A Randomized Approximation Algorithm for MAX 3-SAT

CS 511

Iowa State University

December 8, 2008

Note. This material is based on Kleinberg & Tardos, *Algorithm Design*, Chapter 13, Section 4, and associated slides.

MAX 3-SAT

Given: A 3-SAT formula $\phi(x_1, \ldots, x_n) = C_1 \wedge \cdots \wedge C_k$

Goal: Find a truth assignment that satisfies as many clauses as possible.

MAX 3-SAT

Example

$$C_1 = (x_1 \vee \bar{x}_2 \vee x_3) \qquad C_2 = (x_2 \vee x_3 \vee \bar{x}_4)$$

$$C_3 = (\bar{x}_1 \vee x_2 \vee \bar{x}_4) \qquad C_4 = (x_1 \vee \bar{x}_3 \vee x_4)$$

Random assignment: $x_1 = 0,\ x_2 = 1,\ x_3 = 0,\ x_4 = 1$
$\Rightarrow$ 3 clauses are satisfied.

Optimal assignment: $x_1 = x_2 = x_3 = x_4 = 1$
$\Rightarrow$ all clauses are satisfied.

MAX 3-SAT

Theorem (MAX 3-SAT is NP-hard)

If MAX 3-SAT can be solved in polynomial time, then so can 3-SAT.

Proof.

- Suppose there is a polynomial-time algorithm A for MAX 3-SAT.
- Let formula φ be an instance of 3-SAT.
- Run A on input φ.
- If the assignment returned by A satisfies all clauses of φ, then return YES; else return NO.

□

A Randomized Approximation Algorithm

```
Approx-Max3SAT(φ)
  for i = 1 to n
     do Flip a fair coin
        if Heads
           then x_i ← 1
           else x_i ← 0
  return x
```

Observation

The running time of *Approx-Max3SAT* is $O(n)$

Theorem

Approx-Max3SAT is 7/8-approximate.

Proof.

- Define the random variable

$$Z_j = \begin{cases} 1 & \text{if clause } C_j \text{ is satisfied} \\ 0 & \text{otherwise.} \end{cases}$$

- Then, $Z = \sum_{j=1}^{k} Z_j$ is the number of clauses satisfied.
- The expected number of clauses satisfied is

$$E[Z] \underset{\text{linearity}}{=} \sum_{j=1}^{k} E[Z_j] \underset{Z_j \text{ is } 0/1}{=} \sum_{j=1}^{k} \Pr(C_j \text{ is satisfied}) = \frac{7}{8}k$$

□

A Randomized Approximation Algorithm: Implications

Corollary (Lower Bound on Number of Satisfiable Clauses)

For any instance of 3-SAT, there exists a truth assignment that satisfies at least 7/8 of the clauses.

Proof.

A random variable is at least its expectation some of the time. □

Corollary

Any instance of 3-SAT with at most 7 clauses is satisfiable.

Proof.

Follows from the lower bound on number of satisfiable clauses □

The Probabilistic Method

- The lower bound on number of satisfiable clauses is an example of the probabilistic method.
- We showed the existence of a non-obvious property of 3-SAT by showing that a random construction produces it with positive probability.

An Alternative Randomized Algorithm

Johnson's Algorithm

Repeatedly generate random truth assignments until one of them satisfies at least $7k/8$ clauses.

Theorem

Johnson's algorithm is a $7/8$-approximation algorithm that runs in expected polynomial time.

Lemma (Waiting Time Bound)

Consider a series of independent trials where each trial succeeds with probability p and fails with probability $1 - p$. Then, the expected number of trials until the first success is $1/p$.

Proof.

- Let N be the number of trials until first success.
- The probability that j trials are needed is

$$\Pr(N = j) = p(1 - p)^{j-1}$$

- The expected number of trials until first success is

$$E[N] \;=\; \sum_{j=1}^{\infty} j \cdot \Pr(N = j) \;=\; \sum_{k=1}^{\infty} j \cdot p(1 - p)^{j-1} \;=\; \frac{1}{p}$$

□

Lemma (Satisfaction Probability)

The probability that a random assignment satisfies at least $7k/8$ clauses is at least $1/(8k)$.

Proof.

Let p_j be probability that exactly j clauses are satisfied; let p be probability that $\geq 7k/8$ clauses are satisfied.

$$\frac{7}{8}k = E[Z] = \sum_{0 \leq j \leq k} j\, p_j = \sum_{0 \leq j < 7k/8} j\, p_j + \sum_{7k/8 \leq j \leq k} j\, p_j$$

$$\leq \left(\frac{7}{8}k - \frac{1}{8}\right) \sum_{0 \leq j < 7k/8} p_j + k \sum_{7k/8 \leq j \leq k} p_j$$

$$\leq \left(\frac{7}{8}k - \frac{1}{8}\right) \cdot 1 + kp$$

Solving for p yields $p \geq 1/(8k)$. □

Theorem

Johnson's algorithm is a 7/8-approximation algorithm that runs in expected polynomial time.

Proof.

- Approximation ratio is guaranteed, if algorithm stops.
  - Algorithm only stops if it finds an assignment that satisfies 7/8 of the clauses.
  - *Recall:* Such an assignment must exist.
- By the Satisfaction Probability Lemma, the probability of finding this assignment in the current iteration is $\geq 1/(8k)$.
- By the Waiting Time Bound, the expected number of iterations to find the assignment $\leq 8k$.

□