathbf{1}_{\{X>0\}}\right] \leqslant \sqrt{\mathbb{E}[X^2]}\sqrt{\mathbb{E}\,\mathbf{1}_{\{X>0\}}} = \sqrt{\mathbb{E}[X^2]}\,\sqrt{\mathbb{P}(X>0)}\,.$$

Rearranging and squaring both sides completes the proof. □

Let's now consider the following model of *percolation* on the complete (rooted) binary tree of depth n. Fix a parameter $p \in [0,1]$ and also an orientation of the tree (so that we can refer to "left" and "right" children). Suppose that every left edge is independently deleted with probability $1-p$ and every right edge is independently deleted with probability p. Let $X = \sum_\ell Z_\ell$ denote the number of leaves which can reach the root of the tree. Here, the sum is over all 2^n leaves ℓ and Z_ℓ is the indicator random variable that is 1 precisely when the path from the root to the leaf ℓ remains intact. Here we are interested in the probability that there exists at least one reachable leaf, i.e. $\mathbb{P}[X > 0]$.

The first moment. It is relatively easy to compute the expected value of X:

$$\mathbb{E}\, X = \sum_{i=0}^{n} \binom{n}{i} p^i (1-p)^{n-i} = (p + 1 - p)^n = 1\,.$$

The summation variable i indexes the number of left turns that a root-leaf path makes. If a path makes i left turns, then the probability it remains intact is $p^i(1-p)^{n-i}$. Furthermore, we can specify a root-leaf path by a sequence of "left" and "right" turns; this implies that the number of leaves whose root-leaf path contains exactly i left turns is $\binom{n}{i}$.

This calculation doesn't tell us that there is a root-leaf with decent probability. Certainly if $p = 0$ or $p = 1$, then there is a leaf with probability one (the left-most and right-most paths, respectively). But what about intermediate values of p? What if $p = 1/2$?

The second moment. Let's now compute the second moment, but we'll be a bit more clever. Let X_n denote the number of reachable leaves in a tree of depth n. If X_L and X_R denote the number of reachable leaves under the left and right subtrees, then $X_n^2 = (X_L + X_R)^2 = X_L^2 + X_R^2 + 2X_L X_R$.

Observe that $\mathbb{E}[X_L^2] = p\,\mathbb{E}[X_{n-1}^2]$, $\mathbb{E}[X_R^2] = (1-p)\,\mathbb{E}[X_{n-1}^2]$, $\mathbb{E}[X_L] = p\,\mathbb{E}[X_{n-1}] = p$ and $\mathbb{E}[X_R] = (1-p)\,\mathbb{E}[X_{n-1}]$. Since X_L and X_R are independent, we have

$$\mathbb{E}[X_n^2] = \mathbb{E}[X_{n-1}^2] + 2p(1-p)\,.$$

Since $X_0 = 1$, we get $\mathbb{E}[X_n^2] = 1 + 2np(1-p)$.

Now applying Lemma 1.1 gives

$$\mathbb{P}(X > 0) \geqslant \frac{1}{1 + 2np(1-p)}\,.$$

Thus if $p = \Theta(1/n)$, there exists a reachable leaf with constant probability. If $p = 1/2$, we get a leaf with probability at least $2/(n+2)$.

Notice that this still leaves a number of interesting questions to be answered. We know that $\mathbb{P}(X > 0) \geqslant 2/(n+2)$, but $\mathbb{E}[X_n^2] = \Theta(n)$. Is it the case that we see $\Theta(\sqrt{n})$ reachable leaves with probability $\Theta(1/n)$?

[See video of the discrete torus being covered by random walk—the asymptotics of the cover time are determined by a related percolation process on a complete tree.]

2 Using unbiased estimators to count

Consider a formula in disjunctive normal form (a DNF formula), e.g.

$$\varphi = (X_1 \wedge X_2 \wedge \bar{X}_3) \vee (\bar{X}_2 \wedge X_5) \vee \cdots$$

We can easily determine whether such a formula is satisfiable (we just check whether each term separately is satisfiable). On the other hand, *counting* the number of satisfying assignments to such a formula is #P-hard. (That means the problem is at least as hard as an NP-complete problem; this class of decision problems is thought to be extremely difficult to solve.)

The reason for this hardness is easy to see. Suppose that φ is a CNF formula. Then de Morgan's laws can be used to write $\neg\varphi$ as a DNF formula in polynomial time. But if φ is a formula on n variables, then

$$\#\text{ satisfying assignments to } \varphi = 2^n - \big(\#\text{ satisfying assignments to } \neg\varphi\big)\,.$$

Nevertheless, we will show that one can obtain an efficient algorithm to approximate the number of satisfying assignments.

Theorem 2.1 (Karp and Luby, 1983)**.** *Given a DNF formula φ with n variables and m terms, and a number $\varepsilon > 0$, there is an algorithm running in time $O(mn/\varepsilon^2)$ that outputs a value Z such that*

$$\mathbb{P}\big[(1-\varepsilon)Z \leqslant N(\varphi) \leqslant (1+\varepsilon)Z\big] \geqslant \frac{3}{4}\,,$$

where $N(\varphi)$ is the number of satisfying assignments to φ.

As we saw in the last lecture, one can use the median trick to amplify the probability of correctness to be very close to 1. One should note that a naïve Monte Carlo algorithm will not work so well: If we choose one of the 2^n assignments uniformly at random and check whether it satisfies φ, then the number we are trying to estimate is $\mu = \frac{N(\varphi)}{2^n}$, but this could be exponentially small. The unbiased estimator theorem would dictate that we use exponentially many samples, making our algorithm quite inefficient.

Proof of Theorem 2.1. The main idea will be to apply the Monte Carlo method to a more suitable space. Let S_i be the set of assignments which satisfy the ith term in φ. Our goal is to compute the size of the union $\bigcup_{i=1}^{m} S_i$.

Let $\mathcal{U} = \{(a, i) : a \in S_i\}$. Say that the pair $(a, i) \in \mathcal{U}$ is *special* if i is the first term that the assignment a satisfies. Since every satisfying assignment has exactly one first term that it satisfies, we have the following.

Claim 2.2. The number of special pairs is precisely $\left|\bigcup_{i=1}^{m} S_i\right|$

Now we apply the Monte Carlo algorithm to estimate $|\mathcal{S}|/|\mathcal{U}|$ where $\mathcal{S}$ is the set of special pairs. Note that $|S_i| = 2^{n-q_i}$ where q_i is the number distinct variables in term i. Thus we can easily compute $|\mathcal{U}| = \sum_{i=1}^{m} |S_i|$. In particular, if we obtain a multiplicative approximation to $\mu = |\mathcal{S}|/|\mathcal{U}|$, then we can obtain a multiplicative approximation to $|\mathcal{S}| = N(\varphi)$.

To generate a random sample from $\mathcal{U}$, we choose i with probability $|S_i|/|\mathcal{U}|$, and then pick a random satisfying assignment for term i. Finally, this is extended to a uniformly random assignment on the remaining $n - q_i$ variables.

Thus we are left to estimate $\mu = |\mathcal{S}|/|\mathcal{U}|$. But it's easy to see that $|\mathcal{U}| \leqslant m|\mathcal{S}|$ since each satisfying assignment can satisfy at most m terms. Therefore $\mu \geqslant \frac{1}{m}$. The unbiased estimator theorem now states that we need at most $\frac{4}{\varepsilon^2 \mu} \leqslant \frac{4m}{\varepsilon^2}$ samples in order to achieve our goal.

A naïve implementation of the algorithm requires $O(nm^2/\varepsilon^2)$ time, but further improvements [Karp-Luby-Madras 1989] show how to implement the algorithm in $O(mn/\varepsilon^2)$ time. □

Remark 2.3. The algorithm can be used to approximate the size of the union of any collection of finite sets as long as we can compute their individual sizes and sample random elements from each of them.

A straightforward generalization of the algorithm allows us to compute the probability of a random assignment in a probabilistic DNF model where each variable is true independently with some probability p_i.