4/21/15: Randomized Algorithms II

Solving 3-SAT

Input & Initial Analysis

- *variables* $x_1, ..., x_n$
- *clauses* $C_1, ..., C_m$
  - each with ≤ 3 terms
- Easy Algorithm: try *all* 2^n truth assignments
  - $O(m2^n)$ time
- **Goal:** $O(m^a b^n)$ where $b < 2, a$ is a constant

What happens when set variables?

$$C_1 = \bar{x}_1 \vee x_2$$

- if set $x_1 = 0$,
  - then clause is satisfied
- if set $x_1 = 1$,
  - then $T \vee x_2 = x_2$, have fewer terms in clause
  - **notation:** T = True and F = False
- If a clause has one term, it determines the value of the variable in that term.

Backtracking

Backtrack

```
input:
init:
begin
  Pick any clause, sat $t_1 \vee t_2 \vee t_3$
  Try making $t_1 = T$                          // $n-1$ variables left
  if cannot find a satisfying assignment with $t_1 = T$,
    Try making $t_1 = F$ and $t_2 = T$   // $n-2$ variables left
  if cannot find a satisfying assignment with $t_1 = F$ and $t_2 = T$
    Try making $t_1 = F$, $t_2 = F$, $t_3 = T$ // $n-3$ variables left

  if none of these assignments finds a satisfying assignment
    return "not satisfiable"
end
```

$$T(n) \leq T(n-1) + T(n-2) + T(n-3) + ... + (\text{small terms})$$
$$\text{guess } T(n) = b^n$$

need

$$b^n \leq b^{n-1} + b^{n-2} + b^{n-3} \Leftrightarrow b^3 \leq b^2 + b + 1$$

- $b \approx 1.84$
- $T(n) \leq O(m1.84^n)$

Shoning (1999)

- expected time: $(\frac{3}{4})^n$

Idea

- Assume that there is a satisfying assignment, X^*, and let x be a setting that does not satisfy all clauses.
- Let C_j be any clause that is not satisfied by x, then x and x^* must differ on a variable in C_j.
- So, pick a random variable in that clause, and flip its value.
- Now, that clause is satisfied.

Algorithm 1

```
input:
init:
begin
  Pick x^0 uniformly at random in {0.1}^n. x = x^0
  for i = 1...n :
    if there is clause left unset by x
      let C_j such a clause
      pick a random var in C_j , and flip its value
end
```

- if $\exists$ a satisfying assignment, *Algorithm 1* finds one with $Pr \geq (\frac{2}{3})^n$

Hamming Distance

$$d(x^0, x^*) = \text{ number of vars in which x and } x^* \text{ differ}$$

$$Pr[d(x^0, x^*) = k] = \frac{\binom{n}{k}}{2^n}$$

- in each step, $Pr[d(x, x^*) decreases] \geq \frac{1}{3}$
- let $Y = \{$in all of the first k steps, $d(x^0, x^*)$ decreases or we find a satisfying x before then$\}$

$$Pr[\{d(x,x^*)=k\} \wedge Y] \geq \frac{\binom{n}{k}}{2^n}(\frac{1}{3})^k$$

$$Pr[\text{alg finds a sat. assignment}] \geq \sum_{k=0}^{n} \frac{\binom{n}{k}}{2^n}(\frac{1}{3})^k = \frac{1}{2^n}\sum_{k=0}^{n}\binom{n}{k}(\frac{1}{3})^k$$

- $= \frac{1}{2^n}(1+\frac{1}{3})^n = \frac{1}{2^n}(\frac{4}{3})^n = (\frac{2}{3})^n$
- Run it $(\frac{3}{2})^n$ times

Looking at "Progress"

- a step is "good" if x satisfies, or $d(x,x^*)$ decreases, "bad" otherwise
- if $d(x^0,x^*)=k$ and $\{\#$ "good" steps$\} = \{\#$ "bad" steps$\} \geq k$, we're done.
- so, just look inside the first $3k$ steps!
- that is:

$$Pr[d(x,x^*)=k \text{ and } \geq 2k \text{ of first } 3k \text{ steps are good}]$$
$$\geq \frac{\binom{n}{k}}{2^n}\binom{3k}{2k}(\frac{1}{3})^{2k}(\frac{2}{3})^k = 2^{-n}\binom{n}{k}\binom{3k}{2k}\frac{2k}{3^{3k}}$$

- **recall:** $\binom{n}{k} = \frac{n!}{(n-k)!k!}$
  - $n! \approx \sqrt{2\pi n}\frac{n^n}{e^n}$
  - $\binom{3k}{2k} \geq \frac{1}{\sqrt{5k}}\frac{3^{3k}}{2^{2k}}$, where $k>1$

$$Pr[\text{alg works}] \geq \sum_k 2^{-n}\binom{n}{k}\frac{1}{2^k}\frac{1}{\sqrt{5k}}$$

$$\geq \frac{1}{\sqrt{5k}}2^{-n}\sum_k\binom{n}{k}\frac{1}{2^k} = \frac{1}{\sqrt{5k}}2^{-n}(\frac{3}{2})^n = \frac{1}{\sqrt{5k}}(\frac{3}{4})^n$$