CSE 632: Analysis of Algorithms II: Randomized Algorithms **Spring 2024**

Lecture 9: MAX-SAT

Lecturer: Zongchen Chen

1 MAX-SAT Problem

Consider the MAX-SAT problem (maximum satisfiability problem): Given a Boolean formula F in CNF (conjunctive normal form) with n variables $x_1, \dots, x_n$ and m clauses $c_1, \dots, c_m$, find a truth assignment to the variables that maximizes the number of satisfied clauses.

Example 1. Consider a CNF formula

$$F = (x_1 \vee \neg x_3 \vee x_4) \wedge (x_2 \vee x_3) \wedge (\neg x_2) \wedge (\neg x_1 \vee x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_4).$$

The truth assignment $x_1 = \mathsf{F}$, $x_2 = \mathsf{F}$, $x_3 = \mathsf{T}$, $x_4 = \mathsf{F}$ satisfies four clauses. It can be checked that no assignment satisfies all five clauses.

MAX-SAT is clearly NP-hard, and so we aim for (randomized) approximation algorithms: Output an assignment such that the number of satisfied clauses is at least $\alpha \cdot \mathsf{OPT}$ (in expectation), where $\alpha \in [0, 1]$ is the *approximation ratio* and OPT is the optimal value (i.e., the maximum number of satisfied clauses of an assignment).

2 Simple Approximation Algorithm

Lemma 2. *For a CNF formula F with m clauses, there exists an assignment satisfying at least $m/2$ clauses. In fact, a random assignment satisfies at least $m/2$ clauses in expectation.*

Proof. Consider a uniformly random truth assignment x, where for each i independently we set $x_i = \mathsf{T}$ with probability $1/2$ and $x_i = \mathsf{F}$ with probability $1/2$. For each clause c_j, let Y_j be an indicator random variable such that $Y_j = 1$ if c_j is satisfied by x, and $Y_j = 0$ otherwise. Let $Y = \sum_{j=1}^m Y_j$ be the number of satisfied clauses. Notice that for every j, suppose the size of c_j is $k_j \geq 1$, and we have

$$\mathbb{E}Y_j = \Pr(c_j \text{ is satisfied}) = 1 - \frac{1}{2^{k_j}} \geq 1 - \frac{1}{2} = \frac{1}{2}.$$

Therefore, by the linearity of expectation we have

$$\mathbb{E}Y = \sum_{j=1}^{m} \mathbb{E}Y_j \geq \frac{m}{2}, \tag{1}$$

as claimed. □

Lemma 2 gives a simple randomized approximation algorithm (i.e., outputting a random assignment) such that the number of satisfied clauses is at least $\frac{1}{2}\mathsf{OPT}$ in expectation, since $\mathsf{OPT} \leq m$.

Derandomization via the method of conditional probabilities Note that the guarantee of the simple randomized approximation algorithm is to output an assignment satisfying at least $m/2$ clauses *in expectation*. Meanwhile, we hope to obtain an assignment satisfying $m/2$ clauses with high probability (say, with probability $1 - \delta$ for any small $\delta > 0$). We will actually present a *deterministic* algorithm such that the number of satisfying clauses is at least $m/2$ always (i.e., with probability 1). This is achieved by the *method of conditional probabilities*.

By the law of total expectation, we have

$$\begin{aligned}\mathbb{E}Y &= \Pr(x_1 = \mathsf{T})\, \mathbb{E}[Y|x_1 = \mathsf{T}] + \Pr(x_1 = \mathsf{F})\, \mathbb{E}[Y|x_1 = \mathsf{F}] \\ &= \frac{1}{2} \left(\mathbb{E}[Y|x_1 = \mathsf{T}] + \mathbb{E}[Y|x_1 = \mathsf{F}] \right).\end{aligned}$$

The idea is as follows. Since $\mathbb{E}Y \geq m/2$, we know that either $\mathbb{E}[Y|x_1 = \mathsf{T}] \geq m/2$ or $\mathbb{E}[Y|x_1 = \mathsf{F}] \geq m/2$. In the former case, we should set $x_1 = \mathsf{T}$, and in the latter $x_1 = \mathsf{F}$. A key observation is that we can exactly compute the two conditional expectations $\mathbb{E}[Y|x_1 = \mathsf{T}]$ and $\mathbb{E}[Y|x_1 = \mathsf{F}]$ in linear time by the linearity of expectation, and hence we can determine the way to set x_1 since the larger one is at least $m/2$.

Algorithm 1 Derandomization of the simple algorithm for MAX-SAT

```
while ∃ variable x whose truth value is not assigned do
    Compute E[Y|x = T] and E[Y|x = F] where Y = number of satisfied clauses of a random assignment;
    if E[Y|x = T] ≥ E[Y|x = F] then
        x ← T
    else                                                    ▷ E[Y|x = T] < E[Y|x = F]
        x ← F
    end if
    Simplify F (remove x, satisfied clauses, and empty clauses)
end while
```

Algorithm 1 is deterministic and outputs an assignment satisfying at least $m/2$ clauses.

3 LP-Based Approximation Algorithm

IP for MAX-SAT We represent the MAX-SAT problem as an equivalent 0/1 IP in the following way. We use 1 to represent T and 0 for F; hence for each $i \in [n]$, $x_i = 1$ if x_i is assigned T, and $x_i = 0$ otherwise. For each $j \in [m]$, let z_j be the indicator variable for whether the clause c_j is satisfied by the assignment or not; namely, $z_j = 1$ if c_j is satisfied and $z_j = 0$ otherwise. The objective function is clearly $\sum_{j=1}^{m} z_j$, the number of satisfied clauses, which we want to maximize. For each clause c_j, we add a corresponding linear constraint in the following way. Let P_j be the set of those variables that appear in positive form in c_j, and N_j be the set of those in negative form. Then we add the constraint

$$\sum_{i \in P_j} x_i + \sum_{i \in N_j} (1 - x_i) \geq z_j.$$

Example 3. Suppose $c_4 = (x_3 \vee \neg x_5 \vee x_7 \vee \neg x_8)$. Then $P_j = \{3, 7\}$ and $N_j = \{5, 8\}$, and we add the constraint $x_3 + (1 - x_5) + x_7 + (1 - x_8) \geq z_4$. Observe that c_4 is unsatisfied if and only if $x_3 = 1 - x_5 = x_7 = 1 - x_8 = 0$, in which case z_4 is forced to be 0.

We obtain an equivalent 0/1 IP for MAX-SAT.

$$\begin{aligned} \max \quad & \sum_{j=1}^{m} z_j && \text{(IP for MAX-SAT)} \\ \text{subject to} \quad & \sum_{i \in P_j} x_i + \sum_{i \in N_j} (1 - x_i) \geq z_j, \quad \forall j \in [m] \\ & x_i \in \{0, 1\}, \quad \forall i \in [n] \\ & z_j \in \{0, 1\}, \quad \forall j \in [m] \end{aligned}$$

LP relaxation We replace the integral constraints $x_i, z_j \in \{0,1\}$ by $0 \le x_i, z_j \le 1$, and solve the resulting LP (in polynomial time). Let (x^*, z^*) denote the optimal IP solution which is integral and what we want to approximate. Let $(\hat{x}^*, \hat{z}^*)$ denote the optimal LP solution which is fractional and what we have.

Fact 4. *The IP optimum is at most the LP optimum. More precisely,*

$$\mathsf{OPT} = \sum_{j=1}^{m} z_j^* \le \sum_{j=1}^{m} \hat{z}_j^*.$$

Randomized rounding We round the optimal LP solution $\hat{x}^*$ to a valid integral solution by the following randomized procedure.

Algorithm 2 Randomized rounding

Input: $\hat{x}^*$ optimal LP solution
  for each $i \in [n]$ independently **do**
    $x_i \leftarrow 1$ with probability $\hat{x}_i^*$, and $x_i \leftarrow 0$ with probability $1 - \hat{x}_i^*$ ▷ x_i is a Bernoulli random variable with mean $\hat{x}_i^*$
  end for
    return x

Lemma 5. *For every clause c_j of size $k = k_j$, we have*

$$\Pr\left(c_j \textit{ is satisfied}\right) \ge \beta_k \hat{z}_j^*$$

where $\beta_k = 1 - (1 - \frac{1}{k})^k \ge 1 - \frac{1}{e}$.

Given Lemma 5, we are able to analyze the approximation ratio of our LP-based approximation algorithm. For each $j \in [m]$, let Y_j be an indicator random variable for the event that clause c_j is satisfied by the random assignment x; namely, $Y_j = 1$ if c_j is satisfied and $Y_j = 0$ if not. Let $Y = \sum_{j=1}^{m} Y_j$ be the number of satisfied clauses. Then we have

$$\begin{aligned}
\mathbb{E}Y = \sum_{j=1}^{m} \mathbb{E}Y_j &= \sum_{j=1}^{m} \Pr\left(c_j \text{ is satisfied}\right) \\
&\ge \sum_{j=1}^{m} \beta_{k_j} \hat{z}_j^* && \text{(Lemma 5)} \\
&\ge \left(1 - \frac{1}{e}\right) \sum_{j=1}^{m} \hat{z}_j^* \\
&\ge \left(1 - \frac{1}{e}\right) \sum_{j=1}^{m} z_j^* && \text{(Fact 4)} \\
&= \left(1 - \frac{1}{e}\right) \mathsf{OPT}.
\end{aligned}$$

Therefore, the LP-based randomized approximation algorithm finds a truth assignment that satisfies at least $\left(1 - \frac{1}{e}\right) \mathsf{OPT}$ clauses in expectation. This algorithm can also be derandomized using the method of conditional probabilities.

It remains to show Lemma 5.

Proof of Lemma 5. Without loss of generality, we assume that the k variables in c_j are $x_1, \ldots, x_k$ and they are all in positive form; namely, $c_j = (x_1 \vee \cdots \vee x_k)$. The corresponding LP constraint gives $\sum_{i=1}^{k} \hat{x}_i^* \geq \hat{z}_j^*$. It follows that

$$\begin{aligned}
\Pr\left(c_j \text{ is } \textit{unsatisfied}\right) &= \Pr\left(x_1 = \cdots = x_k = 0\right) \\
&= \prod_{i=1}^{k} (1 - \hat{x}_i^*) \\
&\leq \left(\frac{1}{k} \sum_{i=1}^{k} (1 - \hat{x}_i^*) \right)^k && \text{(AM–GM Inequality)} \\
&= \left(1 - \frac{1}{k} \sum_{i=1}^{k} \hat{x}_i^* \right)^k \\
&\leq \left(1 - \frac{1}{k} \hat{z}_j^* \right)^k && \text{(LP Constraint)} \\
&\leq 1 - \beta_k \hat{z}_j^*,
\end{aligned}$$

where the last inequality follows from that $(1 - \frac{t}{k})^k \leq 1 - \beta_k t$ for all $t \in [0, 1]$. The lemma then follows. □

4 Better-of-Two Algorithm

In Section 2, we showed a simple approximation algorithm which finds a truth assignment such that the number of satisfied clauses is at least $\frac{1}{2}\mathsf{OPT}$. In Section 3, we presented an LP-based algorithm such that the number of satisfied clauses is at least $\left(1 - \frac{1}{e}\right)\mathsf{OPT}$. In this section, we show that by simply combining the two algorithms and choosing a better solution leads to a better approximation ratio of $\frac{3}{4}$.

Algorithm 3 Better-of-two algorithm

Input: F a CNF formula
  $x^{(1)} \leftarrow$ solution from simple algorithm
  $x^{(2)} \leftarrow$ solution from LP-based algorithm
    return the better of $x^{(1)}$ and $x^{(2)}$

Theorem 6. *The better-of-two algorithm outputs a truth assignment satisfying at least $\frac{3}{4}\mathsf{OPT}$ clauses in expectation.*

Proof. Let $Y^{(1)} = \sum_{j=1}^{m} Y_j^{(1)}$ be the number of satisfied clauses for the solution $x^{(1)}$ from the simple algorithm, where $Y_j^{(1)}$ is the indicator random variable for whether clause c_j is satisfied. Similarly, $Y^{(2)} = \sum_{j=1}^{m} Y_j^{(2)}$ be the number of satisfied clauses for the solution $x^{(2)}$ from the LP-based algorithm, where $Y_j^{(2)}$ is the indicator random variable for clause c_j. Let $Y = \max\{Y^{(1)}, Y^{(2)}\}$ be the number of satisfied clauses for the better-of-two algorithm. From our analysis in Sections 2 and 3, we have

$$\begin{aligned}
\mathbb{E}\left[Y^{(1)}\right] &= \sum_{j=1}^{m} \mathbb{E}[Y_j^{(1)}] = \sum_{j=1}^{m} \left(1 - \frac{1}{2^{k_j}}\right) \geq \sum_{j=1}^{m} \left(1 - \frac{1}{2^{k_j}}\right) \hat{z}_j^*, \\
\text{and} \quad \mathbb{E}\left[Y^{(2)}\right] &= \sum_{j=1}^{m} \mathbb{E}[Y_j^{(2)}] \geq \sum_{j=1}^{m} \beta_{k_j} \hat{z}_j^* = \sum_{j=1}^{m} \left(1 - \left(1 - \frac{1}{k_j}\right)^{k_j}\right) \hat{z}_j^*.
\end{aligned}$$

Therefore, we deduce that

$$\begin{aligned}
\mathbb{E}Y &= \mathbb{E}\left[\max\left\{Y^{(1)}, Y^{(2)}\right\}\right] \\
&\geq \mathbb{E}\left[\frac{1}{2}\left(Y^{(1)} + Y^{(2)}\right)\right] \\
&= \frac{1}{2}\left(\mathbb{E}\left[Y^{(1)}\right] + \mathbb{E}\left[Y^{(2)}\right]\right) \\
&= \sum_{j=1}^{m} \frac{1}{2}\left[\left(1 - \frac{1}{2^{k_j}}\right) + \left(1 - \left(1 - \frac{1}{k_j}\right)^{k_j}\right)\right] \hat{z}_j^* \\
&\overset{\text{(i)}}{\geq} \sum_{j=1}^{m} \frac{3}{4} \hat{z}_j^* \\
&\geq \frac{3}{4} \sum_{j=1}^{m} z_j^* \\
&= \frac{3}{4}\mathsf{OPT},
\end{aligned}$$

where (i) follows from

$$\left(1 - \frac{1}{2^k}\right) + \left(1 - \left(1 - \frac{1}{k}\right)^k\right) \geq \frac{3}{2}$$

for all $k \in \mathbb{N}^+$, with equality when $k = 1$ or 2. □