Lecture 12: Approximation Algorithms II

In this lecture, we continue our discussion on approximation algorithms. In particular, we focus on techniques to define an appropriate *relaxed problem* that can be solved in polynomial time and then to convert the solution to an appropriate integral one.

1 Maximum Satisfiability

We return to the setting of boolean formulas and consider a problem related to satisfiability: for a given formula in Conjunctive Normal Form (CNF), what is the maximum number of its clauses that can be satisfied by assigning true and false value to variables? More concretely, suppose we have n variables $x_1, ..., x_n$ and m clauses $C_1, ..., C_m$ where

$$C_i = \left(\bigvee_{i \in S_i^+} x_i \right) \vee \left(\bigvee_{i \in S_i^-} \bar{x}_i \right).$$

The **maximum satisfiability problem** is to find the maximum number of clauses that may be satisfied by an assignment x.

We first propose a simple randomized algorithm to approximate a solution to this problem. Set each x_i independently to be 0 or 1 with probability 1/2. The probability that C_i is satisfied by this assignment is $1 - 2^{|C_i|}$ for every i. If we let Z_i denote the event that clause C_i is satisfied by this random assignment and $Z = \sum_{i=1}^m Z_i$ be the total number of satisfied clauses, we may compute:

$$\mathrm{E}[Z] = \sum_{i=1}^{m} \mathrm{E}[Z_i] = \sum_{i=1}^{m} \left(1 - 2^{|C_i|}\right).$$

In the case that all of our clauses are large, i.e. $|C_i| \geq K$ for each i, then this randomized algorithm has an approximation ratio of $\geq 1 - 2^{-K}$ in expectation:

$$m(1 - 2^{-K}) \leq \mathrm{E}[Z] \leq OPT \leq m.$$

Having a bound on the approximation ratio of the algorithm in expectation is often unsatisfactory. This is because a bound on expected value does not bound the probability that the algorithm returns a good solution. Concentration inequalities can help us estimate such probabilities, but in some cases we may do even better. This algorithm may be **derandomized** using conditional expectation as follows.

Algorithm 1 Derandomized Approximation Algorithm for Maximum Satisfiability

for $i = 1$ to n **do**
  Compute $\frac{1}{2}\mathrm{E}[Z \mid x_i = 1, x_{i-1}, ..., x_1]$ and $\frac{1}{2}\mathrm{E}[Z \mid x_i = 0, x_{i-1}, ..., x_1]$.
  Set $x_i = 1$ if the first expression is larger than the second, set $x_i = 0$ otherwise.
end for
return x

For motivation, we consider the first step of the algorithm. Note that

$$\mathrm{E}[Z] = \frac{1}{2}\mathrm{E}[Z \mid x_1 = 1] + \frac{1}{2}\mathrm{E}[Z \mid x_1 = 0].$$

Both terms on the right hand side may be computed simply by summing over all clauses the probability that C_i is satisfied given the information on x_1. The equation above implies that $\mathrm{E}[Z \mid x_1 = 1] \geq \mathrm{E}[Z]$ or $\mathrm{E}[Z \mid x_1 = 0] \geq \mathrm{E}[Z]$. Thus if we choose the greater expectation in each step of the algorithm, we will deterministically build up an assignment x such that

$$\mathrm{E}[Z|x] \geq \mathrm{E}[Z] \geq m(1 - 2^{-K})$$

where $\mathrm{E}[Z|x]$ is the number of clauses x satisfies.

The approximation ratio of algorithm 1 is good only if the clauses have a large number of variables. We present a different algorithm for dealing with the possibility that some of the clauses may be small. It is based on the now familiar concept of LP relaxation. We write down an integer program for maximum satisfiability.

$$\begin{array}{lll} \textbf{maximize:} & \displaystyle\sum_{i=1}^{m} q_i & \\ \textbf{s.t.} & \displaystyle q_i \leq \sum_{j \in S_i^+} y_j + \sum_{j \in S_i^-} (1 - y_j) & \forall\, i \\ & q_i, y_j \in \{0, 1\} & \forall\, i, j \end{array}$$

The variables q_i correspond to the truth value of each clause C_i, and the variables y_j correspond to the values of each boolean variable x_i. We relax the last condition to be $0 \leq q_i, y_j \leq 1$ in order to get a linear program.

Algorithm 2 An LP Rounding Algorithm for Maximum Satisfiability

Solve the LP given above.
for $j = 1$ to n **do**
  Independently set $x_j = \begin{cases} 1 & \text{: with probability } y_j^* \\ 0 & \text{: with probability } 1 - y_j^* \end{cases}$
end for

In order to analyze algorithm 2 we consider the probability that a particular clause is satisfied; Let u focus on one clause, say C_1 and assume without loss of generality that $C_1 = x_1 \vee ... \vee x_k$. We have $q_1^* = \min\{y_1^* + ... + y_k^*, 1\}$ and by the inequality of arithmetic and geometric means:

$$\begin{aligned} Pr[C_1] &= 1 - \prod_{j=1}^{k} (1 - y_j^*) \\ &\geq 1 - \left(\frac{1}{k} \sum_{j=1}^{k} (1 - y_j^*) \right)^k \\ &\geq 1 - \left(1 - \frac{q_1^*}{k} \right)^k \\ &\geq q_1 \left(1 - \left(1 - \frac{1}{k} \right)^k \right) \\ &\geq q_1 (1 - 1/e) \end{aligned}$$

This last line implies, through the linearity of expectation, that this rounding procedure gives a $(1 - 1/e)$-approximation for maximum satisfiability regardless of the size of the smallest clause. Algorithm 2 may also be derandomized by the method of conditional expectations.

These two derandomized algorithms may be combined to give a factor 3/4 approximation algorithm for maximum satisfiability. We simply run both algorithms on a given problem instance and output the solution with the better value. This procedure itself may be viewed as a derandomization of an algorithm that flips a fair coin to decide which randomized sub-algorithm to run. That algorithm has approximation factor 3/4:

$$\mathrm{E}[Z] = \sum_{i=1}^{m} \mathrm{E}[Z_i] \geq \sum_{i=1}^{m} \frac{1}{2} \left((1 - 2^{-|C_i|}) + \left(1 - \left(1 - \frac{1}{|C_i|} \right)^{|C_i|} \right) \right) \geq (3/4)m.$$