CMPUT 675: Approximation Algorithms **Fall 2014**

Lecture 11 (Sep 26): MAX SAT and Random Variables

Lecturer: Zachary Friggstad *Scribe: Antonio Carlos Salzvedel Furtado Junior*

11.1 MAX SAT

In the previous lecture, we defined the MAX SAT problem. We used the following LP relaxation, where saying $x_i \in C$ means literal x_i appears in C and saying $\bar{x}_i \in C$ means literal $\bar{x}_i$ appears in C.

$$
\begin{array}{lrcll}
\text{maximize :} & \sum_C y_C & & & \\
\text{subject to :} & y_C & \leq & \sum_{x_i \in C} z_i + \sum_{\bar{x}_i \in C} (1 - z_i) & \text{for each clause } C \\
& y_C, z_i & \in & [0,1] & \text{for each clause } C \text{ and each variable } x_i
\end{array}
$$

We considered the following simple randomized algorithm and proved Claim 1.

Algorithm 1 Randomized Rounding

Solve the LP to get an optimal solution $(\mathbf{y}^*, \mathbf{z}^*)$.
Independently for each i, set

$$
x_i = \begin{cases} True & \text{with probability } z_i^* \\ False & \text{with probability } 1 - z_i^* \end{cases}
$$

Claim 1 *For any clause C with, say, k literals,*

$$
\Pr[C \textit{ is satisfied }] \geq \left[1 - \left(1 - \frac{1}{k}\right)^k\right] \cdot y_c \geq \left(1 - \frac{1}{e}\right) \cdot y_c.
$$

Now consider the following simple alternative algorithm.

Algorithm 2 Flipping Coins

Independently for each i, set

$$
x_i = \begin{cases} True & \text{with probability } \frac{1}{2} \\ False & \text{with probability } \frac{1}{2} \end{cases}
$$

Claim 2 *For each clause C with, say, k literals,*

$$
\Pr[C \textit{ is satisfied }] = 1 - \frac{1}{2^k}.
$$

Proof. Simply put, each literal in C is false with probability exactly $1/2$. Since the x_i values are sampled independently, C is not satisfied with probability $1/2^k$. ■

This shows that Algorithm 2 is a $1/2$-approximation for this problem since each clause has $k \geq 1$ literals. Therefore, in expectation we satisfy at least half of the clauses. This is tight; consider the MAX SAT instance with a single clause $C = (x_1)$.

Observe that the randomized rounding algorithm (Algorithm 1) favours small clauses while the algorithm that flips unbiased coins (Algorithm 2) favours large clauses. We can exploit this fact to devise an even better algorithm.

Here is the high level idea. The functions $g(k) = 1 - (1 - 1/k)^k$ and $h(k) = 1 - 1/2^k$ are, respectively, bounded by at least $1 - 1/e$ and $1/2$ for positive integers k. However, their "average" $(g(k) + f(k))/2$ is in fact bounded by at least $3/4$ for every positive integer k. So, if we tried both rounding techniques then on average (between the two rounding algorithms), a clause should be satisfied with probability at least $3/4 \cdot y^*_C$.

Algorithm 3 Best of both worlds.

$$\text{run} \begin{cases} \text{Algorithm 1 with probability } \frac{1}{2} \\ \text{Algorithm 2 with probability } \frac{1}{2} \end{cases}$$

Lemma 1 *For any clause C,* $\Pr[C \textit{ is satisfied}] \geq \frac{3}{4} \cdot y^*_c$

Proof.

$$\begin{aligned}
\Pr[C \text{ is satisfied}] &= \Pr[\text{Run Algorithm 1}] \cdot \Pr[C \text{ is satisfied}|\text{Algorithm 1 is run}] + \\
&\quad \Pr[\text{Run Algorithm 2}] \cdot \Pr[C \text{ is satisfied}|\text{Algorithm 2 is run}] \\
&= \frac{1}{2} \cdot Pr[C \text{ is satisfied}|\text{Algorithm 1 is run}] + \frac{1}{2} \cdot Pr[C \text{ is satisfied}|\text{Algorithm 2 is run}] \\
&\geq \frac{1}{2} \cdot \left(1 - \left(1 - \frac{1}{k}\right)^k\right) \cdot y^*_c + \frac{1}{2} \cdot \left(1 - \frac{1}{2^k}\right) \cdot y^*_C \\
&= \frac{g(k) + h(k)}{2} \cdot y^*_C
\end{aligned}$$

where $g(k)$ and $h(k)$ are defined as above (just before the description of Algorithm 3). Note that while Algorithm 2 does not use an LP solution, it is still valid to say that it satisfies a clause C of length k with probability at least $(1 - 1/2^k) \cdot y^*_C$.

Let $f(k) = (g(k) + h(k))/2$. It is straightforward to verify $f(1) = f(2) = 3/4$. For $k \geq 3$, we have $g(k) \geq 1 - 1/e$ and $h(k) \geq 7/8$ so

$$f(k) \geq \frac{1 - 1/e + 7/8}{2} \approx 0.753 > 3/4.$$

That is, for every $k \geq 1$ we have $f(k) \geq 3/4$, meaning clause C is satisfied with probability at least $3/4 \cdot y^*_C$.

■

By Lemma 1, E[# clauses satisfied] $\geq \frac{3}{4} \cdot OPT_{LP}$.

The tightness of this bound can be demonstrated through the following example:

$$(x_1 \vee x_2), (x_1 \vee \bar{x}_2), (\bar{x}_1 \vee x_2), (\bar{x}_1 \vee \bar{x}_2)$$

Any truth assignment satisfies exactly three clauses so the optimum integer solution is 3. However, setting $z_1 = z_2 = 1/2$ and $y_C = 1$ for every clause C is a feasible LP solution with value 4, so $OPT/OPT_{\text{LP}} = 3/4$.

11.1.1 Other Progress

The current best approximation for MAX SAT is a ≈ 0.7968-approximation [ABZ05] and uses a tool called *semidefinite programming*, a generalization of linear programming that we can still solve in polynomial time. We will cover this tool later in the course.

The problem MAX kSAT is the restriction of MAX SAT to instances where each clause involves exactly k literals (from distinct variables). Two particularly interesting versions are MAX 2SAT and MAX 3SAT. For MAX 3SAT, the random coins algorithm (Algorithm 2) provides a 7/8-approximation. Note that this is an oblivious algorithm that does not even look at the clauses! Surprisingly, this is the best possible.

Theorem 1 ([H01]) *Unless P = NP, there is no c-approximation for* MAX 3SAT *for any constant $c > 7/8$.*

MAX 2SAT is also interesting. The decision problem 3SAT is NP-complete, but the decision problem 2SAT is in fact polynomial time solvable. However, MAX 2SAT is NP-hard; if we cannot satisfy all clauses then it is still hard to determine the maximum number that can be satisfied. This should not be too surprising, consider the following analogous situation. We can efficiently determine if a given set of linear equations can be satisfied using Gaussian elimination. However, if they cannot all be satisfied then it is not at all clear how to satisfy "the most" from this method.

Using either Algorithm 1 or 2 on MAX 2SAT still only results in a 3/4-approximation. It is natural to wonder if this is the best we can do, as was the case with MAX 3SAT. The answer is no; we can do better using semidefinite programming. We may see this algorithm later in the course. Will certainly see a similar algorithm for a problem closely related to MAX 2SAT.

11.2 Random Variables

We briefly recalled a few definitions and basic facts about random variables in the lecture. Most of the proof were omitted from the lecture, but they are included here for the curious.

Let X be a *discrete* random variable. In particular, for every real number a, there is some value $\Pr[X = a]$ that says what is the total probability of all events where X takes value a. These values satisfy:

$$\Pr[X = a] \geq 0 \text{ for all values } a \in \mathbb{R}, \qquad \sum_{a \in \mathbb{R}} \Pr[X = a] = 1.$$

Saying that X is discrete means $\Pr[X = a] > 0$ for only finitely (or countably) many values a. Most every time we use random variables in this course, there are only finitely many possible outcomes in the probability space so the random variables will be discrete.

For example, Algorithm 1 defines a distribution over truth assignments to the variables. One random variable could be the number of satisfied clauses. Call this random variable X. Claim 1 says that on average, the value of X is at least $(1 - 1/e) \cdot OPT_{\text{LP}}$. Note that $\Pr[X = a]$ can only be positive for integers a from 0 to the number of clauses.

What, precisely, does the expected value of X mean? It is simple.

Definition 1 $\mathrm{E}[X] = \sum_a a \cdot \Pr[X = a]$

Again, this should be read as the average value of X, where average is weighted by the probability distribution.

One subtle but powerful fact is the following.

Proposition 1 *There is some outcome such that X takes value $\geq \mathrm{E}[X]$. Similarly, there is some outcome such that X takes value $\leq \mathrm{E}[X]$.*

Proof. Suppose the only values a with $\Pr[X = a] > 0$ satisfy $a < \mathrm{E}[X]$. Then

$$\mathrm{E}[X] = \sum_a a \cdot \Pr[X = a] < \sum_a \mathrm{E}[X] \cdot \Pr[X = a] = \mathrm{E}[X] \cdot \sum_a \Pr[X = a] = \mathrm{E}[X].$$

This is impossible. The second statement is similar. ■

For example, while the randomized algorithm Algorithm 3 does not *guarantee* at least $3/4 \cdot OPT_{\mathrm{LP}}$ clauses are satisfied, the simple fact that this is a bound on the expected number of satisfied clauses means that there *must exist* some truth assignment satisfying at least this many clauses. So, it constitutes a proof that $OPT \geq 3/4 \cdot OPT_{\mathrm{LP}}$.

We can construct new random variables from old ones. For example, if X and Y are random variables then so to is $X + Y$. It is the random variable that takes the value of X plus the value of Y on each outcome.

Proposition 2 (Linearity of Expectation) *For two random variables X, Y, over the same probability space, we have*

$$\mathrm{E}[X + Y] = \mathrm{E}[X] + \mathrm{E}[Y].$$

Furthermore, for a random variable X and a constant α we have

$$\mathrm{E}[\alpha X] = \alpha \mathrm{E}[X].$$

Proof. For the first statement:

$$\begin{aligned}
\mathrm{E}[X + Y] &= \sum_{a,b \in \mathbb{R}} (a + b) \cdot \Pr[X = a \text{ and } Y = b] \\
&= \left(\sum_{a,b} a \cdot \Pr[X = a \text{ and } Y = b] \right) + \left(\sum_{a,b} b \cdot \Pr[X = a \text{ and } Y = b] \right) \\
&= \sum_a a \cdot \Pr[X = a] + \sum_b b \cdot \Pr[X = b] \\
&= \mathrm{E}[X]
\end{aligned}$$

For the second statement:

$$\mathrm{E}[\alpha X] = \sum_a \alpha \cdot a \cdot \Pr[X = a] = \alpha \cdot \sum_a a \cdot \Pr[X = a] = \alpha \cdot \mathrm{E}[X].$$

The first equality is because the random variable αX takes value $\alpha \cdot a$ exactly when the random variable X takes the value a (assuming $\alpha \neq 0$, if $\alpha = 0$ then everything is 0 and the proof is trivial). ■

It is important to know when random variables are independent. For example, we used this fact when proving the performance guarantees of the above algorithms.

Definition 2 *X and Y are independent if for all $a, b \in \mathbb{R}$,*

$$\Pr[X = a] \cdot \Pr[Y = b] = \Pr[X = a \textit{ and } Y = b]$$

Proposition 3 *If X and Y are independent, then*

$$\mathrm{E}[X \cdot Y] = \mathrm{E}[X] \cdot \mathrm{E}[Y].$$

More generally, if $X_1, \ldots, X_n$ are independent (meaning any two of them are independent) then $\mathrm{E}[\prod_i X_i] = \prod_i \mathrm{E}[X_i]$.

Proof.

$$\begin{aligned}
\mathrm{E}[X \cdot Y] &= \sum_{a,b} a \cdot b \cdot \Pr[X = a \text{ and } Y = b] \\
&= \sum_{a,b} a \cdot b \Pr[X = a] \cdot \Pr[Y = b] \\
&= \left(\sum_a a \cdot \Pr[X = a]\right) \cdot \left(\sum_b b \cdot \Pr[Y = b]\right) \\
&= \mathrm{E}[X] \cdot \mathrm{E}[Y].
\end{aligned}$$

The second statement follows using this argument in an inductive fashion, along with the simple observation that the two random variables $\prod_{i=1}^{n-1} X_i$ and X_n are also independent. ■

Say that a random variable is *nonnegative* if $\Pr[X = a] > 0$ only for $a \geq 0$.

Theorem 2 (Markov's Inequality) *If X is a nonnegative random variable, then for any $\alpha > 0$ we have*

$$\Pr[X \geq \alpha \cdot \mathrm{E}[X]] \leq \frac{1}{\alpha}.$$

Equivalently,

$$\Pr[X \geq \alpha] \leq \frac{\mathrm{E}[X]}{\alpha}.$$

Proof. The first statement follows immediately from the second statement. To see the first,

$$\begin{aligned}
\mathrm{E}[X] &= \sum_{a \geq \alpha} a \cdot \Pr[X = a] + \sum_{0 \leq a < \alpha} a \cdot \Pr[X = a] \\
&\geq \sum_{a \geq \alpha} \alpha \cdot \Pr[X = a] + 0 \\
&= \alpha \cdot \Pr[X \geq \alpha].
\end{aligned}$$

■

We say that X is a $\{0,1\}$ random variable if $\Pr[X = a] > 0$ only for $a \in \{0,1\}$. For instance, a $\{0,1\}$ random variable can "signal" if a certain event occurs. Sometimes, we identify $\{0,1\}$ random variables with the property itself.

For example, in the MAX SAT problem we implicitly used a $\{0,1\}$ random variable for each clause C to signal when it was satisfied by the assignment. That is, the event "C is satisfied" can also be viewed as a $\{0,1\}$ random variable that is 1 precisely when C is satisfied.

Sometimes we want to avoid a collection of bad events that may not be independent. The following bound is near trivial but it is quite useful in some cases (e.g. next lecture). It says that the probability that *some* bad event happens is upper bounded by the sum of the individual probabilities of the bad events.

Theorem 3 (Union Bound) *Consider any collection $X_1, X_2, ..., X_n$ of {0,1} random variables. Then*

$$\Pr[X_i = 1 \textit{ for some } 1 \leq i \leq n] \leq \sum_{i=1}^{n} \Pr[X_i = 1].$$

Proof. By induction. For $n = 2$ we have

$$\Pr[X_1 = 1 \text{ or } X_2 = 1] = \Pr[X_1 = 1] + \Pr[X_2 = 1] - \Pr[X_1 = 1 \text{ and } X_2 = 1] \leq \Pr[X_1 = 1] + \Pr[X_2 = 1].$$

The first equality uses the principle of inclusion/exclusion.

For general n,

$$\Pr[X_j = 1 \text{ for some } 1 \leq j \leq n] \leq \Pr[X_i = 1 \text{ for some } 1 \leq i \leq n-1] + \Pr[X_n = 1] \leq \sum_{i=1}^{n} \Pr[X_i = 1]$$

The first inequality uses the fact that the union bound is true for 2 random variables and the last follows by induction on first $n - 1$ random variables. ∎

Finally, we state one of the so-called *Chernoff Bounds*. It asserts that for independent $\{0, 1\}$ random variables that the probability of deviating from the expectation decreases exponentially with the deviation. It is a much sharper bound than Markov's inequality, but it requires additional assumptions on the random variables. In fact, it's proof (next lecture) involves using Markov's inequality in a clever way.

Theorem 4 (Chernoff Bound) *Let $X_1, \ldots, X_n$ be independent $\{0, 1\}$ random variables. Then for all $\delta > 0$ and all $U \geq \mathrm{E}[\sum_i X_i]$,*

$$Pr\left[\sum_i X_i > (1 + \delta) \cdot U\right] \leq \left(\frac{e^{\delta}}{(1 + \delta)^{1+\delta}}\right)^U.$$

If $\delta \leq 1$, the latter quantity is bounded by $\leq e^{-U \cdot \delta^2/3}$.

There are similar bounds on the "lower tail" of the distribution (i.e. bounds on $\Pr[\sum_i X_i < (1 - \delta) \cdot L]$ for $L \leq \mathrm{E}[\sum_i X_i]$). They can be found, for example, in the Williamson and Shmoys text.

For a concrete example, consider n unbiased, independent random "coins". That is, each X_i is a $\{0, 1\}$ random variable with $\Pr[X_i = 0] = \Pr[X_i = 1] = 1/2$. Then $\mathrm{E}[\sum_i X_i] = n/2$. Consider $\Pr[\sum_i X_i > \frac{2}{3} \cdot n]$. Using Markov's inequality, the best we can say is that this is at most 3/4.

However, using this Chernoff bound with $U = \mathrm{E}[\sum_i X_i]$ and $\delta = 1/3$ we can bound this probability by $e^{-n/54}$ (using the latter bound for $\delta \leq 1$) which is a much tighter bound for large enough n. Using the first bound gives something slightly sharper: the probability is at most $e^{-n/40}$ (roughly). Even though the first bound is more cumbersome to deal with, in some cases we can use it to get an asymptotically better bound when analyzing approximation algorithms.

11.2.1 Extensions of MAX SAT

Consider the following slight generalization of MAX SAT. The input is the same except each clause C now has a value $v_C \geq 0$. The goal is to satisfy the maximum *total value* of clauses.

Consider the LP relaxation that is identical to what we used, except the objective function is

$$\text{maximize} : \sum_C v_C \cdot y_C.$$

Running Algorithm 1 on this instance yields the same approximation ratio.

That is, Claim 1 shows $\Pr[C \text{ is satisfied}] y_C^* \geq (1 - 1/e) \cdot y_C^*$. Let Y_C be a $\{0, 1\}$ random variable that signals when clause C is satisfied. By linearity of expectation, the expected total *value* of satisfied clauses is bounded as:

$$\mathrm{E}\left[\sum_C v_C \cdot Y_C\right] = \sum_C v_C \cdot \mathrm{E}[Y_C] = \sum_C v_C \cdot \Pr[C \text{ is satisfied}] \geq (1 - 1/e) \cdot \sum_C v_C \cdot y_c^* = (1 - 1/e) \cdot OPT_{\mathrm{LP}}.$$

Essentially the same arguments also show that the performance guarantees of Algorithms 2 and 3 extend to this weighted generalization.

References

ABZ05 A. Avidor, I. Berkovitch, and U. Zwick, Improved approximation algorithms for max nae-sat and max sat, In Proceedings of *Workshop on Approximation and Online Algorithms*, 2005.

H01 J. Håstad, Some optimal inapproximability results, Journal of the ACM, 48(4):798–859, 2001.