3 Lecture 3, October 6, 2014

3.1 Markov Inequality

Lemma 10 *Let A be a non-negative random variable with well-defined (that is, finite) expectation μ. Then for any $\lambda > 0$, $\Pr(A > \lambda\mu) < 1/\lambda$. (In particular, for $\mu = 0$, $\Pr(A > \mu) = 0$.)*

Proof: Write

$$\begin{aligned} \mu = \quad & E(A|A \leq \lambda\mu)\Pr(A \leq \lambda\mu) + \quad & E(A|A > \lambda\mu)\Pr(A > \lambda\mu) \\ \geq \quad & & E(A|A > \lambda\mu)\Pr(A > \lambda\mu) \end{aligned}$$

In the case $\mu > 0$ the second line is

$$> \lambda\mu \Pr(A > \lambda\mu)$$

and we divide both sides by $\lambda\mu$.

In the case $\mu = 0$ the second line becomes

$$0 \geq E(A|A > 0)\Pr(A > 0) \geq \sum_{n \geq 1} \frac{1}{n} \Pr(\frac{1}{n} \leq A < \frac{1}{n-1})$$

By countable additivity of probabilities, $\Pr(A > 0)$ would imply that at least one term in the summation is positive, a contradiction. □

3.2 Large Girth and Large Chromatic Number

Earlier we saw our first example of the *probabilistic method*. In that case, just picking an element of a set at a random was already enough in order to produce an object that is hard to construct "explicitly".

Nonetheless, of course, the probabilistic method in that form can construct only an object with properties that are shared by a large fraction of objects. Now we will see an example that enables the probabilistic method to construct something that is quite rare—indeed, it is maybe a bit surprising that this kind of object even exists.

We consider graphs here to be undirected and without loops or multiple edges.

The *chromatic number* χ of a graph is the least number of colors with which the vertices can be colored, so that no two neighbors share a color. Clearly, as you add edges to a graph, its chromatic number goes up.

The *girth* γ of a graph is the length of a shortest simple cycle. ("Simple" = no edges repeat.) Clearly, as you add edges to a graph, its girth goes down.

These numbers are both *monotone* in the inclusion partial order on graphs. Chromatic number is monotone increasing, while girth is monotone decreasing. An important theorem we hope to reach later in the course is the FKG Inequality, which implies in this setting that for any $k, g > 0$, if you pick a graph u.a.r., and condition on the event that its chromatic number is above k, that reduces the probability that its girth will be above g.

So in this precise sense, chromatic number and girth are anticorrelated. Indeed, having large girth means that the graph is a tree in large neighborhoods around each vertex. A tree has chromatic number 2. If you just allow yourself 3 colors, you gain huge flexibility in how to color a tree. Surely, with large girth, you might be able to color the local trees so that when they finally meet up in cycles, you can meet the coloring requirement?

No!

Here is a remarkable theorem.

Theorem 11 (Erdös [18]) *For any k, g there is a graph with chromatic number $\chi \geq k$ and girth $\gamma \geq g$.*

Proof: Pick a graph G from $G(n,p)$, where $p = n^{-1+1/g}$. This is likely to be a fairly sparse graph, with average degree $\sim n^{1/g}$.

Let the rv X be the number of cycles in G of length $< g$. $E(X) = \sum_{m=3}^{g-1} p^m n \cdot (n-1) \cdots (n-m+1)/(2m)$. (Pick the cycle sequentially and forget the starting point and orientation.) Then

$$E(X) < \sum_{m=3}^{g-1} p^m n^m/(2m) = \sum_{m=3}^{g-1} n^{m/g}/(2m) \leq \sum_{m=3}^{g-1} n^{m/g}/2.$$

For sufficiently large n, specifically $n > 2^g$, the successive terms in this sum at least double, so $E(X) \leq n^{1-1/g}$. By Markov's inequality, $\Pr(X > 3n^{1-1/g}) < 1/3$.

For the chromatic number we use a simple lower bound. Let I be the size of a largest independent set in G. Since every color class of a coloring must be an independent set,

$$I \cdot \chi \geq n. \tag{7}$$

Now $\Pr(I \geq i) \leq \binom{n}{i}(1-p)^{\binom{i}{2}}$, and recalling (6), the simple inequality for the exponential function, we have $\Pr(I \geq i) \leq \binom{n}{i} e^{-p\binom{i}{2}} = \binom{n}{i} e^{-\binom{i}{2} n^{-1+1/g}}$. Using the wasteful bound $\binom{n}{i} \leq n^i$ we have $\Pr(I \geq i) \leq e^{i \log n - \binom{i}{2} n^{-1+1/g}}$.

Finally we apply this at $i = 3n^{1-1/g} \log n$. $\Pr(I \geq i) \leq e^{(3/2)(\log n - n^{1-1/g} \log^2 n)}$ which for sufficiently large n is $< 1/3$.

Thus, for sufficiently large n, there is probability at least $1/3$ that G has both $I < 3n^{1-1/g} \log n$ and at most $3n^{1-1/g} \leq n/2$ cycles of length strictly less than g.

Removing vertices from G can only reduce I. (Very differently from removing edges!) So, by removing one vertex from each cycle, we obtain a graph with $\geq n/2$ vertices, girth $\geq g$, and $I \leq 3n^{1-1/g} \log n$. Applying (7), we have $\chi \geq n^{1/g}/(6 \log n)$ which for sufficiently large n is $\geq k$. □

3.3 Achieving expectation in MAX-3SAT

Let's start looking at some computational problems. A 3CNF formula on variables $x_1, \ldots, x_n$ is the disjunction of clauses, each of which is a conjunction of at most three literals. (A literal is an x_i or x_i^c, where x_i^c is the negation of x_i.)

You will recall that it is NP-complete to decide whether a 3CNF formula is satisfiable, that is, whether there is an assignment to the x_i's s.t. all clauses are satisfied. Let's take a little different focus: think about the *maximization* problem of satisfying as many clauses as possible. Of course this is NP-hard, since it includes satisfiability as a special case. But, being an optimization problem, we can still ask how well we can do.

Theorem 12 *For any 3CNF formula there is an assignment satisfying $\geq 7/8$ of the clauses. Moreover such an assignment can be found in randomized time $O(m^2)$, where m is the number of clauses (and we suppose that every variable occurs in some clause).*

Proof: The existence assertion is due to linearity of expectation.

The algorithm is often attributed to the English educator Hickson [28]...

> 'Tis a lesson you should heed:
> Try, try, try again.
> If at first you don't succeed,
> Try, try, try again.

Checking an assignment takes time $O(m)$. How many trials do we need to succeed?

Let the rv M be the number of satisfied clauses of a random assignment. $m - M$ is a nonnegative rv, and Markov's inequality tells us that $\Pr(M \leq (7/8 - \varepsilon)m) = \Pr(m - M \geq (1 + 8\varepsilon)m/8) \leq 1/(1 + 8\varepsilon)$.

This says we have a good chance of getting close to the desired number of satisfied clauses; however, we asked to achieve 7/8, not $7/8 - \varepsilon$. We can get this by noting that M is integer-valued, so for $\varepsilon < 1/m$, an assignment satisfying $7/8 - \varepsilon$ of the clauses, satisfies 7/8 of them.

With this choice for ε, then, the probability that a trial succeeds is at least

$$1 - \frac{1}{1 + 8\varepsilon} = \frac{8}{8 + 1/\varepsilon} \in \Omega(1/m)$$

Trials succeed or fail independently so the expected number of trials to success is the expectation of a geometric random variable with parameter $\Omega(1/m)$, which is $O(m)$. □

3.4 Derandomization

How can we improve on this simple-minded method? We do not have a way forward on increasing the fraction of satisfied clauses, because of:

Theorem 13 (Håstad [27]) *For all $\varepsilon > 0$ it is NP-hard to approximate Max-3SAT within factor $7/8 + \varepsilon$.*

But we might hope to reduce the runtime, and also perhaps the dependence on random bits. As it turns out we can accomplish both of these objectives.

Theorem 14 *There is an $O(m^2)$-time deterministic algorithm to find an assignment satisfying 7/8 of the clauses of any 3CNF formula on m clauses.*

Proof: This algorithm illustrates the *method of conditional expectations*. The point is that we can derandomize the randomized algorithm by not picking all the variables at once–instead, we consider the alternative choices to just one of the variables, and choose the branch on which the conditional expected number of satisfying clauses is greater.

This method works in situations in which one can actually quickly calculate (or at least approximate) said conditional expectations.

In the present example this is easy. The probability that a clause of size i is satisfied is $1 - 2^{-i}$. If a formula has m_i clauses of size i, the expected number of satisfied clauses is $\sum m_i(1 - 2^{-i})$. Now, partition the clauses of size i into m_i^1 that contain the literal x_i, m_i^0 that contain the literal x_i^c, and those that contain neither.

The expected number of satisfied clauses conditional on setting $x_i = 1$ is

$$\sum m_i^1 + \sum m_i^0(1 - 2^{-i+1}) + \sum(m_i - m_i^1 - m_i^0)(1 - 2^{-i}). \tag{8}$$

Similarly the expected number of satisfied clauses conditional on setting $x_i = 0$ is

$$\sum m_i^1(1 - 2^{-i+1}) + \sum m_i^0 + \sum(m_i - m_i^1 - m_i^0)(1 - 2^{-i}). \tag{9}$$

We can compute each of these quantities in time $O(m)$ and simply choose the setting to x_1 which gives the larger of them. (Actually, since these quantities average to the current expectation, which we already know, we only have calculate one of them.)

Naively, this process runs in time $O(m^2)$. However, it can actually be performed in time $O(m)$: after assigning each variable we hardwire it into the formula, and while calculating the quantities (8),(9) to choose the assignment to a variable we only need to scan the clauses it belongs to. The runtime is $O(m)$ because there are only $O(m)$ memberships of variables in clauses. □