1 MAX-SAT Continued

We will begin this lecture by filling in some of the details regarding the algorithm for MAX-SAT we presented in the previous lecture. We shall then show how this algorithm can be derandomized. Let us begin with a short reminder: The algorithm comprises two sub-algorithms. The first sub-algorithm simply guesses the assignment of each variable. The expectation that this algorithm satisfies clause $j$ (of length $k$) is clearly $E_{\text{first}}[c_j] = 1 - (1 - \frac{1}{2})^k$ (let $\alpha_k = 1 - (1 - \frac{1}{2})^k$). The linearity of expectation guarantees this approximation for the entire boolean formula. The second algorithm (we shall refer to as $GW$) is such that $E_{\text{second}}[c_j] \geq \beta_k z_j$ (with $b_k = 1 - (1 - \frac{1}{2})^k$).

We shall show that by randomly picking one of the two algorithms we manage to get a good approximation ratio. The intuition for this is that one gets better as $k$ increases and the other gets better as $k$ decreases. The expectation that clause $j$ is satisfied by this new algorithms is $E[c_j] = \frac{1}{2}(\alpha_k + \beta_k)z_j$. We wish to show that $E[c_j] \geq \frac{3}{4}z_j$. Hence, it suffices to show that $\alpha_k + \beta_k \geq \frac{3}{2}$ for all values of $k$. One can easily verify that this is indeed correct (by assigning $k = 1, 2, 3$).

Now that we have designed a randomized algorithm with a good expectation of success, we shall show how it can be converted into an algorithm that succeeds with high probability. Consider a minimization problem. Let $A$ be an algorithm, and $I$ be an instance of the problem, such that $E[A(I)] < \alpha OPT(I)$. According to the Markov bound:

$$Pr[X \geq tE[X]] \leq \frac{1}{t}$$

Hence:

$$Pr[A(I) > (1 + \delta)\alpha OPT(I)] \leq \frac{1}{1 + \delta}$$

And so we have that by repeating our algorithm $m$ times the probability of failure is $\leq (\frac{1}{1+\delta})^m$. For $m = O(\frac{1}{\delta} \log n)$ we have that the probability of failure is polynomially low (in $n$).

We now turn to derandomizing the MAX-SAT algorithm. We do this using the conditional expectation technique. We know that the randomized algorithm we have has an expectation of success of at least $\frac{4}{3}$ of the optimal solution. Define some arbitrary order on the boolean
variables $x_1, ..., x_n$. We shall define a vertex $(a_1, ..., a_l)$ for every $1 \leq l \leq n$ and such that $a_i \in \{0, 1\}$. This vertex corresponds to the instance of the problem we get if assign $x_i$ the value of $a_i$ for every $1 \leq i \leq l$. Observe that for every such vertex $v$ we can compute the expectation of the number of satisfied clauses (given the algorithm) for the instance represented by $v$, in polynomial time. We shall denote this expectation by $E_v$.

We shall now present the simple deterministic algorithm for MAX-SAT.

- Start with $v = (\emptyset)$.
- While the number of coordinates in $v$ is smaller than $n$ perform the following step: if $v = (a_1, ..., a_l)$, Let $v_0 = ((a_1, ..., a_l, 0)$ and $v_1 = ((a_1, ..., a_l, 0)$. Assign $v = \text{argmax}_{i \in \{0, 1\}} E_{v_i}$.

To see why this algorithm let us start by considering the first step. $v = (\emptyset)$, and so we know that $E_v$ is at least a $\frac{3}{4}$ fraction of the optimal solution (we are guaranteed this by the approximation ration of the algorithm). Assume that the randomized algorithm chooses $v_0$ with probability $p_0$ and $v_1$ with probability $p_1$. Then, $E_v = p_0 E_{v_0} + p_1 E_{v_1} \leq \max_i E_{v_i}$. And so, by choosing the $v_i$ that maximizes the expectation $E_{v_i}$ we are still guaranteed a good approximation. We can now repeat this step over and over again without reducing the value of the guaranteed expectation.

## 2 On Chebyshev and Chernoff Bounds

**Theorem 1** (The Chebyshev bound:) $\Pr[|X - E[X]| > t\sigma] < \frac{1}{t^2}$

**Proof:** Set $Y = (X - E[X])^2$ and apply the Markov bound. ■

**Theorem 2** (The Chernoff bound:) Let $X_i$ ($1 \leq i \leq n$) be $n$ random variables such that $Pr[X_i = 1] = p_i$ and $Pr[X_i = 0] = 1 - p_i$. Let $X = \Sigma_i X_i$ and $\mu = E[X]$. Then:

$$Pr[X < (1 - \delta)\mu] < \left(\frac{e^{-\delta}}{(1 - \delta)^{(1-\delta)}}\right)^\mu < e^{-\frac{\delta^2\mu}{2}}$$

$$Pr[X > (1 + \delta)\mu] < \left(\frac{e^{\delta}}{(1 + \delta)^{(1+\delta)}}\right)^\mu$$

For instance, if we were to toss a fair coin 10000 times what is the probability that we get heads in less that 4500 tosses. In this case $\mu = 5000$, $\delta = \frac{1}{10}$ and $n = 10000$, and so, by the Chernoff bound, the probability is less than $e^{-25}$.

Let us consider a use of of the Chebychev bound. The problem we will look at is finding the number of satisfying assignments for a DNF boolean formula. For every clause $c_i$ with
r_i literals denote by S_i the number of satisfying assignments. Clearly, |S_i| = 2^{n-r_i}. And so, we denote the number of satisfying solutions for a formula f by \#f = \bigcup_i S_i. Let M be the multiset that contains the elements is all the S_i’s (including repetitions of the same elements). Obviously, |M| = \sum_i |S_i|. For every assignment a define c(a) to be the number of clauses satisfied by a. We wish to choose a random assignment \rho by assigning a probability of \frac{c(a)}{|M|} to every assignment a. First, we randomly choose a clause c_i with probability \frac{|S_i|}{|M|}. We shall now uniformly choose one of the assignments in S_i. We now have that

Pr[assignment a is chosen] = \sum_{a \in S_i} \left( \frac{|S_i|}{|M|} \right) \frac{1}{|S_i|} = \frac{c(a)}{|M|}

For every assignment a we define a random variable X(a) such that X(a) = \frac{|M|}{c(a)} if a is chosen and 0 otherwise. Let X = \sum_a X(a).

**Lemma 3** \( E[X] = \#f \)

This is easy to verify. The proof of the next lemma is omitted.

**Lemma 4** Let \( \mu_k \) be the average of \( k \) independent samples of \( X \). Then, \( \forall \epsilon > 0, \)

\[ Pr[|\mu_k - \#f| \leq \epsilon \#f] \geq \frac{3}{4} \]