Saarland
Informatics Campus

Script of Lecture 5, Approximation Algorithms
Summer term 2017

Tobias Mömke, Hang Zhou
http://www-cc.cs.uni-saarland.de/course/61/

Written by Hang Zhou

1 Maximum Satisfiability Problem (MAX SAT)

Input: n Boolean variables $x_1, \ldots, x_n$, each having value either true or false; m clauses $C_1, \ldots, C_m$, each being the disjunction of some number of the variables and their negations (e. g., $x_3 \vee \bar{x_5} \vee x_{11}$); and a nonnegative weight w_j for each clause C_j.

Output: an assignment of true/false to the Boolean variables that maximizes the total weight of the satisfied clauses.

Terminology:

- literal: x_i, $\bar{x}_i$
- positive literal: x_i
- negative literal: $\bar{x}_i$
- length of a clause: number of literals in that clause; let l_j denote the length of clause C_j

2 Simple Algorithm by Randomized Sampling

Randomized sampling algorithm: Set each x_i to true with probability 1/2, independently.

Theorem 1. *Randomized sampling gives a* (1/2)*-approximation algorithm for MAX SAT.*

Proof. Consider a random variable Y_j such that $Y_j = 1$ if clause C_j is satisfied and 0 otherwise. Let $W := \sum_{j=1}^m w_j Y_j$ be the total weight of the satisfied clauses. By the linearity of expectation, we have:

$$E[W] = \sum_{j=1}^{m} w_j \cdot E[Y_j] = \sum_{j=1}^{m} w_j \cdot \mathbb{P}[\text{clause } C_j \text{ satisfied}],$$

where

$$\mathbb{P}[\text{clause } C_j \text{ satisfied}] = 1 - \left(\frac{1}{2}\right)^{l_j} \geq \frac{1}{2}, \qquad \text{since } l_j \geq 1.$$

Therefore, we have

$$E[W] \geq \frac{1}{2} \sum_{j=1}^{m} w_j \geq \frac{1}{2}\text{OPT},$$

where OPT is the optimum value for MAX SAT. □

3 Derandomization

To achieve a deterministic algorithm, we set the values of the variables one by one.

How to set the value of x_1 in order to preserve the expected value of W? Consider the expected value under the two settings $x_1 = true$ and $x_1 = false$. We have

$$E[W] = E[W \mid x_1 \leftarrow true] \cdot \mathbb{P}[x_1 \leftarrow true] + E[W \mid x_1 \leftarrow false] \cdot \mathbb{P}[x_1 \leftarrow false].$$

Since $\mathbb{P}[x_1 \leftarrow true] = \mathbb{P}[x_1 \leftarrow false] = 1/2$, one of $E[W \mid x_1 \leftarrow true]$ and $E[W \mid x_1 \leftarrow false]$ is at least $E[W]$. We set x_1 to true if

$$E[W \mid x_1 \leftarrow true] \geq E[W \mid x_1 \leftarrow false],$$

and otherwise we set x_1 to false. The expected value under this setting is at least $E[W]$.

We extend the above argument to the general case, where we have set variables $x_1, \ldots, x_i$ to values $b_1, \ldots, b_i$ respectively. We set x_{i+1} to true if

$$E[W \mid x_1 \leftarrow b_1, \ldots, x_i \leftarrow b_i, x_{i+1} \leftarrow true] \geq E[W \mid x_1 \leftarrow b_1, \ldots, x_i \leftarrow b_i, x_{i+1} \leftarrow false],$$

and otherwise we set x_{i+1} to false. The expected value under this setting is at least $E[W \mid x_1 \leftarrow b_1, \ldots, x_i \leftarrow b_i]$.

It only remains to show how to compute the conditional expectations.

$$\begin{aligned} E[W \mid x_1 \leftarrow b_1, \ldots, x_i \leftarrow b_i] &= \sum_{j=1}^{m} w_j \cdot E[Y_j \mid x_1 \leftarrow b_1, \ldots, x_i \leftarrow b_i] \\ &= \sum_{j=1}^{m} w_j \cdot \mathbb{P}[\text{clause } C_j \text{ satisfied} \mid x_1 \leftarrow b_1, \ldots, x_i \leftarrow b_i]. \end{aligned}$$

The probability that a clause C_j is satisfied given that $x_1 \leftarrow b_1, \ldots, x_i \leftarrow b_i$ is 1 if the settings of $x_1, \ldots, x_i$ already satisfy the clause, and is $1 - (1/2)^k$ otherwise, where k is the number of literals in the clause that remain unset.

Examples:

$$\mathbb{P}[\text{clause } x_3 \vee \bar{x_5} \vee \bar{x_7} \text{ is satisfied} \mid x_1 \leftarrow true, x_2 \leftarrow false, x_3 \leftarrow true] = 1$$

$$\mathbb{P}[\text{clause } x_3 \vee \bar{x_5} \vee \bar{x_7} \text{ is satisfied} \mid x_1 \leftarrow true, x_2 \leftarrow false, x_3 \leftarrow false] = 1 - \left(\frac{1}{2}\right)^2 = \frac{3}{4}$$

4 Randomized Rounding

a) Formulate the problem as an integer program: for every x_i, create a variable $y_i \in \{0, 1\}$ such that $y_i = 1$ corresponds to $x_i \leftarrow true$.

b) Relax the integer program to an linear program: replace the constraints $y_i \in \{0, 1\}$ by $0 \leq y_i \leq 1$.

c) Solve the linear program in polynomial time and obtain an optimal (fractional) solution y^*;

d) Set each x_i to true with probability y_i^*.

4.1 Integer Program

In additional to the variables y_i, we introduce a variable z_j for each clause C_j such that $z_j = 1$ if the clause is satisfied and $z_j = 0$ otherwise. For each clause C_j, let P_j be the indices of the variables that occur positively in the clause, and let N_j be the indices of the variables that occur negatively in the clause. Thus

$$C_j = \bigvee_{i \in P_j} x_i \vee \bigvee_{i \in N_j} \bar{x}_i.$$

We have

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

Integer program:

$$\begin{aligned}
\text{maximize} & \sum_{j=1}^{m} w_j z_j \\
\text{subject to} & \sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j, && \forall C_j = \bigvee_{i \in P_j} x_i \vee \bigvee_{i \in N_j} \bar{x}_i, \\
& y_i \in \{0, 1\}, && i = 1, \ldots, n, \\
& z_j \in \{0, 1\}, && j = 1, \ldots, m.
\end{aligned}$$

4.2 Liner Program Relaxation

$$\begin{aligned}
\text{maximize} & \sum_{j=1}^{m} w_j z_j \\
\text{subject to} & \sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j, && \forall C_j = \bigvee_{i \in P_j} x_i \vee \bigvee_{i \in N_j} \bar{x}_i, \\
& 0 \leq y_i \leq 1, && i = 1, \ldots, n, \\
& 0 \leq z_j \leq 1, && j = 1, \ldots, m.
\end{aligned}$$

4.3 Analysis for the Randomized Rounding

Let (y^*, z^*) be an optimal solution to the linear program relaxation. We consider the result of the randomized rounding, by setting each x_i to true with probability y_i^* independently.

Theorem 2. *Randomized rounding gives a* $(1 - \frac{1}{e})$*-approximation algorithm for MAX SAT.*

Proof. The key is to analyze the probability that a given clause C_j is satisfied.

$$\begin{aligned}
\mathbb{P}[\text{clause } C_j \text{ not satisfied}] &= \prod_{i \in P_j} (1 - y_i^*) \prod_{i \in N_j} y_i^* \\
&\leq \left[\frac{1}{l_j} \left(\sum_{i \in P_j} (1 - y_i^*) + \sum_{i \in N_j} y_i^* \right) \right]^{l_j} \\
&\qquad \text{(geometric mean} \leq \text{arithmetic mean)} \\
&= \left[1 - \frac{1}{l_j} \left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*) \right) \right]^{l_j} \\
&\leq \left(1 - \frac{z_j^*}{l_j} \right)^{l_j}, \qquad \text{using linear program constraints.}
\end{aligned}$$

Fact 1. *If a function $f(x)$ is concave on the interval $[0,1]$ (that is, $f''(x) \le 0$), and $f(0) = a$ and $f(1) = b + a$, then $f(x) \ge bx + a$ for $x \in [0,1]$.*

The function $f(z_j^*) := 1 - \left(1 - \frac{z_j^*}{l_j}\right)^{l_j}$ is concave. So from Fact 1, we have:

$$\begin{aligned}\mathbb{P}[\text{clause } C_j \text{ satisfied}] &\ge 1 - \left(1 - \frac{z_j^*}{l_j}\right)^{l_j} \\ &\ge \left[1 - \left(1 - \frac{1}{l_j}\right)^{l_j}\right] z_j^*.\end{aligned}$$

The expected value of the solution from the randomized rounding algorithm is

$$\begin{aligned}E[W] &= \sum_{j=1}^{m} w_j \cdot \mathbb{P}[\text{clause } C_j \text{ satisfied}] \\ &\ge \sum_{j=1}^{m} w_j \cdot \left[1 - \left(1 - \frac{1}{l_j}\right)^{l_j}\right] z_j^* \\ &\ge \min_{k \ge 1} \left[1 - \left(1 - \frac{1}{k}\right)^{k}\right] \sum_{j=1}^{m} w_j z_j^*.\end{aligned}$$

Note that the function $1 - \left(1 - \frac{1}{k}\right)^k$ is non-increasing in k and approaches $1 - \frac{1}{e}$ when k tends to infinity. So

$$E[W] \ge \left(1 - \frac{1}{e}\right) \sum_{j=1}^{m} w_j z_j^* \ge \left(1 - \frac{1}{e}\right) \text{OPT}.$$

□

The derandomization of the randomized rounding algorithm is left as an exercise.

5 Choosing the Better of Two Solutions

For a given clause of length k, the randomized sampling algorithm (Section 2) satisfies the clause with probability $1 - \left(\frac{1}{2}\right)^k$ (which is increasing in k), while the randomized rounding algorithm (Section 4) satisfies the clause with probability at least $\left[1 - \left(1 - \frac{1}{k}\right)^k\right] z_j^*$ (which is decreasing in k).

Since the bad cases for the two algorithms are opposite, we can achieve a better approximation guarantee by choosing the better solution of the two algorithms.

Theorem 3. *Choosing the better solution of the randomized sampling algorithm and the randomized rounding algorithm gives a $(3/4)$-approximation algorithm for MAX SAT.*

The proof is left as an exercise.

Since both of the two algorithms can be derandomized, we then obtain a *deterministic* $(3/4)$-approximation algorithm for MAX SAT.