Max-SAT:

input: Boolean formula f in CNF with variables $x_1, \dots, x_n$ & 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 clauses & no assig. satisfies all 5.

Max-SAT is NP-hard so can't hope to solve exactly

Lemma 1: For a CNF f, there is an assignment satisfying $\geq \frac{m}{2}$ clauses

Lemma 2: If every clause has size $= k$, then there is an assignment satisfying $\geq (1 - 2^{-k})$ clauses.

Proof: Fix f.

Random assignment: set $x_i = \begin{cases} T & \text{with prob. } 1/2 \\ F & \text{with prob. } 1/2 \end{cases}$

Let Y = # of satisfied clauses.

For clause C_i, let $Y_i = \begin{cases} 1 & \text{if } C_i \text{ is satisfied} \\ 0 & \text{if not} \end{cases}$

Then, $Y = \sum_{i=1}^{m} Y_i$

$$E[Y] = \sum_i E[Y_i] = \sum_i \Pr(Y_i = 1) = \sum_i (1 - 2^{-k_i})$$

where $k_i = |C_i|$.

this proves lemma 2.

$$E[Y] = \sum_i (1 - 2^{-k_i}) \geq \sum_i \frac{1}{2} \quad \text{since } k_i \geq 1$$

$$= \frac{m}{2}$$

$\boxtimes$

This gives a randomized algorithm that approximates (i) Max-SAT within $\geq \frac{1}{2}$ of optimal & (ii) Max-Exact-k-SAT within $\geq 1-2^{-k}$

Can get a deterministic algorithm that achieves (i) by "method of conditional expectations."

$$E[Y] = E[Y \mid X_1 = T] \times Pr(X_1 = T) + E[Y \mid X_1 = F] \times Pr(X_1 = F)$$
$$= \frac{1}{2}\left(E[Y \mid X_1 = T] + E[Y \mid X_1 = F]\right)$$

Since $E[Y] \geq m/2$ then
$$\max\{E[Y \mid X_1 = T], E[Y \mid X_1 = F]\} \geq \frac{m}{2}.$$

Note, we can compute $E[Y \mid X_1 = T]$ & $E[Y \mid X_1 = F]$
$\Rightarrow$ simplify the formula after setting x_1 & then $E[Y]$ is a function of the size of each clause.

After setting x_1, then repeat for x_2, etc.

Alg. to achieve assignment satisfying $\geq \frac{m}{2}$ clauses:

For $i = 1 \to n$:

- Try $x_i = T$ & $x_i = F$
  - for each compute:

    $E[Y \mid \text{setting for } x_1, \dots, x_i]$

    = expected # of satisfied clauses for random to $x_{i+1}, \dots, x_n$ & fixed to $x_1, \dots, x_i$

- Take better of two assignments for x_i & fix then repeat for x_{i+1}

Alternative randomized algorithm:

Integer Programming: (or integer linear programming)

$$\max \; c^T x$$
$$\text{s.t.} \quad Ax \le b$$
$$x \ge 0$$
$$\& \; x \in \mathbb{Z}^n \text{ (each variable } x_i \text{ is integer-valued)}$$

ILP is NP-hard: SAT $\rightarrow$ ILP.

- Take input f for SAT
- For each $x_i \in f$, create variable y_i for our ILP.
  Add constraint $0 \le y_i \le 1$
  (so that in ILP, $y_i \in \{0,1\}$)
  $y_i = 1$ corresponds to $x_i = T$
  $y_i = 0$ " $x_i = F$.
- For clause C_j,
  add variable z_j & $0 \le z_j \le 1$
  ($z_j = 1$ corresponds to C_j satisfied
  $z_j = 0$ " C_j unsatisfied)

For clause C_j let:

C_j^+ denote those variables appearing in positive form in C_j

& C_j^- denote those in negative form.

For example, $C_j = (x_5 \vee \bar{x}_3 \vee x_7)$

then $C_j^+ = \{x_5, x_7\}$ & $C_j^- = \{x_3\}$.

Note that: $y_5 + (1 - y_3) + y_7 \geq 1$

for $y_i \in \{0, 1\}$

means that either $y_5 = 1$, $y_3 = 0$, &/or $y_7 = 1$,

(thus C_j is satisfied)

— So for each clause C_j,

add the constraint:

$$\sum_{i \in C_j^+} y_i + \sum_{i \in C_j^-} (1 - y_i) \geq 1$$

In summary, for SAT input f,
consider the ILP:

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

s.t. $\forall_{1 \le i \le n}, 0 \le y_i \le 1$

$\forall_{1 \le j \le m}, 0 \le z_j \le 1$

and $\sum_{i \in C_j^+} y_i + \sum_{i \in C_j^-} (1 - y_i) \ge z_j$

& y_i's & z_j's are integers.

This ILP is equivalent to Max-SAT.
& hence we've shown Max-SAT $\rightarrow$ ILP.

Drop the integer constraints & we have an LP.
this is the <u>LP relaxation</u>

Solve this LP.

Let $\hat{y}_i^*$ & $\hat{z}_j^*$ be the optimal <u>LP</u> solution which we have.

Let y_i^* & z_j^* be the optimal <u>ILP</u> solution which we don't have.

Can we use $\hat{y}_i^*$ & $\hat{z}_j^*$ to approximate y_i^* & z_j^*

How do they compare?

Any ILP feasible solution is a LP feasible,

hence: $$\sum_j z_j^* \leq \sum_j \hat{z}_j^*$$

ILP optimum $\leq$ LP optimum

that's why we say the LP is a relaxation

We want a feasible ILP point = valid assignment for f.

<u>Goal</u>: change fractional ~~ILP~~ LP $\hat{y}_i^*, \hat{z}_j^*$ to integer y_i, z_j & then show this is close to optimal y_i^*, z_j^*

Consider optimal LP solution $\hat{y}_i^*, \hat{z}_j^*$

We'll construct a feasible ILP point:

let $y_i = \begin{cases} 1 & \text{with prob. } \hat{y}_i^* \\ 0 & \text{with prob. } 1-\hat{y}_i^* \end{cases}$

this is valid since $0 \le \hat{y}_i^* \le 1$

Called <u>randomized rounding</u>.

Let W = # of satisfied clauses where we set $x_i = T$ if $y_i = 1$
& $x_i = F$ if $y_i = 0$.

let $W_j = \begin{cases} 1 & \text{if clause } C_j \text{ is satisfied} \\ 0 & \text{if not.} \end{cases}$

Then, $W = \sum_{j=1}^{m} W_j$

Lemma: For C_j with k literals,

$$E[W_j] \geq \beta_k \hat{z}_j^* \text{ where } \beta_k := 1 - \left(1 - \frac{1}{k}\right)^k$$

Note, $\beta_k = 1 - \left(1 - \frac{1}{k}\right)^k \geq 1 - \frac{1}{e}$ for $k \geq 1$

Since $1 - \frac{1}{k} \leq \cancel{\;} e^{-1/k}$

So: $\left(1 - \frac{1}{k}\right)^k \leq 1/e$

Therefore,

$$E\left[\begin{array}{c}\#\text{ of}\\ \text{satisfied clauses}\end{array}\right] = E[W] \qquad (\text{Def'n. of } W)$$

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

$$\geq \sum_j \beta_{k_j} \hat{z}_j^* \qquad (\text{by the lemma})$$

$$\geq \left(1-\frac{1}{e}\right)\sum_j \hat{z}_j^* \qquad \left(\text{by above observation } \beta_k \geq 1 - 1/e\right)$$

$$\geq \left(1-\frac{1}{e}\right)\sum_j z_j^* \qquad \left(\text{since ILP optimum} \leq \text{LP opt. } \sum z_j^* \leq \sum \hat{z}_j^*\right)$$

$$\geq \left(1-\frac{1}{e}\right)\left(\begin{array}{c}\text{max \# of}\\ \text{satisfied clauses}\end{array}\right)$$

So we satisfy $\geq \left(1-\frac{1}{e}\right)$ fraction of the max # of satisfied clauses

$\Rightarrow \left(1-\frac{1}{e}\right)$-approximation algorithm.

Proof of Lemma:

Fix C_j

Consider C_j & ignore other clauses, so can consider all variables in C_j to be positive form

Say $C_j = (x_1 \vee x_2 \vee \cdots \vee x_k)$

LP constraint is: $\hat{y}_1^* + \hat{y}_2^* + \cdots + \hat{y}_k^* \geq \hat{z}_j^*$

$\Pr(C_j \text{ is unsatisfied})$

$= \Pr(\text{all } y_i \text{ set to } 0)$

$= \prod_{i=1}^{k} (1 - \hat{y}_i^*)$

AM-GM = arithmetic mean - geometric mean inequality:

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

let $w_i = 1 - \hat{y}_i^*$

then,

$$\prod_{i=1}^{k} (1 - \hat{y}_i^*) \leq \left[\frac{1}{k} \sum_{i=1}^{k} (1 - \hat{y}_i^*) \right]^k = \left[1 - \frac{\sum \hat{y}_i^*}{k} \right]^k$$

So we have:

$$\Pr(C_j \text{ is satisfied})$$

$$= 1 - \prod_{i=1}^{k}(1-\hat{y}_i^*)$$

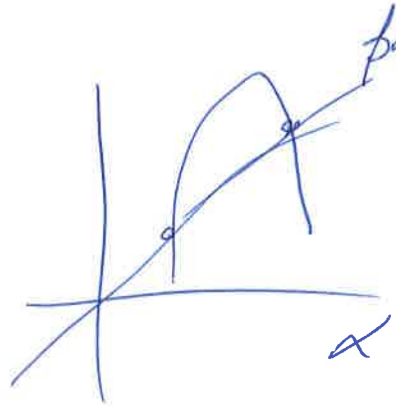
$$\geq 1 - \left(1 - \frac{\sum \hat{y}_i^*}{k}\right)^k$$

$$\geq 1 - \left(1 - \frac{\cancel{\sum}\hat{z}_j^*}{k}\right)^k \quad \text{since } \hat{y}_1^* + \cdots + \hat{y}_k^* \geq \hat{z}_j^*$$

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

$$f''(\alpha) < 0$$

so $f(\alpha)$ is concave



hence to show $f(\alpha) \geq p_k \alpha$ for all $\alpha \in [0,1]$

we just need to check for $\alpha = 0$ & $\alpha = 1$:

$$\alpha = 0: f(\alpha) = 1 - \left(1 - \frac{0}{k}\right)^k = 0 = p_k\alpha. \checkmark$$

$$\alpha = 1: f(\alpha) = 1 - \left(1 - \frac{1}{k}\right)^k = p_k\alpha \checkmark$$

Finally we have:

$$\Pr(C_j \text{ is satisfied})$$

$$= 1 - \prod_{i=1}^{k} (1 - \hat{y}_i^*)$$

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

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

for $\beta_k = 1 - \left(1 - \frac{1}{k}\right)^k$. $\blacksquare$

We now have 2 algorithms:

Simple achieves $(1-2^{-k})$-approx on clauses of size k

& LP gets $\beta_k = 1-(1-\frac{1}{k})^k$

k	Simple	LP	max	avg
1	$\frac{1}{2}$	1	1	$\frac{3}{4}$
2	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$
3	$\frac{7}{8}$	$1-(\frac{2}{3})^3 \approx .704$	$\frac{7}{8}$	$> \frac{3}{4}$
$k \geq 4$			$1-2^{-k}$	

Best of 2 algorithms:

Run both algorithms – simple & LP

Let $x_1^1, \ldots, x_n^1$ & $x_1^2, \ldots, x_n^2$ be the 2 solutions

Output the better of the two.

Let m_1 = expected # of satisfied clauses by simple

& m_2 = 〃 by LP

m^* = optimal # of satisfied clauses.

Theorem: $\max\{m_1, m_2\} \geq \frac{3}{4} m^*$

Hence this is a $\frac{3}{4}$-approx. algorithm.

Proof:

$$\max\{m_1, m_2\} \geq \text{avg}(m_1, m_2) = \frac{m_1 + m_2}{2}$$

So it suffices to show that: $\frac{m_1+m_2}{2} \geq 3/4$

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

since $\hat{z}_j^* \leq 1$

S_k = clauses of size k

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

Then,
$$\frac{m_1+m_2}{2} \geq \sum_k \sum_{C_j \in S_k} \underbrace{\frac{(1-2^{-k}) + \left(1-\left(1-\frac{1}{k}\right)^k\right)}{2}}_{\geq \frac{3}{4} \text{ for all } k \geq 1} \hat{z}_j^* \geq \sum_k \sum_{C_j \in S_k} \frac{3}{4} \hat{z}_j^*$$

$$\geq \frac{3}{4} \hat{m}^* \geq \frac{3}{4} m^*$$