Chapter 49

The Probabilistic Method III

By Sariel Har-Peled, April 26, 2022[1]

> 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

49.1. The Lovász Local Lemma

Lemma 49.1.1. *(i)* $\mathbb{P}\left[A \,\middle|\, B \cap C\right] = \dfrac{\mathbb{P}\left[A \cap B \,\middle|\, C\right]}{\mathbb{P}\left[B \,\middle|\, C\right]}$

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

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

Proof: (i) We have that

$$\frac{\mathbb{P}\left[A \cap B \,\middle|\, C\right]}{\mathbb{P}\left[B \,\middle|\, C\right]} = \frac{\mathbb{P}[A \cap B \cap C]}{\mathbb{P}[C]} \Bigg/ \frac{\mathbb{P}[B \cap C]}{\mathbb{P}[C]} = \frac{\mathbb{P}[A \cap B \cap C]}{\mathbb{P}[B \cap C]} = \mathbb{P}\left[A \,\middle|\, B \cap C\right].$$

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

Definition 49.1.2. An event $\mathcal{E}$ is mutually independent of a set of events $\mathcal{C}$, if for any subset $\mathcal{U} \subseteq \mathcal{C}$, we have that $\mathbb{P}[\mathcal{E} \cap (\bigcap_{\mathcal{E}' \in U} \mathcal{E}')] = \mathbb{P}[\mathcal{E}]\, \mathbb{P}[\bigcap_{\mathcal{E}' \in U} \mathcal{E}']$.

Let $\mathcal{E}_1, \ldots, \mathcal{E}_n$ be events. A ***dependency graph*** for these events is a directed graph $\mathsf{G} = (\mathsf{V}, \mathsf{E})$, where $\{1, \ldots, n\}$, such that $\mathcal{E}_i$ is mutually independent of all the events in $\left\{\mathcal{E}_j \,\middle|\, (i,j) \notin \mathsf{E}\right\}$.

Intuitively, an edge (i, j) in a dependency graph indicates that $\mathcal{E}_i$ and $\mathcal{E}_j$ have (maybe) some dependency between them. We are interested in settings where this dependency is limited enough, that we can claim something about the probability of all these events happening simultaneously.

Lemma 49.1.3 (Lovász Local Lemma). *Let $\mathsf{G}(\mathsf{V}, \mathsf{E})$ be a dependency graph for events $\mathcal{E}_1, \ldots, \mathcal{E}_n$. Suppose that there exist $x_i \in [0,1]$, for $1 \le i \le n$ such that $\mathbb{P}[\mathcal{E}_i] \le x_i \prod_{(i,j) \in E} (1 - x_j)$. Then $\mathbb{P}\left[\cap_{i=1}^{n} \overline{\mathcal{E}_i}\right] \ge \prod_{i=1}^{n} (1 - x_i)$.*

We need the following technical lemma.

Lemma 49.1.4. *Let $\mathsf{G}(\mathsf{V},\mathsf{E})$ be a dependency graph for events $\mathcal{E}_1, \ldots, \mathcal{E}_n$. Suppose that there exist $x_i \in [0,1]$, for $1 \le i \le n$ such that $\mathbb{P}[\mathcal{E}_i] \le x_i \prod_{(i,j)\in E} (1 - x_j)$. Now, let S be a subset of the vertices from $\{1,\ldots,n\}$, and let i be an index not in S. We have that*

$$\mathbb{P}\left[\mathcal{E}_i \,\middle|\, \cap_{j\in S} \overline{\mathcal{E}_j}\right] \le x_i. \tag{49.1}$$

Proof: The proof is by induction on $k = |S|$.

For $k = 0$, we have by assumption that $\mathbb{P}\left[\mathcal{E}_i \,\middle|\, \cap_{j\in S} \overline{\mathcal{E}_j}\right] = \mathbb{P}[\mathcal{E}_i] \le x_i \prod_{(i,j)\in E}(1 - x_j) \le x_i$.

Thus, let $N = \left\{ j \in S \,\middle|\, (i,j) \in E \right\}$, and let $R = S \setminus N$. If $N = \emptyset$, then we have that $\mathcal{E}_i$ is mutually independent of the events of $\mathcal{C}(R) = \left\{ \mathcal{E}_j \,\middle|\, j \in R \right\}$. Thus, $\mathbb{P}\left[\mathcal{E}_i \,\middle|\, \cap_{j\in S} \overline{\mathcal{E}_j}\right] = \mathbb{P}\left[\mathcal{E}_i \,\middle|\, \cap_{j\in R} \overline{\mathcal{E}_j}\right] = \mathbb{P}[\mathcal{E}_i] \le x_i$, by arguing as above.

By Lemma 49.1.1 (i), we have that

$$\mathbb{P}\left[\mathcal{E}_i \,\middle|\, \bigcap_{j\in S} \overline{\mathcal{E}_j}\right] = \frac{\mathbb{P}\left[\mathcal{E}_i \cap \left(\cap_{j\in N} \overline{\mathcal{E}_j}\right) \,\middle|\, \cap_{m\in R} \overline{\mathcal{E}_m}\right]}{\mathbb{P}\left[\cap_{j\in N} \overline{\mathcal{E}_j} \,\middle|\, \cap_{m\in R} \overline{\mathcal{E}_m}\right]}.$$

We bound the numerator by

$$\mathbb{P}\left[\mathcal{E}_i \cap \left(\cap_{j\in N} \overline{\mathcal{E}_j}\right) \,\middle|\, \cap_{m\in R} \overline{\mathcal{E}_m}\right] \le \mathbb{P}\left[\mathcal{E}_i \,\middle|\, \cap_{m\in R} \overline{\mathcal{E}_m}\right] = \mathbb{P}[\mathcal{E}_i] \le x_i \prod_{(i,j)\in E} (1 - x_j),$$

since $\mathcal{E}_i$ is mutually independent of $\mathcal{C}(R)$. As for the denominator, let $N = \{j_1, \ldots, j_r\}$. We have, by Lemma 49.1.1 (ii), that

$$\begin{aligned}
\mathbb{P}\left[\overline{\mathcal{E}_{j_1}} \cap \ldots \cap \overline{\mathcal{E}_{j_r}} \,\middle|\, \cap_{m\in R} \overline{\mathcal{E}_m}\right] &= \mathbb{P}\left[\overline{\mathcal{E}_{j_1}} \,\middle|\, \cap_{m\in R} \overline{\mathcal{E}_m}\right] \mathbb{P}\left[\overline{\mathcal{E}_{j_2}} \,\middle|\, \overline{\mathcal{E}_{j_1}} \cap \left(\cap_{m\in R} \overline{\mathcal{E}_m}\right)\right] \\
&\quad \cdots \mathbb{P}\left[\overline{\mathcal{E}_{j_r}} \,\middle|\, \overline{\mathcal{E}_{j_1}} \cap \ldots \cap \overline{\mathcal{E}_{j_{r-1}}} \cap \left(\cap_{m\in R} \overline{\mathcal{E}_m}\right)\right] \\
&= \left(1 - \mathbb{P}\left[\mathcal{E}_{j_1} \,\middle|\, \cap_{m\in R} \overline{\mathcal{E}_m}\right]\right) \left(1 - \mathbb{P}\left[\mathcal{E}_{j_2} \,\middle|\, \overline{\mathcal{E}_{j_1}} \cap \left(\cap_{m\in R} \overline{\mathcal{E}_m}\right)\right]\right) \\
&\quad \cdots \left(1 - \mathbb{P}\left[\mathcal{E}_{j_r} \,\middle|\, \overline{\mathcal{E}_{j_1}} \cap \ldots \cap \overline{\mathcal{E}_{j_{r-1}}} \cap \left(\cap_{m\in R} \overline{\mathcal{E}_m}\right)\right]\right) \\
&\ge (1 - x_{j_1}) \cdots (1 - x_{j_r}) \ge \prod_{(i,j)\in E} (1 - x_j),
\end{aligned}$$

by Eq. (49.1) and induction, as every probability term in the above expression has less than $|S|$ items involved. It thus follows, that $\mathbb{P}\left[\mathcal{E}_i \,\middle|\, \cap_{j\in S} \overline{\mathcal{E}_j}\right] \le x_i$. ∎

Proof of Lovász local lemma (Lemma 49.1.3): Using Lemma 49.1.4, we have that

$$\mathbb{P}\left[\cap_{i=1}^{n} \overline{\mathcal{E}_i}\right] = (1 - \mathbb{P}[\mathcal{E}_1])\left(1 - \mathbb{P}\left[\mathcal{E}_2 \,\middle|\, \overline{\mathcal{E}_1}\right]\right) \cdots \left(1 - \mathbb{P}\left[\mathcal{E}_n \,\middle|\, \cap_{i=1}^{n-1} \overline{\mathcal{E}_i}\right]\right) \ge \prod_{i=1}^{n} (1 - x_i).$$

∎

Corollary 49.1.5. *Let $\mathcal{E}_1, \ldots, \mathcal{E}_n$ be events, with $\mathbb{P}[\mathcal{E}_i] \le p$ for all i. If each event is mutually independent of all other events except for at most d, and if $ep(d+1) \le 1$, then $\mathbb{P}\left[\cap_{i=1}^n \overline{\mathcal{E}_i}\right] > 0$.*

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 49.1.3 with $x_i = \frac{1}{d+1}$. Observe that

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

by assumption and the since $\left(1-\frac{1}{d+1}\right)^d > 1/e$, see Lemma 49.1.6 below. ∎

The following is standard by now, and we include it only for the sake of completeness.

Lemma 49.1.6. *For any $n \ge 1$, we have $\left(1-\frac{1}{n+1}\right)^n > \frac{1}{e}$.*

Proof: This is equivalent to $\left(\frac{n}{n+1}\right)^n > \frac{1}{e}$. Namely, we need to prove $e > \left(\frac{n+1}{n}\right)^n$. But this obvious, since $\left(\frac{n+1}{n}\right)^n = \left(1+\frac{1}{n}\right)^n < \exp(n(1/n)) = e$. ∎

49.2. 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 $\mathcal{E}_i$ be the event that the ith clause was not satisfied. We know that $p = \mathbb{P}[\mathcal{E}_i] = 2^{-k}$, and furthermore, $\mathcal{E}_i$ depends on at most $d = k2^{k/50}$ other events. Since $ep(d+1) = e\left(k\cdot 2^{k/50}+1\right)2^{-k} < 1$, for $k \ge 4$, we conclude that by Corollary 49.1.5, that

$$\mathbb{P}\Big[I \text{ have a satisfying assignment}\Big] = \mathbb{P}\Big[\cup_i \mathcal{E}_i\Big] > 0.$$

49.2.1. An efficient algorithm

The above just proves 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 for I, where the vertices are the clauses of I, and two clauses are connected if they share a variable. In the first stage of the algorithm, we assign values to the variables one by one, in an arbitrary order. In the beginning of this process all variables are unspecified, at each step, we randomly assign a variable either 0 or 1 with equal probability.

Definition 49.2.1. A clause $\mathcal{E}_i$ is ***dangerous*** if both the following conditions hold:

(i) $k/2$ literals of $\mathcal{E}_i$ have been fixed.

(ii) $\mathcal{E}_i$ is still unsatisfied.

After assigning each value, we discover all the dangerous clauses, and we defer ("freeze") all the unassigned variables participating in such a clause. We continue in this fashion till all the unspecified variables are frozen. This completes the first stage of the algorithm.

At the second stage of the algorithm, we will compute a satisfying assignment to the variables using brute force. This would be done by taking the surviving formula I' and breaking it into fragments, so

that each fragment does not share any variable with any other fragment (naively, it might be that all of I' is one fragment). We can find a satisfying assignment to each fragment separately, and if each such fragment is "small" the resulting algorithm would be "fast".

We need to show that I' has a satisfying assignment and that the fragments are indeed small.

49.2.1.1. Analysis

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. Furthermore, every clause of I' has at most $d = k2^{k/50}$ neighbors in G', where G' is the dependency graph for I'. It follows, that again, we can apply Lovász local lemma to conclude that I' has a satisfying assignment.

Definition 49.2.2. Two connected graphs $\mathsf{G}_1 = (V_1, E_1)$ and $\mathsf{G}_2 = (V_2, E_2)$, where $V_1, V_2 \subseteq \{1, \ldots, n\}$ are ***unique*** if $V_1 \neq V_2$.

Lemma 49.2.3. *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: Consider a unique subgraph $\widehat{G}$ of G, which by definition is connected. Let H be a connected subtree of G spanning $\widehat{G}$. Duplicate every edge of H, and let H' denote the resulting graph. 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 49.2.4. *With probability $1 - o(1)$, all connected components of G' have size at most $O(\log m)$, where G' denote the dependency graph for I'.*

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 of a clause to be dangerous is at most $2^{-k/2}$, as we assign (random) values to half of the variables of this clause. Now, a clause survive only if it is dangerous or one of its neighbors is dangerous. Thus, the probability that a clause survive is bounded by $2^{-k/2}(d+1)$.

Furthermore, the survival of two clauses $\mathcal{E}_i$ and $\mathcal{E}_j$ in U is an independent event, as no *neighbor* of $\mathcal{E}_i$ shares a variable with a neighbor of $\mathcal{E}_j$ (because of the distance 4 requirement). We conclude, that the probability that all the vertices of U to appear in G' is bounded by

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

In fact, we are interested in sets U that induce a connected subgraphs of G_4. 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 49.2.3. Thus, the probability of any connected subgraph of G_4 of size $r = \log_2 m$ to survive in G' is smaller than

$$md^{8r}\left(2^{-k/2}(d+1)\right)^r = m\left(k2^{k/50}\right)^{8r}\left(2^{-k/2}(k2^{k/50}+1)\right)^r \leq m2^{kr/5}\cdot 2^{-kr/4} = m2^{-kr/20} = o(1),$$

since $k \geq 50$. (Here, a subgraph survive of G_4 survive, if all its vertices appear in G'.) 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 49.2.5. *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.*