MAX SAT:

input: Boolean formula f in CNF with n variables $x_1, \dots, x_n$
& m clauses $C_1, \dots, C_m$

output: assignment maximizing the # of clauses satisfied.

Example:

$$f = (x_1 \vee \overline{x_3} \vee x_4) \wedge (x_2 \vee x_3) \wedge (\overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3} \vee x_2 \vee x_4) \wedge (\overline{x_4})$$

Setting $x_1 = F, x_2 = F, x_3 = T, x_4 = F$ satisfies 4 of the 5 clauses, and there is no assignment satisfying all 5.

We saw a $\frac{1}{2}$-approximation algorithm.
In fact, the algorithm satisfies $\geq \frac{m}{2}$ clauses (regardless of the max # of clauses satisfiable)
& if there are exactly k literals in every clause then it satisfies $\geq (1 - 2^{-k})m$ clauses.

Integer linear Programming (ILP) is a linear program where the variables are restricted to integer values.

ILP is NP-complete.

Let's see: SAT $\rightarrow$ ILP.

For variable x_i in SAT input f,
create variable y_i for our ILP instance.
Restrict $y_i \in \{0,1\}$ where
$y_i = 1$ corresponds to $x_i = T$
& $y_i = 0$ " $x_i = F$.

For clause C_j
create variable $z_j \in \{0,1\}$
where $z_j = 1$ corresponds to C_j satisfied
& $z_j = 0$ " C_j unsatisfied

Further, let C_j^+ = variables in C_j in positive form
C_j^- = " C_j in negative form

For $C_j = (x_5 \vee \overline{x_3} \vee x_7)$, $C_j^+ = \{x_5, x_7\}$, $C_j^- = \{x_3\}$

max-SAT $\rightarrow$ ILP:

Consider input f for max-SAT with variables $x_1, \ldots, x_n$ & clauses $C_1, \ldots, C_m$.

Create the following ILP instance:

$$\text{Maximize} \sum_{j=1}^{m} z_j$$

subject to:

for all $i=1,\ldots,n$, $\quad y_i \in \{0,1\}$

for all $j=1,\ldots,m$, $\quad z_j \in \{0,1\}$

for all $j=1,\ldots,m$, $\quad \sum_{i \in C_j^+} y_i + \sum_{i \in C_j^-} (1-y_i) \geq z_j$

For the last constraint to get $z_j = 1$ we need ≥ 1 positive literal having $y_i = 1$ &/or ≥ 1 negative literal with $y_i = 0$.

Since we maximize $\sum_j z_j$ we'll set $z_j = 1$ if possible.

Solving this ILP will give a solution to max-SAT.

Consider the following LP which changes constraints of the form $y_i \in \{0,1\}$ to $0 \le y_i \le 1$

LP:

$$\max \sum_{j=1}^{m} \hat{z}_j$$

Subject to:

for all $i = 1, \ldots, n$: $0 \le \hat{y}_i \le 1$

for all $j = 1, \ldots, m$: $0 \le \hat{z}_j \le 1$

$$\sum_{i \in C_j^+} \hat{y}_i + \sum_{i \in C_j^-} (1 - \hat{y}_i) \ge \hat{z}_j$$

This is a LP so we can solve it in poly-time.

This is a "relaxation" of the original <u>ILP</u> in the following sense:

any solution to the ILP is also a solution to the LP

hence, objective value for optimal of ILP $\le$ objective value for optimal of LP.

$$\sum_j z_j^* \le \sum_j \hat{z}_j^*$$

Take optimal solution for LP, call it
$\hat{y}^*$ & $\hat{z}^*$

We want to then create a solution y, z for the ILP which is close to the optimal for the ILP. How do we measure close to the ILP optimal? By saying it's close to the LP optimal.

Take $\hat{y}^*$ & $\hat{z}^*$.

Set $y_i = \begin{cases} 1 & \text{with probability } \hat{y}_i^* \\ 0 & \text{" \quad " \quad } 1-\hat{y}_i \end{cases}$

This is called "randomized rounding."

Let's look at expected # of satisfied clauses.

For $k \geq 1$, let $\beta_k = 1-\left(1-\frac{1}{k}\right)^k$

<u>Lemma</u>: For clause C_j with k literals,

$$Pr(C_j \text{ is satisfied}) \geq \beta_k \hat{z}_j$$

Note, $1-\frac{1}{k} \le e^{-k}$ for $k \ge 1$ since $e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} \pm \dots$

hence, $1-\left(1-\frac{1}{k}\right)^k \ge 1-\frac{1}{e}$ for $k \ge 1$

Let $Z = \#$ of satisfied clauses.

$$E[Z] = \sum_{j=1}^{m} \Pr(C_j \text{ is satisfied})$$

$$\ge \sum_{j=1}^{m} \beta_{k(j)} \hat{z}_j \quad \text{where } C_j \text{ has } k(j) \text{ variables.}$$

$$\ge \left(1-\frac{1}{e}\right) \sum_{j=1}^{m} \hat{z}_j$$

Recall, $\sum_j \hat{z}_j \ge \sum z_j = m^* =$ the optimal # of satisfied clauses

Hence, $E[Z] \ge \left(1-\frac{1}{e}\right) m^*$.

So in expectation we satisfy $\ge \left(1-\frac{1}{e}\right)$ times the maximum # of satisfied clauses.

This is a $\left(1-\frac{1}{e}\right)$-*expected* approximation algorithm, and we can find such an assignment using the *method of conditional expectations* that we saw before for Max-cut.

<u>Proof of lemma:</u> Fix C_j.

Suppose all of the variables in C_j are in positive form, so say $C_j = (x_1 \vee x_2 \vee \cdots \vee x_k)$.

The LP constraint is then:

$$\hat{y}_1 + \hat{y}_2 + \cdots + \hat{y}_k \geq \hat{z}_j \qquad (*)$$

Clause C_j is unsatisfied if every y_i for $i=1,\dots,k$ is rounded to 0.

This happens with probability $\prod_{i=1}^{k}(1-\hat{y}_i)$

$$\Pr(C_j \text{ is unsatisfied}) = \prod_{i=1}^{k}(1-\hat{y}_i)$$

<u>we need to show:</u> $\leq \beta_k \hat{z}_k$

Recall the arithmetic mean - geometric mean inequality

AM - GM

$$AM = \frac{1}{k}\sum_{i=1}^{k} w_i \geq \left(\prod_{i=1}^{k} w_i\right)^{1/k} = GM$$

in our case let $w_i = 1-\hat{y}_i$.

Then, $\prod_{i=1}^{k}(1-\hat{y}_i) \le \left[\frac{1}{k}\sum_{i=1}^{k}(1-\hat{y}_i)\right]^k$

$$= \left[1 - \frac{\sum_{i=1}^{k}\hat{y}_i}{k}\right]^k$$

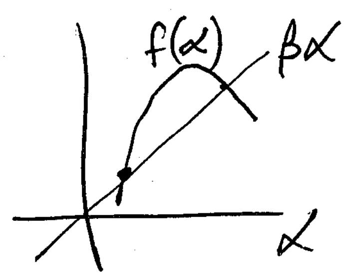
Thus, $1-\prod_{i=1}^{k}(1-\hat{y}_i) \ge 1-\left(1-\frac{\sum_{i=1}^{k}\hat{y}_i}{k}\right)^k$

$$\ge 1-\left(1-\frac{\hat{z}_j}{k}\right)^k$$

since $\hat{y}_1+\hat{y}_2+\cdots+\hat{y}_k \ge \hat{z}_j$ from (*) on last page.

Let $f(\alpha)=1-(1-\frac{\alpha}{k})^k$

$f''(\alpha)<0$ so $f(\alpha)$ is concave



to show $f(\alpha)\ge\beta_k\alpha$ for all $\alpha\in[0,1]$ we just need to check $\alpha=0$ & $\alpha=1$:

for $\alpha=0$: $f(\alpha)=1-(1-\frac{0}{k})^k=0=\beta_k\alpha$ ✓

for $\alpha=1$: $f(\alpha)=1-(1-\frac{1}{k})^k=\beta_k\alpha$ ✓

So $f(\alpha)\ge\beta_k\alpha$ for $\alpha\in[0,1]$.

Hence, $f(\hat{z}_j) \geq \beta_k \hat{z}_j$

Therefore,

$$\Pr(C_j \text{ is satisfied}) = 1 - \prod_{i=1}^{k}(1-\hat{y}_i)$$

$$\geq 1 - \left(1 - \frac{\hat{z}_j}{k}\right)^k \quad \text{by AM-GM}$$

$$\geq \beta_k \hat{z}_j \quad \text{since } f(\alpha) \geq \beta_k \alpha$$

$\boxtimes$

Recall, our old $\frac{1}{2}$-approx. alg. achieves $1-2^{-k}$ on clauses of size k

& this new alg. achieves $\beta_k = 1-\left(1-\frac{1}{k}\right)^k$

k	old	new
1	½	1
2	¾	$\frac{3}{4}$
3	⅞	$1-\left(\frac{2}{3}\right)^3 \simeq .704$
⋮	⋮	

new is better for $k \leq 2$.

old is better for $k \geq 2$

Best of 2 algorithms:

Run best both algorithms & take the better of the 2 solutions.

Let m_1 = expected # of clauses satisfied by old algorithm
& m_2 = expected # for new algorithm.

Theorem:

$$\max\{m_1, m_2\} \geq \frac{3}{4} \sum_{j=1}^{M} \hat{z}_j \geq \frac{3}{4} m^*$$

So we have a $\frac{3}{4}$-approximation algorithm.

(Can use method of conditional expectations to derandomize)

<u>Proof:</u> $\max\{m_1, m_2\} \geq \dfrac{m_1 + m_2}{2} = \text{average}(m_1, m_2)$

So it suffices to show $\dfrac{m_1+m_2}{2} \geq \dfrac{3}{4}$

Let S_k = set of clauses with k literals.

$$m_1 = \sum_k \sum_{C_j \in S_k} \left(1 - 2^{-k}\right) \geq \sum_k \sum_{C_j \in S_k} \left(1 - 2^{-k}\right) \hat{z}_j$$

Since $\hat{z}_j \leq 1$

$$m_2 \geq \sum_k \sum_{C_j \in S_k} \left(1 - \left(1 - \tfrac{1}{k}\right)^k\right) \hat{z}_j$$

Thus,

$$\frac{m_1+m_2}{2} \geq \sum_k \sum_{C_j \in S_k} \frac{\left(1-2^{-k}\right) + \left(1 - \left(1 - \frac{1}{k}\right)^k\right)}{2} \hat{z}_j$$

Need to show:

$$\alpha_k \geq \frac{3}{4} \text{ for all } k \geq 1.$$

$\underline{k=1:}$ $\quad \alpha_k = \dfrac{\frac{1}{2}+1}{2} = \dfrac{3}{4}$ ✓

$\underline{k=2:}$ $\quad \alpha_k = \dfrac{\frac{3}{4}+\frac{3}{4}}{2} = \dfrac{3}{4}$ ✓

$\underline{k\geq 3:}$

$$\alpha_k \geq \frac{\frac{7}{8}+\left(1-\frac{1}{e}\right)}{2} = \frac{1}{2}\left(\frac{15}{8}-\frac{1}{e}\right)$$

$$\geq \frac{1}{2}\left(\frac{6}{4}\right) \quad \text{since } \underset{.368}{\frac{1}{e}} \leq \underset{.375}{\frac{3}{8}}$$

$$= \frac{3}{4}$$ ✓