# Probability and Computing Chapter 6 Notes and Questions

Angjoo Kanazawa

July 27, 2011

# 1 Chapter 6 The Probabilistic Method

## 1.1 Notes

- To prove teh existence of an object with specific properties, construct an approximate probability space of objects $S$, and show that the probability that an object (in $S$ with the specific proprties) is selected is $> 0$. Strictly.

## 1.2 Exercises

**6.1** Consider an instance of SAT with $m$ clauses, where every clause has exactly $k$ literals.

(a) Give a Las Vegas algorithm that finds an assignment that satisfies at least $m(1-2^{-k})$ clauses, anayze its expected running time:

The first part is straight from the book (6.2.2). Note:

$$P(\text{a clause is satisfied}) = 1 - P(\text{all literals are false}) = 1 - 2^{-k}$$

Let $i = 1 \dots m$, $X_i = 1$ if $i$th clause is satisfied, $X_i = 0$ otherwise. Let $X = \sum_i^m X_i$, the total number of satisfied clauses. So we get

$$E(X) = \sum_i^m X_i P(X_i = 1) = m(1-2^{-k}) = \mu$$

Now given this, the LV algorithm goes like this:

Repeat until $X \geq \mu$:

- assign values to all boolean variables independently and uniformly.
- Check the value of $X$.

What is the expected number of runs until LV finishes? We're done when $X \geq \mu$. Note that $X$ has a binomial distribution*. Let $Y$ be the number of runs needed for the LV to terminate, and $Y_i$ be a random variable indicating if the algorithm terminated at $i$th run or not with probability $\hat{p} = P(Y_i = 1)$. So $Y$ has a geometric distribution and $E(Y)$, what we want, is

$$E(Y) = \sum_{I=1}^{\infty} i\hat{p}(1-\hat{p})^{i-1} \tag{1.1}$$

So for each run, what is $\hat{p} = P(Y_i = 1) = P(X \geq \mu)$? Using $p = P(\text{a clause is satisfied}) = 1 - 2^{-k}$:

$$\begin{aligned}
P(X \geq \mu) &= 1 - P(X < \mu) \\
&= 1 - [P(X = 0) + P(X = 1) + \cdots + P(X = \mu - 1)] \\
&= 1 - [(1-p)^m + \binom{m}{1} p(1-p)^{m-1} + \cdots + \binom{m}{\mu - 1} p^{\mu-1}(1-p)^{m-(\mu-1)}] \\
&= 1 - \sum_{i=0}^{\mu-1} \binom{m}{i} p^i (1-p)^{m-i}
\end{aligned}$$

Now recall $\sum_i^m \binom{m}{i} = 2^m$, so $\sum_i^{\mu-1} \binom{m}{i} \leq \frac{2^m}{2} = 2^{m-1}$.
Also, notice that here $p^i(1-p)^{m-i} = (1-2^{-k})^i(1-2^{-k})^{m-i} = (1-2^{-k})^m$.
So we can continue the above inequalitywith:

$$\begin{aligned}
P(X \geq \mu) &= 1 - \sum_{i=0}^{\mu-1} \binom{m}{i} p^i (1-p)^{m-i} \\
&\geq 1 - [2^{m-1}(1-2^{-k})^m] = 1 - 2^{-k}
\end{aligned}$$

As an example with $k = 1$, this $P(X \geq \mu) = \hat{p}$ is just 1/2. Using that 1.1, we get $E[Y] = ....$ The algorithm is.. very efficient.

...
Really..? * is where I'm not sure. Perhaps this is just too much. Can one always say $P(X \geq \mu) \geq 1/2$?? I wasn't sure.

(b) Give a derandomization of the randomized algorithm using the method of conditional expectations: This I also just followed the book. Maybe too closely.

We know setting variables independently and uniformly gives us $E(X) \geq m(1 - 2^{-k})$. Now set the boolean variables $x_1, x_2, \ldots$ up to $r$ deterministically one at a time.

Consider the expected total # of satisfied clauses if the remaining boolean variables are selected independently and uniformly. Write this as $E(X|x_1, x_2, \ldots, x_r)$. We want a away o set the next variable s.t.

$$E(X|x_1, \ldots, x_r) \leq E(X|x_1, \ldots, x_r, x_{r+1}) \tag{1.2}$$

Inductively, the base case is $E(X|x_1) = E(X)$. Now, consider setting $x_{r+1}$ randomly to true or false. Each has probability 1/2. So $E(X|x_1, \ldots, x_r) = \frac{1}{2}E(X|x_1, \ldots, x_{r+1} = 1) + \frac{1}{2}E(X|x_1, \ldots, x_{r+1} = 0)$ From this we can deduce

$$\max(E(X|x_1, \ldots, x_r, x_{r+1} = 1), E(X|x_1, \ldots, x_r, x_{r+1} = 0)) \geq E(X|x_1, \ldots, x_r)$$

So we just have to chose the assignment that increases the conditional expectation the most.. we only have two options $x_{r+1}$ is T or F, so look at clauses that contain the $x_{r+1}$ variable twice and see how the expectation changes based on the assignment and take the better one? Something like that.