Advanced Algorithms - Lecture 3	November 17, 2005

Lecture 3

Lecturer: Yair Bartal — *Scribe: Anton Bar*

MAX-SAT

Note: LP problem (relaxation of ILP) is solvable in polynomial time. In this course we will not study solving LP, instead we will use it as a black box technique. We solve ILP problem by solving a similar LP problem and rounding the fractions to 0 and 1.

Today we will handle a slightly more difficult problem: MAX-SAT.

Given CNF formula: $f = \bigwedge_{j=1}^{m} C_j$, where C_j is a clause of size k_j. Each clause is k_j iterals u_i of the following form: $C_j = \bigvee_i^{k_j} u_i$, where u_i is one of variables $x_1, ..., x_n$ or its negation. w_j is weight of clause C_j. Without loss of generality we can assume that each variable or its negation appears only once in each clause.

The goal is to find an assignment for all variables $x_1, ..., x_n$, s.t. weight of all satisfied clauses is maximal.

$$C_j = \begin{cases} 1, & if\ C_j\ is\ satisfied \\ 0, & otherwise \end{cases}$$

There are two interesting variations:

1. MAX-K-SAT, $\forall j, size(C_j) \leq k$
2. MAX-EK-SAT, $\forall j, size(C_j) = k$ (Exact K-SAT)

Now we will analyze few randomized algorithms - which are algorithms that flip coin and decide. Its natural to talk about expectation.

Definition 1 *Randomized algorithm A has approximation ratio $\alpha < 1$ if A runs in polynomial time and $\forall I : E[A(I)] \geq \alpha \cdot OPT(I)$, where I is input of A.*

Algorithm I - Johnson

Algorithm: In order to round a solution for LP to a solution for ILP - replace each fraction by randomly picked 0 or 1, so that $Pr[0] = Pr[1] = \frac{1}{2}$.

Theorem 1: *Approximation ratio of algorithm I is 2.*

Proof: Lets analyze clause c_j of size k_j. We want to calculate the expectation $E[c_j]$. Expectation of the solution is $E[W] = E[\sum_j w_j c_j]$, where $W = \sum_{1 \geq j \geq m} w_j c_j$ and $w_j \geq 0$.

$E[W] = E[\sum_j w_j c_j] = \sum_j w_j E[c_j]$

Lets define $\alpha = 1 - (\frac{1}{2})^{k_j}$.

$Pr[c_j] = 1 \cdot Pr[c_j = 1] + 0 \cdot Pr[c_j = 0] = Pr[c_j = 1] = 1 - Pr[c_j = 0] = 1 - (\frac{1}{2})^{k_j} = \alpha_{k_j} \geq \frac{1}{2}$

Notes:

- Probability of a negative clause is $\frac{1}{2}$ in power of size of the clause.
- The larger is the clause, the bigger is the expected value.
- Shorter clauses is the weak point of this algorithm. Examples: when $k = 1 : \alpha = \frac{1}{2}, k = 2 : \alpha = 1 - \frac{1}{4} = \frac{3}{4}$.

Now we replace $E[c_j]$ by α and receive: $\sum_j w_j E[c_j] \geq \frac{1}{2} \sum w_j \geq \frac{1}{2} OPT(I)$, because we can't satisfy more than $OPT(I)$ clauses.

Note: It's possible to use this algorithm in order to build deterministic algorithm. We will learn it later on as part of an exercise.

Algorithm II - Goemans-Williamson

This algorithm is based on ILP, lets define it:

Maximize: $\sum_{1 \leq j \leq m} w_j c_j$. **Subject to:**

- $\forall j : \sum_{i \in P_j} x_i + \sum_{i \in N_j} (1 - x_i) \geq c_j$, where $P_j = \{i | x_i \in c_j\}$ (all indexes of positive variables) and $N_j = \{i | \overline{x_i} \in c_j\}$ (all indexes of negative variables)
- $c_j \in \{0, 1\}$
- $x_i \in \{0, 1\}$

Now lets define the related LP:

Maximize: $\sum_j w_j z_j$. **Subject to:**

- $\forall j : \sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j$
- $\forall j : 0 \leq z_j \leq 1$
- $\forall i : 0 \leq y_i \leq 1$

The algorithm will solve the LP and produce optimal fractional solution OPT_F with variables z_j, where $1 \geq j \geq m$, and y_i, where $1 \geq i \geq m$. We will flip a biased coin that shows value $x_i = 1$ with probability y_i and value $x_i = 0$ with probability $1 - y_i$, and the value of c_j is calculated directly from the assignment.

This technique is called Rounding because we round fractional solution to integer. In this case we used randomization, thus it is also called Randomized Rounding.

Theorem 2: *Approximation ratio of algorithm II* $\geq 1 - \frac{1}{e}$.

Claim: Let c_j be a clause of size k_j. $E[c_j] \geq \beta_{k_j} \cdot z_j$, where $\beta_k = 1 - (1 - \frac{1}{k})^k$. $\lim_{k \to \infty} \beta_k = 1 - \frac{1}{e}$. For example, for $k = 1 : \beta_k = 1$.

Note: we want an algorithm where short clauses will contribute more.

Proof of the theorem using the claim:
$E[w] = \sum_j w_j E[c_j] \geq \sum_j \beta_k w_j z_j \geq (1 - \frac{1}{e}) \sum_j w_j z_j = (1 - \frac{1}{e}) OPT_F \geq (1 - \frac{1}{e}) OPT$.

Proof of the claim:
$E[c_j] = Pr[c_j = 1] = 1 - Pr[c_j = 0]$
$Pr[c_j = 0] = \prod_{i \in P_j} 1 - y_i \prod_{i \in N_j} y_i$

Now we use the Arithmetic-Geometric Means Inequality:
for $a_1, ..., a_k \geq 0$: $\sqrt[k]{a_1 \cdot a_2 \cdot ... \cdot a_k} \leq \frac{a_1 + a_2 + ... + a_k}{k}$).

From the inequality follows:

$$Pr[c_j = 0] = \left(\sqrt[k_j]{\prod_{i \in P_j} (1 - y_i) \prod_{i \in N_j} y_i} \right)^{k_j} \leq \left[\frac{\sum_{i \in P_j} (1 - y_i) + \sum_{i \in N_j} y_i}{k_j} \right]^{k_j} =$$
$$\left[\frac{|P_j| - \sum_{i \in P_j} y_i + |N_j| - \sum_{i \in N_j} (1 - y_i)}{k_j} \right]^{k_j} = \left[1 - \frac{\sum_{i \in P_j} y_i + |N_j| - \sum_{i \in N_j} (1 - y_i)}{k_j} \right]^{k_j} \leq \left(1 - \frac{z_i}{k_j} \right)^{k_j}$$

$E[c_j] \geq 1 - \left(1 - \frac{z_i}{k_j}\right)^{k_j} \geq \left(1 - \left(1 - \frac{1}{k_j}\right)^{k_j}\right) \cdot z_j$

In order to prove that $f(z) = 1 - (1 - \frac{z}{k})^k \geq \beta_k \cdot z$, it is enough to observe that for $0 \leq z \leq 1$, $1 - \frac{1}{e} > \beta_k \cdot z$, where k is natural ≥ 1. ■

Algorithm III - (combination of I and II)

The algorithm flips a fair coin and if it shows 1 - runs algorithm I, otherwise algorithm II.

Theorem 3: *Approximation ratio of algorithm III is* $\frac{3}{4}$

Claim:
$\forall c_j : E[c_j] \geq \frac{3}{4} z_j$

Proof of the theorem using the claim:
$E[w] = \sum_j w_j c_j \geq \frac{3}{4} \sum_j w_j z_j \geq \frac{3}{4} OPT_F(I) \geq \frac{3}{4} OPT(I)$. ■

Proof of the claim:
$E[c_j] = \frac{1}{2}\alpha_{k_j} + \frac{1}{2}\beta_{k_j} z_j \geq \frac{1}{2}(\alpha_{k_j} + \beta_{k_j}) z_j$. This is true because $0 \leq z_j \leq 1$.

$\forall k : (\alpha_{k_j} + \beta_{k_j}) \geq \frac{3}{2}$.

We divide it by 2 and receive the desired approximation ratio of $\frac{3}{4}$:
$\forall k : \frac{(\alpha_{k_j} + \beta_{k_j})}{2} \geq \frac{3}{4}$.

Integrality Gap

The question is whether exists algorithm B, s.t. $\frac{B(I)}{OPT_F(I)} > \frac{3}{4}$, where $OPT_F(I)$ is the optimal fractional solution. We will see that there exists problem instance I, s.t. $\frac{OPT(I)}{OPT_F(I)} \leq \frac{3}{4}$.

Lets analyze the following expression: $f = (x_1 \vee x_2) \wedge (\overline{x_1} \vee x_2) \wedge (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_2})$, where all clauses have equal weights w_j. Obviously, $OPT(I) = 3$ because we can't satisfy all 4 clauses. $OPT_F(I) = 4$ because $y_1 = y_2 = \frac{1}{2} \Rightarrow \forall i : y_1 + y_2 \geq z_i$. The integrality gap therefore is $\frac{OPT(I)}{OPT_F(I)} = \frac{3}{4}$.