SAT

SAT

- $x_1, \ldots, x_n$ Boolean variables.
- Clause is an "OR" of Boolean variables or their negations $x_a \vee \neg x_b \vee \neg x_c \ldots$
- Goal: Given a set of m clauses, compute an assignment that satisfies the largest fraction of clauses.

- Algorithm

1. For each $i \in [n]$, independently set x_i to be $true$ or $false$ with probability ½.

- Fix a clause $C_i = x_{i_1} \vee \neg x_{i_2} \vee \cdots \vee \neg x_{i_k}$.

$$\Pr[C_i \text{ is satisfied}] = 1 - \frac{1}{2^k}$$

SAT

$$\mathrm{E}[\text{number of clauses satisfied}] = \sum_{i\in[m]} \left(1 - \frac{1}{2^{|C_i|}}\right)$$

- k-SAT: each clause has exactly k variables.

$$\mathrm{E}[\text{number of clauses satisfied}] = \sum_{i\in[m]} \left(1 - \frac{1}{2^{|C_i|}}\right) = m\left(1 - \frac{1}{2^k}\right)$$

- Since $OPT \leq m$, this gives a $\left(1 - \frac{1}{2^k}\right)$-approximation for k-SAT.
- In general, $|C_i| \geq 1$. Therefore,

$$\mathrm{E}[\text{number of clauses satisfied}] = \sum_{i\in[m]} \left(1 - \frac{1}{2^{|C_i|}}\right) \geq \sum_{i\in[m]} \left(1 - \frac{1}{2}\right) = \frac{m}{2}.$$

- Therefore ½ approximation for SAT.

Integer Program

- Introduce variable y_i to indicate the value of x_i, i.e. $y_i = \begin{cases} 1 \text{ if } x_i \text{ is true} \\ 0 \text{ if } x_i \text{ is false} \end{cases}$.
- Introduce variable z_j to indicate whether clause C_j is satisfied.
- For a clause C_j, let P_j be the set of indices of variables that occur positively and let N_j be the set of indices of variables that occur negated. $C_j = \bigvee_{i \in P_j} x_i \bigvee_{i \in N_j} \neg x_i$. Add constraint

$$z_j \leq \sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i)$$

$$\max \sum_{j \in [m]} z_j$$

subject to

$$z_j \leq \sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \ \forall j \in [m]$$

$$z_j \in \{0,1\} \ \forall j \in [m]$$

$$y_i \in \{0,1\} \ \forall i \in [n]$$

Randomized Rounding

$$\max \sum_{j\in[m]} z_j$$

subject to

$$z_j \le \sum_{i\in P_j} y_i + \sum_{i\in N_j} (1-y_i)\ \forall j \in [m]$$

$$0 \le z_j \le 1\ \forall j \in [m]$$

$$0 \le y_i \le 1\ \forall i \in [n]$$

- Algorithm-2

1. For each i, independently set x_i to $true$ with probability y_i and $false$ with probability $1-y_i$.

$$C_j = \bigvee_{i\in P_j} x_i \bigvee_{i\in N_j} \neg x_i$$

- Fix a clause C_j and let $k = |C_j|$.

$$\Pr[C_j \text{ is not satisfied}] = \Pi_{i\in P_j}(1-y_i)\Pi_{i\in N_j}(1-(1-y_i))$$

AM≥GM $$\le \left(\frac{1}{k}\left(\sum_{i\in P_j}(1-y_i) + \sum_{i\in N_j}(1-(1-y_i))\right)\right)^k = \left(1-\frac{1}{k}\left(\sum_{i\in P_j} y_i + \sum_{i\in N_j}(1-y_i)\right)\right)^k \le \left(1-\frac{z_j}{k}\right)^k$$

- Therefore, $\Pr[C_j \text{ is satisfied}] \ge 1-\left(1-\frac{z_j}{k}\right)^k$.

- Lemma: $\forall x \in [0,1], 1 - \left(1 - \frac{x}{k}\right)^k \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) x$
- Lemma implies $\Pr[C_j \text{ is satisfied}] \geq 1 - \left(1 - \frac{z_j}{k}\right)^k \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) z_j \geq \left(1 - \frac{1}{e}\right) z_j$

$$\mathrm{E}[\text{number of clauses satisfied}] = \sum_{j \in [m]} \Pr[C_j \text{ is satisfied}] \geq \left(1 - \frac{1}{e}\right) \sum_{j \in [m]} z_j$$

$$= \left(1 - \frac{1}{e}\right) LP$$

- Therefore, Algorithm-2 is a $\left(1 - \frac{1}{e}\right)$-approximation algorithm.

Lemma: $\forall x \in [0,1], 1-\left(1-\frac{x}{k}\right)^k \geq \left(1-\left(1-\frac{1}{k}\right)^k\right)x$

1. The function $f(x) = 1-\left(1-\frac{x}{k}\right)^k$ is concave in $[0,1]$.

   $f'(x) = \left(1-\frac{x}{k}\right)^{k-1}$ and $f''(x) = -\frac{(k-1)}{k}\left(1-\frac{x}{k}\right)^{k-2}$. Therefore, $f''(x) \leq 0\ \forall x \in [0,1]$.

2. $f(x)$ does not lie below the line segment joining $f(0)$ and $f(1)$ for $x \in [0,1]$.

$$f(x) \geq x\big(f(1)-f(0)\big)+f(0) \geq x\left(1-\left(1-\frac{1}{k}\right)^k\right)$$

- For clause C_j

1. $\Pr[\text{Alg1 satisfies } C_j] = \left(1 - \frac{1}{2^k}\right)$
2. $\Pr[\text{Alg2 satisfies } C_j] = \left(1 - \left(1 - \frac{1}{k}\right)^k\right) z_j$

- For "large" values of k, Alg1 is better and for "small" values of k, Alg2 is better.

- Algorithm 3

1. Choose a random algorithm from {Alg1, Alg2} and run it on the instance.

$$\Pr[\text{Alg3 satisfies } C_j] = \frac{1}{2}\Pr[\text{Alg1 satisfies } C_j] + \frac{1}{2}\Pr[\text{Alg2 satisfies } C_j]$$

$$= \frac{1}{2}\left(1 - \frac{1}{2^k}\right) + \frac{1}{2}\left(\left(1 - \left(1 - \frac{1}{k}\right)^k\right) z_j\right)$$

$$\geq \left(\frac{1}{2}\left(1 - \frac{1}{2^k}\right) + \frac{1}{2}\left(1 - \left(1 - \frac{1}{k}\right)^k\right)\right) z_j$$

$3/4$-approximation

k	$\left(1-\frac{1}{2^k}\right)$	$\left(1-\left(1-\frac{1}{k}\right)^k\right)$	$\frac{1}{2}\left(1-\frac{1}{2^k}\right)+\frac{1}{2}\left(1-\left(1-\frac{1}{k}\right)^k\right)$
1	$1/2$	1	$3/4$
2	$3/4$	$3/4$	$3/4$
≥ 3	$\geq 7/8$	$\geq 1-1/e$	$\geq 3/4$

$$\Pr[\text{Alg3 satisfies } C_j] \geq \left(\frac{1}{2}\left(1-\frac{1}{2^k}\right)+\frac{1}{2}\left(1-\left(1-\frac{1}{k}\right)^k\right)\right) z_j \geq \frac{3}{4} z_j$$

Max 2SAT- Integer Program

$$\max \sum_{j \in [m]} \text{val}(C_j)$$
subject to
$$y_i^2 = 1 \; \forall i \in [n]$$

- For each x_i, introduce variable $y_i \in \{+1, -1\}$ to "indicate" if x_i is true.
- clause $\neg x_a \vee x_b$ is not satisfied if $\frac{1}{4}(1 + y_a)(1 - y_b) = 1$.
- Therefore, $\neg x_a \vee x_b$ is satisfied if $1 - \frac{1}{4}(1 + y_a)(1 - y_b) = 1$.

$$\max \sum_{j \in [m]} \text{val}(C_j)$$
subject to
$$\|v_i\|^2 = 1 \; \forall i \in [n]$$
$$\|v_0\|^2 = 1$$

- Introduce "one" vector v_0.
- For clause $\neg x_a \vee x_b$

$$\text{val}(C_j) = 1 - \frac{1}{4}\langle v_0 + v_a, v_0 - v_b \rangle$$
$$= 1 - \frac{1}{4}(\langle v_0, v_0 \rangle + \langle v_0, v_a \rangle - \langle v_0, v_b \rangle - \langle v_a, v_b \rangle)$$
$$= \frac{1}{4}(1 - \langle v_0, v_a \rangle) + \frac{1}{4}(1 + \langle v_0, v_b \rangle) + \frac{1}{4}(1 + \langle v_a, v_b \rangle)$$

GW algorithm

1. Sample $g \sim \mathcal{N}(0,1)^{n+1}$ and let $S_g = \{i \in [n]: \langle v_i, g\rangle \text{ has same sign as } \langle v_0, g\rangle\}$.
2. Set all variables in S_g to "True" and variables in $[n] \setminus S_g$ to "False".

- $(1 - \langle v_0, v_a\rangle)$ gets rounded to 2 with probability θ_{0a}/π and $1 - \langle v_0, v_a\rangle = 1 - \cos\theta_{0a}$
- $(1 + \langle v_0, v_b\rangle)$ gets rounded to 2 with probability $(\pi - \theta_{0b})/\pi$ and
$$1 + \langle v_0, v_b\rangle = 1 + \cos\theta_{0b} = 1 - \cos(\pi - \theta_{0b})$$
- Therefore, α_{GW} (≈ 0.878) approximation to Max 2SAT (H.W.).

- $\frac{7}{8}$-approximation for Max-3SAT.
- NP-hard to do better than this [Hastad - 2001].