CMPUT 675: Approximation Algorithms	Winter 2005
Lecture 9: Feb 9	
Lecturer: Mohammad R. Salavatipour	*Scribe: Junfeng Wu*

Today we will continue studying Max-SAT. The main result presented today is a $\frac{3}{4}$-approximation for Max-SAT.

9.1 MAX-SAT (Continued)

Recall the Max-SAT problem defined in the last lecture:

Definition 9.1 Max-SAT: *Given a boolean formula ϕ in CNF over boolean variables $x_1, ..., x_n$ plus non-negative weights w_j for each clause C_j, $1 \leq j \leq m$, find a truth assignment to x_i's, that maximizes the total weight of satisfied clauses.*

Last time, we presented two algorithms. The first one was a simple randomized algorithm (called Alg1) based on flipping fair coins. We saw that this is a $\frac{1}{2}$-approximation. Also, if clauses are all large, say at least k, then this algorithm is a $(1 - \frac{1}{2^k})$-approximation. Algorithm 2, which was based on flipping a baised coin, was a p-approximation algorithm, where $p \simeq 0.618$. Now we present a different algorithm that works well if clauses are small. This algorithm is based on an IP/LP formulation of Max-SAT and LP-rounding. First we show how to formulate the problem as an IP/LP.

Let P_j (N_j) be the indices of variables in clauses C_j that are in positive (negative) form. For every x_i, we have an indicating variable y_i which is set to 1 (0) iff x_i is set to Ture (False). Also, for every clause C_j, we have a variable z_j which is 1 iff C_j is satisfied. Then the Max-SAT problem can be stated as:

$$\begin{array}{ll} \text{maximize} & \sum w_j z_j \\ \text{subject to} & \forall j : \sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j \\ & \forall j : z_j \in \{0, 1\} \\ & \forall i : y_i \in \{0, 1\} \end{array}$$

The LP-relaxation is:

$$\begin{array}{ll} \text{maximize} & \sum w_j z_j \\ \text{subject to} & \forall j : \sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j \\ & \forall j : 0 \leq z_j \leq 1 \\ & \forall i : 0 \leq y_i \leq 1 \end{array}$$

Alg 3 (Randomized-Rounding):
- Solve the LP; let (y^*, z^*) be the optimal solution
- For each x_i, set it to True with probability y_i^*
- Let $\widehat{x}$ (vector) be the integer solution obtained.

Theorem 9.2 *(Gocmans&Williamson '94): Randomized-Rounding is a $(1 - \frac{1}{e})$-approximation algorithm for MAX-SAT.*

Proof: We will use the following two facts:

Fact 1 (Arithmetic-Geometric inequality): If $a_1, ..., a_n$ are numbers then:

$$a_1 + ... + a_n \geq \sqrt[n]{a_1 \cdot a_2 \ldots a_n}$$

Fact 2: If $f(x)$ is a real-valued function and is concave in $[0, 1]$ (i.e. $f''(x) \leq 0$), $f(0) = 0$, and $f(1) = \alpha$ then the function is lower-bounded in $[0, 1]$ by the line that goes through $(0, 0)$ and $(1, \alpha)$.

Let W_j be the contribution of C_j to the total weight of the solution and let W be the total weight of the solution returned by the algorithm.

Lemma 9.3 *For every clause C_j with size k,* $\mathrm{E}[W_j] \geq [1 - (1 - \frac{1}{k})^k] W_j z_j^*$.

Let $\beta_k = [1 - (1 - \frac{1}{k})^k]$. First we show why proving this lemma implies the theorem. Suppose that all clauses have size $\leq k$. Then

$$\begin{aligned}
\mathrm{E}[W] &= \sum_j W_j \cdot \Pr[C_j \text{ is satisfied}] \\
&= \sum_j \mathrm{E}[W_j] \\
&\geq \beta_k \sum_j W_j z_j^* \\
&= \beta_k \cdot OPT_f \\
&\geq \beta_k \cdot OPT
\end{aligned}$$

When k goes to infinity, $(1 - \frac{1}{k})^k$ goes to $\frac{1}{e}$ from below. Therefore $(1 - \frac{1}{k})^k \leq \frac{1}{e}$ and $1 - \beta_k \geq 1 - \frac{1}{e}$ and this completes the proof of theorem ■

Remark: we can turn this algorithm into a deterministic algorithm by using the method of conditional probability.

Proof of Lemma 9.3:

$$\begin{aligned}
\Pr[C_j \text{ is satisfied}] &= 1 - \prod_{i \in P_j} (1 - y_i^*) \prod_{i \in N_j} y_i^* \\
\text{(by Fact 1 above)} &\geq 1 - \left(\frac{\sum_{i \in P_j} (1 - y_i^*) + \sum_{i \in P_j} y_i^*}{k}\right)^k \\
&= 1 - \left(\frac{\sum_{i \in P_j} y_i^* + \sum_{i \in P_j} (1 - y_i^*)}{k}\right)^k \\
\text{(by constraint for } C_j \text{ in the LP)} &\geq 1 - \left(1 - \frac{z_j^*}{k}\right)^k
\end{aligned}$$

Let's define $g(z) = 1 - (1 - \frac{z}{k})^k$, $g(0) = 0$, $g(1) = \beta_k$ and g is concave. So by fact 2: $g(z) \geq \beta_k \cdot z$. Therefore $\Pr[C_j \text{ is satisfied}] \geq \beta_k \cdot z_j^*$. From the definition of E[.]:

$$\begin{aligned}
\mathrm{E}[W_j] &= w_j \cdot \Pr[C_j \text{ is satisfied}] \\
&\geq \beta_k w_j z_j^*
\end{aligned}$$

and this completes the proof of lemma.

■

Note that $1 - \frac{1}{e} \approx 0.632$ which is greater than 0.618 in algorithm 2. Also, if all clauses have size at most k and k is relatively small, then the approximation ratio of this algorithm is $1 - (1 - \frac{1}{k})^k > 1 - \frac{1}{e}$. So we get better approximation factor for smaller k's while Alg1 gives better approximation factor for larger k's. So it seems reasonable to run both algorithms and return the better solution. This is the main idea of our 3rd algorithm which gives a $\frac{3}{4}$-approximation ratio.

Suppose we flip a coin and based on the outcome ($a = 0$ or $a = 1$) we run algorithm 1 (simple randomized) or algorithm 3 (randomized-roundinig).

Lemma 9.4 *For each C_j, $\text{E}[W_j] \geq \frac{3}{4} w_j z_j^*$.*

Proof: Let's assume that C_j has k variables and define $\alpha_k = 1 - \frac{1}{2^k}$. From the proof of Theorem 8.2 (in lecture 8):

$$\text{E}[W_j | a = 0] \geq (1 - \frac{1}{2^k}) w_j \geq \alpha_k w_j z_j^*$$

and

$$\text{E}[W_j | a = 1] \geq \beta_k z_j^* w_j.$$

Therefore, combining these two:

$$\text{E}[W_j] = \text{E}[W_j | a = 0] \Pr[a = 0] + \text{E}[W_j | a = 1] \Pr[a = 1] \geq \frac{1}{2}(\alpha_k + \beta_k) w_j z_j^*$$

Since $\alpha_1 + \beta_1 = \frac{1}{2} + 1 = \frac{3}{2}$, $\alpha_2 + \beta_2 = \frac{3}{4} + \frac{3}{4} = \frac{3}{2}$, and for $k \geq 3$: $\alpha_k = 1 - \frac{1}{2^k} \geq 1 - \frac{1}{8} = 0.875$, $\beta_k = 1 - (1 - \frac{1}{k})^k \geq 1 - \frac{1}{e}$; $\alpha_k + \beta_k \geq \frac{3}{2}$ for all values of k. Therefore,

$$\text{E}[W] = \sum_j \text{E}[W_j] \geq \frac{3}{4} \sum_j w_j z_j^* \geq \frac{3}{4} OPT.$$

■

We can easily derandomize this algorithm:

Deterministic $\frac{3}{4}$-approximation Alg3
- use derandomized version of algorithm 1
- use derandomized version of algorithm 3
- return whichever is better

Theorem 9.5 (Goemans&Williamson '94) *This is a $\frac{3}{4}$-approximation algorithm.*

Proof: At least one of $\text{E}[W | a = 0]$ or $\text{E}[W | a = 1]$ must be as large as $\text{E}[W]$ which is $\geq \frac{3}{4} OPT$. ■

The following example shows that the analysis of Algorithm 3 is tight, i.e. the integrality gap of the given LP is at least $\frac{4}{3}$.

Example: Consider the following instance of Max-SAT: $(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_2})$, and assume that all the weights are 1. Clearly the cost of OPT is 3. On the other hand, if we set $y_i = 1/2$ and $z_j = 1$ for every i, j we get a feasible fractional solution with weight 4. Therefore, the integrality gap is at least $\frac{4}{3}$.

The best known approximation factor for MAX-SAT is 0.7846 using semi-difinite programming. Based on a conjecture (by Uri Zwick), which is supported by experimental results, we can get 0.8331-approximation. Recall that the lower bound (from the hardness of MAX-E3SAT) is 7/8, i.e. we cannot get an $(\frac{7}{8} - \epsilon)$-approximation for any $\epsilon > 0$, unless P=NP.