The Probabilistic Method IV

497 - Randomized Algorithms

Sariel Har-Peled

October 10, 2002

At other times you seemed to me either pitiable or contemptible, eunuchs, artificially confined to an eternal childhood, childlike and childish in your cool, tightly fenced, neatly tidied playground and kindergarten, where every nose is carefully wiped and every troublesome emotion is soothed, every dangerous thought repressed, where everyone plays nice, safe, bloodless games for a lifetime and every jagged stirring of life, every strong feeling, every genuine passion, every rapture is promptly checked, deflected and neutralized by meditation therapy. – The Glass Bead Game, Hermann Hesse

1 The Lovász Local Lemma

Lemma 1.1 *(i)* $\mathbf{Pr}\Big[A \;\Big|\; B \cap C\Big] = \frac{\mathbf{Pr}\Big[A \cap B \;\Big|\; C\Big]}{\mathbf{Pr}\Big[B \;\Big|\; C\Big]}$

(ii) Let $\eta_1, \ldots, \eta_n$ be n events which are not necessarily independent. Then,

$$\Pr\left[\cap_{i=1}^{n} \eta_i\right] = \Pr\left[\eta_1\right] * \Pr\left[\eta_2 \;\middle|\; \eta_1\right] * \Pr\left[\eta_3 \;\middle|\; \eta_1 \cap \eta_2\right] * \ldots * \Pr\left[\eta_n \;\middle|\; \eta_1 \cap \ldots \cap \eta_{n-1}\right].$$

Proof:

$$\frac{\mathbf{Pr}\Big[A \cap B \;\Big|\; C\Big]}{\mathbf{Pr}\Big[B \;\Big|\; C\Big]} = \frac{\mathbf{Pr}[A \cap B \cap C]}{\mathbf{Pr}[C]} \Big/ \frac{\mathbf{Pr}[B \cap C]}{\mathbf{Pr}[C]} = \frac{\mathbf{Pr}[A \cap B \cap C]}{\mathbf{Pr}[B \cap C]} = \mathbf{Pr}\Big[A \;\Big|\; B \cap C\Big].$$

As for (ii), we already saw it and used it in the minimum cut algorithm lecture. ■

Lemma 1.2 (Lovász Local Lemma) *Let $G(V, E)$ be a dependency graph for events $C_1, \ldots, C_n$. Suppose that there exist $x_i \in [0, 1]$, for $1 \leq i \leq n$ such that*

$$\mathbf{Pr}[C_i] \leq x_i \prod_{(i,j) \in E} (1 - x_j).$$

Then

$$\mathbf{Pr}\big[\cap_{i=1}^{n} \overline{C_i}\big] \geq \prod_{i=1}^{n} (1 - x_i).$$

Proof: Let S denote a subset of the vertices from $\{1,\ldots,n\}$. We first establish by induction on $k = |S|$ that for any S and for any i such that $i \notin S$,

$$\mathbf{Pr}\Big[C_i \;\Big|\; \cap_{j\in S}\overline{C_j}\Big] \leq x_i. \tag{1}$$

For $S = \emptyset$, we have by assumption that $\mathbf{Pr}\Big[C_i \;\Big|\; \cap_{j\in S}\overline{C_j}\Big] = \mathbf{Pr}[C_i] \leq x_i \prod_{(i,j)\in E}(1-x_j) \leq x_i$.

Thus, let $N = \Big\{j \in S \;\Big|\; (i,j) \in E\Big\}$, and let $R = S \setminus N$. If $N = \emptyset$, then we have that C_i is mutually independent of the events of $\mathcal{C}(R) = \Big\{C_j \;\Big|\; j \in R\Big\}$. Thus, $\mathbf{Pr}\Big[C_i \;\Big|\; \cap_{j\in S}\overline{C_j}\Big] = \mathbf{Pr}\Big[C_i \;\Big|\; \cap_{j\in R}\overline{C_j}\Big] = \mathbf{Pr}[C_i] \leq x_i$, by arguing as above.

By Lemma 1.1 (i), we have that

$$\mathbf{Pr}\left[C_i \;\middle|\; \bigcap_{j\in S}\overline{C_j}\right] = \frac{\mathbf{Pr}\Big[C_i \cap \big(\cap_{j\in N}\overline{C_j}\big) \;\Big|\; \cap_{m\in R}\overline{C_m}\Big]}{\mathbf{Pr}\Big[\cap_{j\in N}\overline{C_j} \;\Big|\; \cap_{m\in R}\overline{C_m}\Big]}.$$

We bound the numerator by

$$\mathbf{Pr}\Big[C_i \cap \big(\cap_{j\in N}\overline{C_j}\big) \;\Big|\; \cap_{m\in R}\overline{C_m}\Big] \leq \mathbf{Pr}\Big[C_i \;\Big|\; \cap_{m\in R}\overline{C_m}\Big] = \mathbf{Pr}[C_i] \leq x_i \prod_{(i,j)\in E}(1-x_j)\,,$$

since C_i is mutually independent of $\mathcal{C}(R)$. As for the denominator, let $N = \{j_1,\ldots,j_r\}$. We have, by Lemma 1.1 (ii), that

$$\begin{aligned}
\mathbf{Pr}\Big[\overline{C_{j_1}} \cap \ldots \cap \overline{C_{j_r}} \;\Big|\; \cap_{m\in R}\overline{C_m}\Big] &= \mathbf{Pr}\Big[\overline{C_{j_1}} \;\Big|\; \cap_{m\in R}\overline{C_m}\Big]\, \mathbf{Pr}\Big[\overline{C_{j_2}} \;\Big|\; \overline{C_{j_1}} \cap \big(\cap_{m\in R}\overline{C_m}\big)\Big] \\
&\quad \cdots \mathbf{Pr}\Big[\overline{C_{j_r}} \;\Big|\; \overline{C_{j_1}} \cap \ldots \cap \overline{C_{j_{r-1}}}\big(\cap_{m\in R}\overline{C_m}\big)\Big] \\
&= \Big(1 - \mathbf{Pr}\Big[C_{j_1} \;\Big|\; \cap_{m\in R}\overline{C_m}\Big]\Big)\Big(1 - \mathbf{Pr}\Big[C_{j_2} \;\Big|\; \overline{C_{j_1}} \cap \big(\cap_{m\in R}\overline{C_m}\big)\Big]\Big) \\
&\quad \cdots\Big(1 - \mathbf{Pr}\Big[C_{j_r} \;\Big|\; \overline{C_{j_1}} \cap \ldots \cap \overline{C_{j_{r-1}}}\big(\cap_{m\in R}\overline{C_m}\big)\Big]\Big) \\
&\geq (1-x_{j_1})\cdots(1-x_{j_r}) \geq \prod_{(i,j)\in E}(1-x_j)\,,
\end{aligned}$$

by Equation (1) and induction, as every probability term in the above expression has less than $|S|$ items involved. It thus follows, that $\mathbf{Pr}\Big[C_i \;\Big|\; \bigcap_{j\in S}\overline{C_j}\Big] \leq x_i$.

Now, the proof of the lemma, follows from

$$\mathbf{Pr}\big[\cap_{i=1}^{n}\overline{C_i}\big] = (1-\mathbf{Pr}[C_1])\Big(1 - \mathbf{Pr}\Big[C_2 \;\Big|\; \overline{C_1}\Big]\Big)\cdots\Big(1 - \mathbf{Pr}\Big[C_n \;\Big|\; \cap_{i=1}^{n-1}\overline{C_i}\Big]\Big) \geq \prod_{i=1}^{n}(1-x_i)\,. \;\blacksquare$$

Corollary 1.3 *Let $C_1,\ldots,C_n$ be events, with $\mathbf{Pr}[C_i] \leq p$ for all i. If each event is mutually independent of all other events except for at most d, and if $ep(d+1) \leq 1$, then $\mathbf{Pr}\big[\cap_{i=1}^{n}\overline{C_i}\big] > 0$.*

Proof: If $d = 0$ the result is trivial, as the events are independent. Otherwise, there is a dependency graph, with every vertex having degree at most d. Apply Lemma 1.2 with $x_i = \frac{1}{d+1}$. Observe that

$$x_i(1-x_i)^d = \frac{1}{d+1}\left(1-\frac{1}{d+1}\right)^d > \frac{1}{d+1}\cdot\frac{1}{e} \geq p,$$

by assumption and the fact that $\left(1-\frac{1}{d+1}\right)^d > 1/e$. To see that, observe that, observe that we need to show that $1/\left(1-\frac{1}{d+1}\right)^d < e$, which is equivalent to $((d+1)/d) < e^{1/d}$. However,

$$\frac{d+1}{d} = 1+\frac{1}{d} < 1+\left(\frac{1}{d}\right)+\frac{1}{2!}\left(\frac{1}{d}\right)^2+\frac{1}{3!}\left(\frac{1}{d}\right)^3+\cdots = e^{1/d},$$

establishing the claim. ∎

1.1 Application to k-SAT

We are given a instance I of k-SAT, where every clause contains k literals, there are m clauses, and every one of the n variables, appears in at most $2^{k/50}$ clauses.

Consider a random assignment, and let C_i be the event that the i-clause was not satisfied. We know that $p = \mathbf{Pr}[C_i] = 2^{-k}$, and furthermore, C_i depends on at most $d = k2^{k/50}$ other events. Since $ep(d+1) = e\left(k^{k/50}+1\right)2^{-k} < 1$, for $k \geq 4$, we conclude that by Corollary 1.3, that

$$\mathbf{Pr}[I \text{ have a satisfying assignment}] = \mathbf{Pr}[\cup_i C_i] > 0.$$

1.1.1 An efficient algorithm

The above, just prove that a satisfying assignment exists. We next show a polynomial algorithm (in m) for the computation of such an assignment (the algorithm will not be polynomial in k).

Let G be the dependency graph I, where two clauses are connected if they share a variable. We start assigning values to the variables one by one.

Definition 1.4 A clause C_i is *dangerous* if both the following conditions hold:

1. $k/2$ literals of C_i have been fixed.
2. C_i is still unsatisfied.

After assigning each value, we discover all the dangerous clauses, and we defer all the unassigned variables participating in such a clause.

A clause had *survived* if it is not satisfied by the variables fixed in the first stage. Note, that a clause that survived must have a dangerous clause as a neighbor in the dependency graph G. Not that I', the instance remaining from I after the first stage, has at least $k/2$ unspecified variables in each clause. An every clause of I' has at most $d = 2^{k/50}$ neighbors in G. It follows, that again, we can apply Lovász local lemma to conclude that I' has a satisfying assignment.

Let G' denote the dependency graph for I'. We need the following technical lemma.

Definition 1.5 Two connected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, where $V_1, V_2 \subseteq \{1, \ldots, n\}$ are *unique* if $V_1 \neq V_2$.

Lemma 1.6 *Let G be a graph with degree at most d and with n vertices. Then, the number of unique subgraphs of G having r vertices is at most nd^{2r}.*

Proof: Let H be a connected subtree of G, and duplicate every edge of H. Let H' be the resulting graph from H. Clearly, H' is Eulerian, and as such posses a Eulerian path π of length at most $2(r-1)$, which can be specified, by picking a starting vertex v, and writing down for the i-th vertex of π which of the d possible neighbors, is the next vertex in π. Thus, there are st most $nd^{2(r-1)}$ ways of specifying π, and thus, there are at most $nd^{2(r-1)}$ unique subgraphs in G of size r. ∎

Lemma 1.7 *With probability $1 - o(1)$, all connected components of G' have size at most $O(\log m)$.*

Proof: Let G_4 be a graph formed from G by connecting any pair of vertices of G of distance *exactly* 4 from each other. The degree of a vertex of G_4 is at most $O(d^4)$.

Let U be a set of r vertices of G, such that every pair is in distance at least 4 from each other in G. We are interested in bounding the probability that all the clauses of U survive the first stage.

The probability that a clause survive is bounded by $2^{-k/2}(d+1)$. Furthermore, the survival of two clauses in U is an independent event, as no *neighbor* of $C_i.C_j \in U$ share a variable (because of the distance 4 requirement). We conclude, that the probability that all the vertices U appear in G' is bounded by

$$\left(2^{-k/2}(d+1)\right)^r.$$

On the other hand, the number of unique such sets of size r, is bounded by the number of unique subgraphs of G_4 of size r, which is bounded by md^{8r}, by Lemma 1.6. Thus, the probability of any connected subgraph of G_4 of size $r = b \log m$ to survive in G' is smaller than

$$md^{8r}\left(2^{-k/2}(d+1)\right)^r = o(1).$$

Note, however, that if a connected component of G' has more than L vertices, than there must be a connected component having L/d^3 vertices in G_4 that had survived in G'. We conclude, that with probability $o(1)$, no connected component of G' has more than $O(d^3 \log m) = O(\log m)$ vertices (note, that we consider k to be a constant, and thus, also d). ∎

Thus, after the first stage, we are left with fragments of $(k/2)$-SAT, where every fragment has size at most $O(\log m)$, and thus having at most $O(\log m)$ variables. Thus, we can by brute force find the satisfying assignment to each such fragment in time polynomial in m. We conclude:

Theorem 1.8 *The above algorithm finds a satisfying truth assignment for any instance of k-SAT containing m clauses, which each variable is contained in at most $2^{k/50}$ clauses, in expected time polynomial in m.*