MAX-SAT: (Lecture 17: Derandomization)

Given a ~~2 SAT~~ SAT formula ϕ, with n variables, m clauses, and non negative weights w_c on clauses, find an assignment that maximizes weight of satisfied clauses. ~~[Let's assume each clause has > 1 variable]~~

Randomized algo: Set each variable X_i to T or F w/ equal prob.

~~[Then $\mathbb{E}$ [wt. of satisfied clauses]~~

Let $Y_c = +1$ if clause C is satisfied, $= -1$ o.w.

Then $\mathbb{E}\left[\sum_c w_c Y_c\right] = \sum_c w_c \mathbb{E}[Y_c]$

If C has k variables, $\Pr[Y_c = 1] \geq 1 - \frac{1}{2^k}$

Hence exp. wt. of satisfied clauses $\geq \sum_c w_c/2 \geq OPT/2$.

Now suppose we want a deterministic algo that has this guarantee. We will show how to derandomize this algorithm.

A key step will be to evaluate conditional probabilities:

$$\mathbb{E}\left[\sum_c w_c Y_c \mid X_1 \dots X_i\right] = \sum_c w_c \Pr[Y_c = 1 \mid X_1 \dots X_i] \leftarrow \text{can be easily evaluated.}$$

using shorthand, not a r.v.

$= \mathbb{E}\left[\sum_c w_c Y_c \mid X_1 \dots X_i = b_1 \dots b_i\right]$

some string of $\{T, F\}^i$

To evaluate $\Pr[Y_c = 1 \mid X_1 \dots X_i = b_1 \dots b_i]$

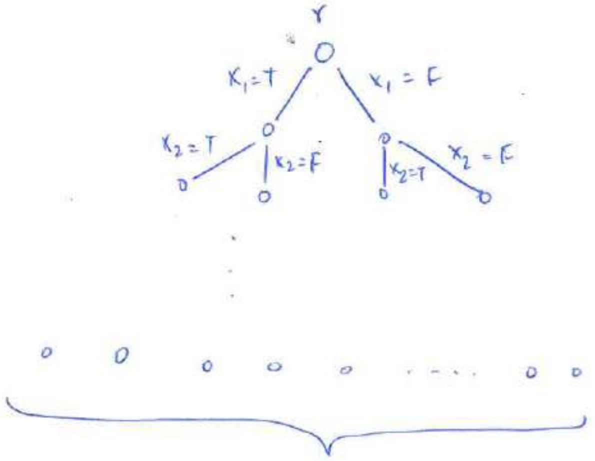
~~For each $j \leq i$~~ if

If C is already satisfied, by the setting of $X_1 \dots X_i$, then $Y_c = 1$. Else if C has r variables from $X_{i+1} \dots X_n$, then $\Pr = 1 - \frac{1}{2^k}$.

So how do we derandomize the algo?

We will choose values for $x_1, x_2, \ldots$ in sequence, ensuring that when we've chosen ~~(x)~~ $x_1, \ldots x_i$, $\mathbb{E}\left[\sum_c w_c Y_c \mid x_1 \ldots x_i\right] \geq \sum_c w_c/2$.

We can think of the algorithm as descending a tree:



For a node v at height i, we've chosen values for $x_1, \ldots x_i$, and the node stores $\mathbb{E}\left[\sum_c w_c Y_c \mid x_1 \ldots x_i\right]$

Suppose at a leaf, $\mathbb{E}\left[\sum_c w_c Y_c \mid x_1 \ldots x_n = b_1 \ldots b_n\right] \geq \sum_c w_c/2$. But then it must be true that, for $x_1 \ldots x_n = b_1 \ldots b_n$, ~~$\sum w_c$~~] wt. of satisfied clauses is at least $\sum_c w_c/2$.

At a node v, let $E_v = \mathbb{E}\left[\sum_c w_c Y_c \mid x_1 \ldots x_i = b_1 \ldots b_i\right]$
at height i

If node v has children b, c, then $E_v = \frac{1}{2} E_b + \frac{1}{2} E_c$, hence $\max\{E_b, E_c\} \geq E_v$

Thus if $E_v \geq$ ~~$\sum w_c$~~ $OPT/2$, then either $E_b \geq OPT/2$
or $E_c \geq OPT/2$.

Starting from the root, we evaluate E_b, E_c for both children, and choose x_i accordingly. We reach a leaf, and this gives us $x_1, \ldots x_n$.

Set Balancing.

Given : A is an $n \times n$ 0-1 matrix

A_i is the i^{th} row of A, A_{ij} are entries.

Problem : Find $b \in \{-1,-1\}^n$ to minimize

$$\|Ab\|_\infty = \max_{i \in [n]} |A_i b|$$

we won't optimize, but will give bounds on how ~~sm~~ large $\min \|Ab\|_\infty$ can be, for <u>any</u> A.

Randomized algo : for each $j \in [n]$, let $b_j \sim \{-1,1\}$ independently with equal probability.

Clearly, for each row i, $\mathbb{E}[A_i b] = 0$

$$\left(\text{since } \mathbb{E}[A_i b] = \mathbb{E}\left[\sum_j A_{ij} b_j\right] = \sum_j \mathbb{E}[A_{ij} b_j] = 0\right)$$

But this does not mean that $\mathbb{E}[|A_i b|] = 0$ ... (why?)

To get a bound on $\mathbb{E}[|A_i b|]$, we will use <u>Hoeffding's inequality</u>

<u>Hoeffding's Inequality:</u> Let $Y_1, \ldots, Y_n$ be independent r.v.'s, with bounded support $[l_i, u_i]$, and let $Y = \sum_i Y_i$. Then:

$$P_r\left[|Y - \mathbb{E}[Y]| > \delta\right] \leq 2e^{-2\delta^2 / \sum_i (u_i - l_i)^2}$$

In our case, let $Y_{ij} = A_{ij} b_j$, and $Y_i = \sum_j A_{ij} b_j$.

Then each $Y_{ij} \in \{-1, 0, +1\}$, $\mathbb{E}[Y_{ij}] = 0$, $\mathbb{E}[Y_i] = 0$, and hence:

$$\Pr\left[|Y_i| \geq \delta\right] \leq e^{-2\delta^2/4n}$$

Suppose we choose $\delta = 2\sqrt{n \log n} = 2\sqrt{n \ln n}$

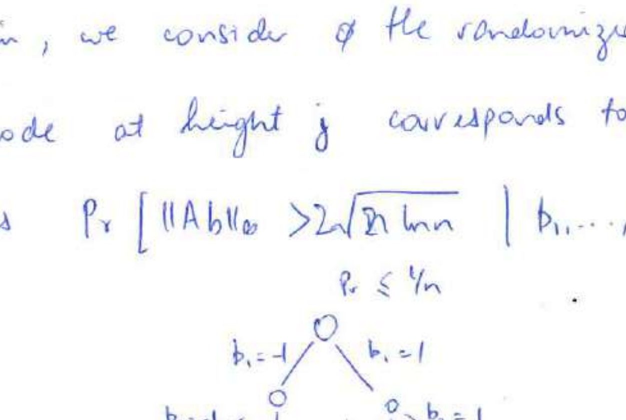
$$\text{Then } \Pr\left[|Y_i| > \delta\right] \leq e^{-4n\ln n/2n} = \frac{1}{n^2}$$

$$\text{Hence, } \Pr\left[\exists i : |Y_i| > \delta\right] \leq \frac{1}{n}$$

Thus by choosing each entry b uniformly from $+1, -1$, we can obtain that $\|Ab\|_\infty \leq 2\sqrt{n \ln n}$ w.h.p.

Can we do this with certainty? Can we derandomize this algorithm?

Again, we consider the randomized algo as a tree, with n levels. A node at height j corresponds to a setting of $b_1, \dots, b_j$, and stores $\Pr\left[\|Ab\|_\infty > 2\sqrt{2n \ln n} \mid b_1, \dots, b_j\right]$.



2^n leaves, for 2^n values of $b_1, \dots, b_n$

~~Since each leaf is equally likely, we know that at~~

Consider a leaf (corr. to some choice of $b_1, \dots, b_n$) s.t.

the value of the leaf is < 1.

Then $\quad \Pr\left[\|Ab\|_\infty > 2\sqrt{n \ln n} \mid b_1 \dots b_n \right] < 1$

But then there is no randomness at the leaf, since all values are already chosen. Hence for this choice of $b_1, \dots, b_n$, it must be true (w.p. 1) that $\|Ab\|_\infty \leq 2\sqrt{n \ln n}$.

For a node v at level j, let $p_v = \Pr\left[\|Ab\|_\infty > 2\sqrt{n \ln n} \mid b_1 \dots b_j \right]$ be the value stored at the node. Note that at the root, r, $p_r \leq \frac{1}{n}$. For a node a at level j w/ children b, c, further

$$p_a = \frac{1}{2} p_b + \frac{1}{2} p_c .$$

Since
$$p_a = \Pr\left[\|Ab\|_\infty > M \mid b_1 \dots b_j \right] = \Pr\left[\|Ab\|_\infty > M \mid b_1 \dots b_{j+1} = +1 \right] \Pr\left[b_{j+1} = 1 \right] + \Pr\left[\|Ab\|_\infty > M \mid b_1 \dots b_{j+1} = -1 \right] \Pr\left[b_{j+1} = -1 \right]$$
$$= \frac{1}{2} p_b + \frac{1}{2} p_c$$

Thus, in particular, either p_b or $p_c \leq p_a$.

~~From~~ Since $p_r \leq 1$, we just follow a path $r, v_1, \dots, v_{n-1}$ where v_{n-1} is a leaf, and $p_{v_{n-1}} \leq p_{v_{n-2}} \leq \dots \leq p_r < 1$. Then at leaf v_{n-1}, for the corresponding choice of b, $\|Ab\|_\infty \leq M = 2\sqrt{n \ln n}$.

However, this depends on being able to calculate $\Pr\left[\|Ab\|_\infty > M \mid b_1 \dots b_j \right]$, which we don't know how to do in polynomial time.

Instead, we use a "pessimistic estimator":

$\hat{p}_v \geq p_v$ for each node v. (hence pessimistic)

~~[Let $\mathcal{E}_i$ be the event that]~~

$$\cancel{\hat{p}_v = \sum_{j \in [} \Pr \{|A_i b| > M\}}$$

$$\hat{p}_v = \sum_{i \in [n]} \Pr\left[|A_i b| > M \mid b_1 \dots b_j\right] \quad , \quad v \text{ is at level } j.$$

and this we can calculate; Further, can show that if v has

~~Fix i, k. Then $\Pr[|A_i b| > M \mid b_1 \dots b_j]$ Let $S_i = \{$~~

~~The~~

children c, d, then $\min\{\hat{p}_c, \hat{p}_d\} < \hat{p}_v$.

Since $\hat{p}_r < 1$ (by our proof technique that shows $\Pr < 1$), we can then follow the path downwards to obtain a leaf s.t. $\hat{p}_v < 1$.

$$\Pr\left[\,|A_i b| > M\,\right] \qquad M \geq 0$$

$$= \Pr\left[A_i b > M\right] + \Pr\left[A_i b < M\right]$$

let $k = \sum_{j' \leq j} A_{ij'}$

then $\Pr\left[\,|A_i b|\right.$

$$\Pr\left[\,|A_i b| > M\,\right] = \sum_{R = M+1}^{n} \Pr\left[\,|A_i b| = R\,\right]$$

now $\Pr\left[\,|A_i b| = R\,\right] = \Pr\left[A_i b = R\right] + \Pr\left[A_i b = -R\right]$

let $K = \sum_{j' \leq j} A_{ij'}$, $S_i = \{ j' > j : A_{ij'} = 1 \}$

Then $\Pr\left[A_i b = R \mid b_1 \dots b_j\right] = \Pr\left[\sum_{j \in S_i} A_{ij} b_j = R - K\right] = \Pr\left[\sum_{j \in S_i} b_j = R - K\right]$

say in S_i, n_1 coordinates of b are $+1$,
$|S_i| - n_1$ " " " are -1.

Then $\sum_{j \in S_i} A_{ij} b_j = \sum_{j \in S_i} b_j = 2n_1 - |S_i| = R - K$

$$\Rightarrow \quad n_1 = \tfrac{1}{2}\left(R - K + |S_i|\right)$$

and thus, $\Pr\left[A_i b = R \mid b_1 \dots b_j\right] = \binom{|S_i|}{\frac{1}{2}(R - K + |S_i|)} \dfrac{1}{2^{|S_i|}}$

we can thus obtain $\Pr\left[\,|A_i b| > M \mid b_1 \dots b_j\right]$ which is $= \dfrac{\text{sum of binom. coeffs.}}{2^{|S_i|}}$