

8/10/2015

Russell Lec 13

How much can we save over exhaustive search?

Generic problem: n : # of vars, Σ : set of possible values, m : # of clauses

Exhaustive search: $|\Sigma|^n \cdot \text{poly}(n)$

Ex: Σ : $\{0,1\}$, n bool vars, 2 values, $\{0,1\}$ m clauses, $2^n \cdot \text{poly}(n)$

3-col: $n = |V|$, $\Sigma = \{R, G, B\}$, $m = |E|$ $3^{|V|} \cdot |E|$

Ind. set: $n = |V|$, $\Sigma = \{\text{in ind. set}, \text{out}\}$, $m = |E|$, $2^{|V|} \cdot |E|$

An alg is subexponential if it is in time $2^{o(n)} \cdot \text{poly}(n)$

Moderate improvement: $|\Sigma|^{(1-d)n} \cdot \text{poly}(n)$, $d > 0$.

Non-trivial: $|\Sigma|^n \cdot \text{poly}(m) / n^{O(1)}$

Most reductions: $(n, m) \rightarrow (n', m')$, unless $n' = O(n + m)$, $m' = O(n + m)$

Given m small, or $n' = f(n)$. $n' = O(n)$

$m = O(n)$: Sparsification.

Strongly subexp: $2^{n^{o(1)}}$ - standard reductions work

Sparsification & trade n vs m time vs describing parameters

Schoning's alg for k -SAT.

almost Schoning's alg: Sch A: Guess assignment \mathcal{C}_0 at random.

(Note that \mathcal{C}_0 is close to a SAT assign)

Let $d = \text{Ham}(\mathcal{C}_0, \mathcal{C}_i) = \#\{j: \mathcal{C}_0[j] \neq \mathcal{C}_i[j]\}$

until \exists unsat clause, or until \mathcal{C}_i is \mathcal{C}_0 (in n times) not in original Schoning's alg.

Pick such clause (l_1, \dots, l_k) : in \mathcal{C}_0 , all $l_i = 0$

pick i at random, and flip the bit in \mathcal{C}_0 corresponding to \mathcal{C}_i . Never if fixed.

If \mathcal{C}_0 got to SAT assign, great. If fixed vars force a clause to unsat, restart.

worst case 1 uniform \mathcal{C}_0 , and each clause just has one positive var.

Let $d_t = \text{Ham}(\mathcal{C}_t, \mathcal{C}_0)$ idid x at time step t . w/ prob $\geq 1/k$, $d_{t+1} = d_t - 1$

Prob (recur \mathcal{C}_0 or some other \mathcal{C}_i exists $\geq d_t$ from \mathcal{C}_0) $\leq (1/k)^{d_t}$

$$\sum_{d=0}^n \binom{n}{d} \cdot 2^{-n} \cdot (1/k)^d = 2^{-n} \sum_{d=0}^n \binom{n}{d} \cdot k^{-d} = 2^{-n} (1 + 1/k)^n = 2^{-n} \left(\frac{k+1}{k}\right)^n$$

- gives moderate improvement for each fixed \mathcal{C}_0 . Exp. time is

$$2^n \cdot \left(\frac{k}{k+1}\right)^n = 2^n \left(1 - \frac{1}{k+1}\right)^n = 2^n (1 - O(1/k))$$

Sch B (actual Schoning's alg): w/o fixing, repeat main loop $\leq n$ times

how prob. $\frac{k-1}{k}$ of \mathcal{C}_i increasing d_i .

$$\binom{n}{d} \cdot 2^{-n} \sum_{i=0}^{d-1} \binom{d+i}{i} \cdot \left(\frac{1}{k}\right)^{d+i} \left(\frac{k-1}{k}\right)^i$$

$$= 2^n \left(1 - \frac{1}{k}\right)^n \text{ time } (k \text{ instead of } k+1)$$

Between this and PPSZ, best exact alg for k -SAT (worst case).

For ind. set, a different approach: backtracking in this next prog

Tarjan & Trojanovski:

$TT(G)$ "T(G, N_1, \dots, N_i, a_i)"

If $V = \emptyset$, return \emptyset . If $|E| = \emptyset$, return V .

Pick $x \in V$ "candidate"

$$S_1 = \{x\} \cup TT(G - \{x\} - N(x))$$

$$S_0 = TT(G - \{x\})$$

Return S_1 unless S_0 is larger; if S_0 , return S_0

works well on high-degree graphs. On a chain, either choose 1 or 2 nodes

for S_0 to be larger, need to pick at least two neighbors of x

N_1, \dots, N_i : # of nodes to pick from each set (??)

- with picking carefully ~~enumeration~~ $\downarrow^{n^{13}}$.

Robson, w/ memoization & other tricks, helps improve 2^{2n} to $2^{2\sqrt{n}}$

Sparsification Johnson & Stepanov $|E| \leq O(|V|)$.

$TTS(G)$:

if G is sparse, return G

pick $x \in V$ "candidate" of degree $\geq 2d$

$$G_1 = \{x\} \cup TTS(G - \{x\} - N(x))$$

$$G_0 = TTS(G - \{x\})$$

Return $\{G_1\} \cup \{G_0\}$. // max ind set $\in G$ is max ind set in G_0 or G_1

$$\begin{aligned} \text{if } |E| \leq \epsilon \text{ degrees: } T(n) &= T(n - 2d - 1) + T(n - 1) \leq \binom{n}{n - 2d - 1} = \\ &= 2^n \frac{(2d+1)!}{2^{2d+1}} \end{aligned}$$

So $\forall \epsilon \exists d$ s.t. can reduce general ind. set to d -sparse d -ind set in 2^{2n} time

~ map one instance to many instances, but each instance is sparse.

Cor If \exists S s.t. time $T_{MS} \geq 2^{2n}$, then $\exists d$ s.t. $T_{\text{sparse}}(n) \geq 2^{2n}$

For k -SAT: same kind of sparsification requirement for CIM.

[IPZ] Sparsification lemma for k -SAT: $\forall k, \epsilon \exists \delta = O((k/\epsilon)^k)$ s.t. every

[IP] formula \mathcal{U} can be written as $\mathcal{U} = \bigvee_{i=1}^{2^{2n}} \mathcal{U}_i$ where each \mathcal{U}_i is a k -CNF with some vars and $\leq \delta n$ clauses. (in time $2^{\epsilon n}$).

(so \mathcal{U} is SAT $\Leftrightarrow \mathcal{U}_i$ is SAT)