Max SAT — Given a set of \( m \) clauses (CNF)
in \( n \) Boolean variables \( x_1, \ldots, x_n \), find True/False assignment to variables to make a max. number of clauses true.

e.g. \( (\overline{a}_1 \lor a_2) \ (a_1 \lor a_2) \ (\overline{a}_1 \lor \overline{a}_3) \ (\overline{a}_3) \)
to make all \( T \) \( a_2 = F \ a_2 = T \) but can’t get 2 middle clauses.
Max is 3.

Recall. Even Max 2-SAT is NP-hard

Today: a randomized poly time alg. for Max SAT with expected approx. factor \( 3/4 \). It can be derandomized.

We saw a randomized alg. to satisfy \( \frac{m}{2} \) clauses in expected case.

- set each variable True/False with Prob. \( \frac{1}{2} \)

Analysis: For any clause \( C \)
\[
\Pr [\text{C is satisfied}] = 1 - \frac{1}{2^t} \quad t = \text{# variables in } C
\geq 1 - \frac{1}{2} = \frac{1}{2}
\]

So \( E [\# \text{ clauses satisfied}] \geq \frac{m}{2} \)

This gives expected \( \frac{1}{2} \)-approx. since \( m \geq \text{OPT} \)

Note: this alg. is better for large clauses

We will combine this with a method that is better for small clauses

First: how to de-randomize this alg. to get

a deterministic \( \frac{1}{2} \)-approx alg. for Max SAT,

shown by means of an example.
Example clauses \( \overline{a_1}, a_1 \vee \overline{a_2}, a_1 \vee a_2 \vee \overline{a_3}, \overline{a_1} \vee a_2 \vee \overline{a_3}, a_1 \vee a_2 \vee a_3 \)

Expected # clauses satisfied
\[
\frac{1}{2} + \frac{3}{4} + \frac{7}{8} + \frac{7}{8} + \frac{7}{8} = \frac{37}{8}
\]

\( (1 - \frac{1}{2^t}) \) t=# variables

If we set \( a_1 = \text{False} \) we get expectation
\[
1 + \frac{1}{2} + \frac{3}{4} + 1 + \frac{3}{4} = \frac{11}{2}
\]

1 if \( \overline{a_1} \) in clause; else \( (1 - \frac{1}{2^t}) \) counting \( a_1 \)

If we set \( a_1 = \text{True} \) we get expectation
\[
0 + 1 + 1 + \frac{3}{4} + 1 = \frac{33}{4}
\]

Note: \( \overline{E_{sat}} = \frac{1}{2} E_{sat} (a_1 = F) + \frac{1}{2} E_{sat} (a_1 = T) \)

So one choice gives expectation at least \( \overline{E_{sat}} \)
\[
\frac{37}{8} = \frac{1}{2} \cdot 4 + \frac{1}{2} \cdot \frac{33}{4}
\]

We pick \( a_1 = \text{False} \) because it gives higher expectation.

Continue to \( a_2 \)

Setting \( a_2 \) to False gives expectation
\[
1 + 1 + \frac{1}{2} + 1 + \frac{1}{2} = 4
\]

Setting \( a_2 \) to True gives
\[
1 + 0 + 1 + 1 + 1 = 4
\]

Here all clauses are satisfied and we have 4 satisfied clauses, this example can be generalized. (out of 5)
Improved Approx. Alg. for Max SAT.
- formulate as Integer Linear Program (ILP)
- Solve LP relaxation. Use “randomized rounding”.

Make variables
\[ x_i \quad i = 1 \ldots n \quad \text{for each Boolean variable} \quad a_i \]
\[ y_i \quad i = 1 \ldots m \quad \text{for each clause} \]

\[
\max \sum y_i \\
\text{one constraint per clause}
\]

\[ c_1 = (\overline{a}_1 \lor a_2) \]
\[ y_1 \leq (1-x_1) + x_2 \quad \text{-- i.e. in order to set} \]
\[ y_i = 1 \quad \text{we need} \quad x_2 = 1 \]
\[ \text{or} \quad x_1 = 0 \quad \text{(or both)} \]

\[ 0 \leq x_i \leq 1 \quad i = 1 \ldots n \]
\[ 0 \leq y_i \leq 1 \quad i = 1 \ldots m \quad \text{this is LP relaxation} \]

If \[ x_i, y_i \in \{0,1\} \] then this ILP is exactly MaxSAT

Use poly-time LP alg. to solve the LP
- Suppose \( \hat{x}_i, \hat{y}_i \) is solution
- Set \( a_i = \hat{x}_i \) with Prob \( \hat{x}_i \), “randomized rounding”
- This gives a truth value assignment
Analysis for Max 2-SAT.

e.g. $C_2 = (a_1 \lor a_2)$

In LP:

$$\hat{y}_2 \leq \hat{x}_1 + \hat{x}_2$$

$$\text{Prob} \left[ C_2 \text{ is satisfied} \right] = \hat{x}_1 + \hat{x}_2 - \hat{x}_1 \hat{x}_2$$

$$\geq \hat{x}_1 + \hat{x}_2 - \left( \frac{\hat{x}_1 + \hat{x}_2}{2} \right)^2$$

since geometric mean \leq arithmetic mean

$$\sqrt{\hat{x}_1 \hat{x}_2} \leq \frac{\hat{x}_1 + \hat{x}_2}{2}$$

since $\hat{y}_2 \leq \hat{x}_1 + \hat{x}_2 \leq 2$

and the function $x \to \frac{x^2}{4}$ is increasing for $x \in [0,2]$.

$$\Rightarrow \hat{y}_2 - \frac{\hat{y}_2^2}{4} \leq 0 \leq \hat{y}_2 \leq 1$$

$$\Rightarrow \frac{3}{4} \hat{y}_2 \leq \hat{y}_2$$

Thus $\frac{3}{4} \hat{y}_2 \leq \hat{y}_2$

Expected # clauses satisfied:

$$\leq \frac{3}{4} \sum \hat{y}_i = \frac{3}{4} \text{OPT}_{LP} \leq \frac{3}{4} \text{OPT}_{ILP} = \frac{3}{4} \text{OPT}_{\text{Max SAT}}$$

Analysis of general case (which we won't do)

shows this method is better for small clauses.

Previous alg. better for large clauses.

Can combine & de-randomize to get approx factor $\frac{3}{4}$

for general Max SAT.

Best known: Goemans & Williamson '94

0.878 approx factor.

Lower bound: no approx factor $\geq 0.942$ unless $P = NP$. 


We've seen:

- metric TSP \( \approx 1.5 \) approx (we saw 2-approx)
- vertex cover \( \approx 2 \)
- set cover \( \approx O(\log n) \) and \( O(f) \)

\[
\begin{align*}
\text{max cut} & \approx \frac{1}{2} \\
\text{max SAT} & \approx \frac{3}{4}
\end{align*}
\]

Questions
- which problems have constant-factor approx algos
- how close to 1 can constant be?

Note: These questions are only relevant assuming \( P \neq NP \).

Next day: problems where approx-factor can get arbitrarily close to 1
- **PTAS** poly-time approximation scheme (of course run-time increases)

**Hardness of Approximation**

results of the form:

If we could approx problem \( Q \) in poly.time with constant factor/PTAS, then \( P = NP \).

Some results like this are straightforward.

**Example.** Recall from 1st lecture: 2-approx for Travelling Salesman Problem for points in plane.

**Lemma** If we could 2-approx general TSP in poly. time then \( P = NP \),

because could then solve Hamiltonian cycle in poly. time.
other hardness of approx. results are harder.
Breakthrough result '92.
If Max 3-SAT has PTAS then P = NP.
This is a bit like first NP-hardness result.
To leverage it, we need reductions that preserve good approx.

Example

Thm. Poly-time $\alpha$-approx. for Ind. Set $\implies$ poly-time
$\omega$-approx. for Max 3-SAT.

Proof of Thm

We want poly-time reduction Max 3-SAT $\leq$ Ind. Set
That preserves approx. factor
Given Max 3-SAT formula $F$ transform to Ind. set problem
The standard reduction works.

Recall that reduction

$C = (x_1 \lor \overline{x_2} \lor \overline{x_3})$ becomes $\triangle$

\[ x_1 \]
\[ \overline{x_2} \]
\[ x_3 \]

\[ \overline{x_3} \]

Put edge $(x, \overline{x})$ for all occurrences.
Graph $G$ with $3m$ vertices, $m = \#$ clauses.

Truth value assign.

Satisfies $k$ clauses $\iff$ maps to Ind. Set of $k$ vertices.
in particular \[ \text{OPT}_{\text{Max 3-SAT}}(F) = \text{OPT}_{\text{Ind Set}}(G) \]

and poly. time approx alg. for Ind. Set that gives
\[ \text{OPT}_{\text{Ind Set}}(G) \geq \alpha \cdot \text{OPT}_{\text{Ind Set}}(G) \]

implies poly. time approx alg. for Max SAT that gives
\[ \text{OPT}_{\text{Max 3-SAT}}(F) \geq \alpha \cdot \text{OPT}_{\text{Max 3-SAT}}(F) \]