

Lecture 9. Algorithmic LLL for general probability spaces

In the last two lectures we presented an algorithmic proof of the LLL (and in fact Shearer’s lemma) when the underlying probability space consists of independent random variables. In this lecture we present an “algorithmic proof” of Shearer’s Lemma for a general probability space. We note that the first abstract algorithmic framework for the LLL beyond [Moser and Tardos (2010)] was proposed by [Achlioptas and Iliopoulos (2014)]. Here we follow [Harvey and Vondrak (2015)].

Let Ω be a finite space with probability measure μ , and $\mathcal{E}_1, \dots, \mathcal{E}_n \subset \Omega$ events in that probability space with dependency graph G . Similarly to the Moser-Tardos algorithm, we will choose an $\omega \in \Omega$ at random and then repeatedly “resample” in a certain way until we find an $\omega \in \bigcap_{i=1}^n \mathcal{E}_i$. To formalize this we define *resampling oracles*.

9.1 Resampling oracles

Definition 9.1 A *resampling oracle* for event \mathcal{E}_i with respect to events $\mathcal{E}_1, \dots, \mathcal{E}_n$ and a graph G is a mapping $r_i : \Omega \rightarrow \mathcal{D}(\Omega)$, where $\mathcal{D}(\Omega)$ is the space of random variables with values in Ω (distributed arbitrarily, not necessarily according to μ), such that the following hold:

(1) If ω is random, distributed according to $\mu|_{\mathcal{E}_i}$ (the restriction of μ to where \mathcal{E}_i occurs), then the random variable $r_i(\omega)$ is distributed according to μ .

(2) If $\omega \notin \mathcal{E}_j$ is fixed, $i \notin \Gamma^+(j)$, then with probability 1 we have $r_i(\omega) \notin \mathcal{E}_j$.

To understand condition (1) we can look at the special case where Ω consists of underlying independent random variables X_1, \dots, X_m with μ the uniform probability distribution, \mathcal{E}_1 is determined by a particular subset of these variables: (X_1, X_2, X_3) , and the resampling oracle r_1 is as in Moser-Tardos, where X_1, X_2, X_3 are resampled uniformly and the other variables left unchanged. Then if $\omega = (X_1, \dots, X_m)$ is distributed according to $\mu|_{\mathcal{E}_1}$, the variables $\{X_4, \dots, X_m\}$ are uniformly random (as \mathcal{E}_1 is only dependent on X_1, X_2, X_3). The resampled state $r_1(\omega)$ is obtained from ω by replacing X_1, X_2, X_3 by independent new samples X'_1, X'_2, X'_3 . I.e., $r_1(\omega) = (X'_1, X'_2, X'_3, X_4, \dots, X_m)$. Since all these variables are independent and uniformly random, $r_1(\omega)$ is distributed according to the uniform probability distribution μ . So the resampling satisfies condition (1).

Continuing with the above example, if \mathcal{E}_j is an event independent of \mathcal{E}_1 , i.e. \mathcal{E}_j does not depend in any way on X_1, X_2, X_3 , and $\omega = (X_1, \dots, X_m) \notin \mathcal{E}_j$, then resampling X_1, X_2, X_3 cannot cause \mathcal{E}_j to occur. Therefore the Moser-Tardos resampling also satisfies condition (2) and it is a special case of a resampling oracle.

Given the definition, one may ask whether such a resampling oracle even exists. The following lemma characterizes the existence of resampling oracles and relates this notion to the notion of a dependency graph.

Lemma 9.2 For any events $\mathcal{E}_1, \dots, \mathcal{E}_n$ with a dependency graph $G = (V, E)$, resampling oracles r_1, \dots, r_n exist.

More generally, r_i satisfying conditions (1) and (2) exists iff \mathcal{E}_i satisfies "lopsided association" with $\{\mathcal{E}_j : j \notin \Gamma^+(i)\}$: Namely, for any monotonically increasing function $f : \{0, 1\}^{V \setminus \Gamma^+(i)} \rightarrow \{0, 1\}$,

$$\mathbb{E}[f(\chi_{\mathcal{E}_j} : j \in V \setminus \Gamma^+(i)) \mid \mathcal{E}_i] \geq \mathbb{E}[f(\chi_{\mathcal{E}_j} : j \in V \setminus \Gamma^+(i))]$$

where $\chi_{\mathcal{E}_j}$ is the 0/1 indicator variable of \mathcal{E}_j .

Note that the condition of the lopsided LLL, $\mathbb{P}[\bigcup_{i \in J} \mathcal{E}_i \mid \mathcal{E}_i] \geq \mathbb{P}[\bigcup_{i \in J} \mathcal{E}_i]$ for $J \subseteq V \setminus \Gamma^+(i)$, is a special case of the "lopsided association" condition where $f(x) = \bigvee_{j \in J} x_j$.

Example: random trees. Let Ω be the space of spanning trees of K_n and μ the uniform probability distribution over Ω . For A a forest in K_n , let $\mathcal{E}_A = \{\omega \in \Omega : A \subset \omega\}$ denote the event that a random spanning tree ω contains A . As we mentioned earlier in Lecture 4, it is known that for A, B vertex-disjoint forests,

$$\mathbb{P}[A \subseteq \omega \ \& \ B \subseteq \omega] = \mathbb{P}[A \subseteq \omega] \cdot \mathbb{P}[B \subseteq \omega].$$

Therefore, the events \mathcal{E}_A with the relation of vertex non-disjointness form a dependency graph. Lemma 9.2 implies that resampling oracles for these events exist. It is instructive to see how these resampling oracles operate.

The resampling oracle r_A for a forest A works as follows: Let W denote the vertices of A . For a given spanning tree ω containing A , "freeze" the edges and non-edges on $V \setminus W$, and extend uniformly at random to a spanning tree of K_n . Equivalently, start from K_n , contract the edges of ω on $V \setminus W$, remove the non-edges of ω on $V \setminus W$, and choose uniformly at random a spanning tree of the resulting (multi-)graph. This resampling oracle satisfies condition (1) as (roughly) if ω is distributed according to $\mu|_A$, the part of the tree ω outside of W is chosen as it would be in a uniformly random tree, and then for $r_A(\omega)$ the rest of the tree is chosen uniformly, so $r_A(\omega)$ is a uniformly random tree. It also satisfies condition (2) as, if \mathcal{E}_B is an event independent of \mathcal{E}_A , that means A and B are vertex-disjoint, and if ω does not contain B , then $r_A(\omega)$ again does not contain B , since none of the edges of B are modified in the resampling.

We now prove the first part of Lemma 9.2, where G is a dependency graph. We skip the proof of the more general lopsided association condition.

Proof: Let \mathcal{E}_i be independent of $\{\mathcal{E}_j : j \in V \setminus \Gamma^+(i)\}$. Then for $\omega \in \mathcal{E}_i$, there is a unique partition $V \setminus \Gamma^+(i) = J_1 \cup J_2$ (determined by which events in $V \setminus \Gamma^+(i)$ hold and which don't) such that $\omega \in \bigcap_{j \in J_1} \mathcal{E}_j \cap \bigcap_{j' \in J_2} \overline{\mathcal{E}_{j'}}$. Note that by the independence property, $\mathbb{P}[\mathcal{E}_i \mid \bigcap_{j \in J_1} \mathcal{E}_j \cap \bigcap_{j' \in J_2} \overline{\mathcal{E}_{j'}}] = \mathbb{P}[\mathcal{E}_i]$.

Given J_1, J_2 , we let $r_i(\omega)$ be distributed according to μ conditioned on $\bigcap_{j \in J_1} \mathcal{E}_j \cap \bigcap_{j' \in J_2} \overline{\mathcal{E}_{j'}}$. In other words, we generate a random state $r_i(\omega)$ restricted to the subspace of Ω determined by the occurrence/non-occurrence of events in $V \setminus \Gamma^+(i)$.

Then r_i satisfies (1), since if ω is distributed according to $\mu|_{\mathcal{E}_i \cap \bigcap_{j \in J_1} \mathcal{E}_j \cap \bigcap_{j' \in J_2} \overline{\mathcal{E}_{j'}}}$, then $r_i(\omega)$ is distributed according to $\mu|_{\bigcap_{j \in J_1} \mathcal{E}_j \cap \bigcap_{j' \in J_2} \overline{\mathcal{E}_{j'}}}$. Hence, by the independence property above, if $\omega \sim \mu|_{\mathcal{E}_i}$ then $r_i(\omega) \sim \mu$. This shows property (1). Property (2) also holds, because the resampling does not change the status of any event in $V \setminus \Gamma^+(i)$. \square

We remark that this only demonstrates the *existence* of a resampling oracle. To obtain an efficient algorithm, we must implement the resampling oracles in an efficient way (as we did above

for random trees). This is not always possible, and indeed it is not always possible to make LLL or Shearer’s Lemma algorithmically efficient (see [Harvey and Vondrak (2015)]). On the other hand, it seems that efficient resampling oracles exist in all combinatorial settings of interest. Therefore, what we mean here by an “algorithmic proof of Shearer’s Lemma” is a proof that proves the existential lemma in general, and algorithmically reduces it to the problem of designing efficient resampling oracles.

9.2 The Algorithm

Given resampling oracles r_1, \dots, r_n for events $\mathcal{E}_1, \dots, \mathcal{E}_n$ and a graph G , we analyze the following algorithm for finding $\omega \in \bigcap_{i=1}^n \bar{\mathcal{E}}_i$.

Algorithm 9.3

1. Initialize ω by a random sample from μ .
2. As long as any events are satisfied by ω , repeat the following:
 - Set $J := \emptyset$;
 - While there is some event \mathcal{E}_i , $i \notin \Gamma^+(J)$, satisfied by ω , pick the minimum such i , set $\omega := r_i(\omega)$, and add i to J .

We note that in each step t (iteration of (2.)), an independent set I_t of events are resampled. Furthermore, any resamplings of events in I_{t+1} at time $t + 1$ must result from a resampling of an event in I_t at time t causing an event in I_{t+1} to occur. But by (2), the resampling of an event can only cause new events that are neighbors of that event, so $I_{t+1} \subseteq \Gamma^+(I_t)$. Thus for $t \geq 0$, (I_1, \dots, I_t) is a stable set sequence. Note that this sequence goes forward in time, in a fashion opposite to the backward analysis of [Kolipaka and Szegedy (2011)]. We call this the *forward analysis*, where we estimate the probability that the algorithm resamples some stable set sequence $\mathcal{I} = (I_1, \dots, I_t)$ by resampling the events in I_s in its s -th step. We have the following lemma, analogous to the main lemma of Lecture 8.

Lemma 9.4 $\mathbb{P}[\text{Algorithm resamples } \mathcal{I} = (I_1, I_2, \dots, I_t)] \leq \prod_{s=1}^t p^{I_s}$, provided that $\mathbb{P}_\mu[\mathcal{E}_i] \leq p_i$.

Proof: Given $\mathcal{I} = (I_1, \dots, I_t)$, we construct an “ \mathcal{I} -checking process.” Start with ω chosen according to μ . Beginning with $s = 1$, go through each $I_s = \{\mathcal{E}_{s_1}, \mathcal{E}_{s_2}, \dots, \mathcal{E}_{s_m}\}$, $s_1 < s_2 < \dots < s_m$, checking whether $\omega \in \mathcal{E}_{s_i}$ and, if so, resample \mathcal{E}_{s_i} , replacing ω with $r_{s_i}(\omega)$. If at any point the relevant event is not satisfied, we say the process *fails*. If this process goes through the entire sequence without failing, we say that it *succeeds*.

Now we couple the randomness of the algorithm with the randomness of the \mathcal{I} -checking process. Whenever the algorithm resamples \mathcal{I} , other things may be checked too, but the verifications of the events $\mathcal{E}_{s_1}, \mathcal{E}_{s_2}, \dots, \mathcal{E}_{s_m}$, each followed by resamplings, at each step s , must occur, so the \mathcal{I} -checking process must succeed. Hence the probability that the algorithm resamples \mathcal{I} is at most the probability that the \mathcal{I} -checking process succeeds.

Now in the \mathcal{I} -checking process, the distribution after each resampling of \mathcal{E}_{s_t} is μ ; assuming \mathcal{E}_{s_t} occurred, the distribution before resampling is the distribution $\mu|_{\mathcal{E}_{s_t}}$ — since by induction, μ was the distribution after the previous resampling, not assuming \mathcal{E}_{s_t} — so by (1) the distribution after resampling \mathcal{E}_{s_t} is μ . Thus, for \mathcal{E}_{s_t} one of the events in I_s , the probability that the \mathcal{I} -checking process

continues when checking \mathcal{E}_{s_t} is the probability that \mathcal{E}_{s_t} occurs after the previous resampling, which is $\mathbb{P}_\mu[\mathcal{E}_{s_t} \leq p_i]$. Since at the checking of \mathcal{E}_{s_t} the probability of the process continuing is $\mathbb{P}_\mu[\mathcal{E}_{s_t} \leq p_i]$, the probability of the entire process succeeding is the product of probabilities of the process continuing at each step, so is at most $\prod_{s=1}^t p^{I_s}$. Thus this is an upper bound on the probability that the algorithm resamples \mathcal{I} . \square

9.3 Expected Running Time of the Algorithm

Let Prop denote the set of all proper stable set sequences, that is stable set sequences consisting of nonempty sets. A straightforward application of Lemma 9.4, together with computations from the last lecture, gives the following bound.

$$\begin{aligned} \mathbb{E}[\# \text{ of resamplings}] &\leq \sum_{\mathcal{I}=(I_1, \dots, I_t) \in \text{Prop}} \mathbb{P}[\text{Algorithm resamples } \mathcal{I}] = \sum_{\mathcal{I}=(I_1, \dots, I_t) \in \text{Prop}} \prod_{s=1}^t p^{I_s} \\ &\leq \sum_{I \in \text{Ind}(G)} \frac{q_I}{q_\emptyset} = \frac{1}{q_\emptyset}. \end{aligned}$$

But $\frac{1}{q_\emptyset}$ is typically exponentially large; in fact this result can be obtained by simply sampling ω repeatedly according to μ , until we hit a point in $\bigcap_{i=1}^n \bar{\mathcal{E}}_i$. So this is no better than a brute-force algorithm!

Fortunately, there is a way around this which makes the bound much more efficient. By openness of Shearer's region, we have that if \mathbf{p} satisfies Shearer's conditions then so does $(1 + \varepsilon)\mathbf{p}$ for some $\varepsilon > 0$. This allows us to apply the following theorem bounding the running time of the algorithm.

Theorem 9.5 *Suppose that for $\varepsilon > 0$ and $\mathbb{P}[\mathcal{E}_i] \leq p_i$, $(1 + \varepsilon)\mathbf{p}$ satisfies Shearer's conditions. Then for any $t > 0$ and $D = \frac{2}{\varepsilon}(\log \frac{1}{q_\emptyset} + t)$,*

$$\mathbb{P}[\# \text{ of resamplings} \geq D] \leq e^{-t}.$$

Proof: Let $p'_i = (1 + \varepsilon)p_i$, so $\mathbf{p}' = (1 + \varepsilon)\mathbf{p}$ satisfies Shearer's conditions. By Lemma 9.4, we have

$$\mathbb{P}[\text{Algorithm resamples } \mathcal{I} = (I_1, \dots, I_t)] \leq \prod_{s=1}^t p^{I_s} = \frac{1}{(1 + \varepsilon)^{\sum_{s=1}^t |I_s|}} \prod_{s=1}^t (p')^{I_s}.$$

This gives us

$$\begin{aligned} \mathbb{P}[\# \text{ of resamplings} \geq D] &\leq \sum_{\mathcal{I}=(I_1, \dots, I_t), \sum_{s=1}^t |I_s| \geq D} \mathbb{P}[\text{Algorithm resamples } \mathcal{I}] \\ &\leq \frac{1}{(1 + \varepsilon)^D} \sum_{\mathcal{I}=(I_1, \dots, I_t) \in \text{Prop}} \prod_{s=1}^t (p')^{I_s} \leq \frac{1}{(1 + \varepsilon)^D} \frac{1}{q_\emptyset} \leq e^{-t}. \end{aligned}$$

\square

This already implies that under a slack of constant $\varepsilon > 0$, the expected number of resamplings is $O(\log(1/q_\emptyset))$, typically a polynomial quantity. However, one can go further and make a statement without assuming slack explicitly. It actually holds that for any point satisfying Shearer's conditions, there is some amount of slack, and the following lemma quantifies the slack in terms in Shearer's coefficients. (See [Harvey and Vondrak (2015)] for a proof.)

Lemma 9.6 *For any \mathbf{p} satisfying Shearer's conditions and $\varepsilon = \frac{q_\emptyset(\mathbf{p})}{2 \sum_{i=1}^n q_{\{i\}}(\mathbf{p})}$, $(1 + \varepsilon)\mathbf{p}$ still satisfies Shearer's conditions, and $q_\emptyset((1 + \varepsilon)\mathbf{p}) \geq \frac{1}{2}q_\emptyset(\mathbf{p})$.*

Substituting the above formula for ε in the statement of Theorem 10.5, we find that

$$\mathbb{P} \left[\# \text{ of resamplings} \geq \sum_{i=1}^n \frac{q_{\{i\}}}{q_\emptyset} \left(\log \frac{1}{q_\emptyset} + t \right) \right] \leq e^{-t}$$

for $t > 0$, so $\mathbb{E}[\# \text{ of resamplings}] = O(\sum_{i=1}^n \frac{q_{\{i\}}}{q_\emptyset} \log \frac{1}{q_\emptyset})$.

Under the LLL conditions, we have $\frac{q_{\{i\}}}{q_\emptyset} \leq \frac{x_i}{1-x_i}$ and $q_\emptyset \geq \prod_{i=1}^n (1-x_i)$. Also, it can be shown that $\log \frac{1}{q_\emptyset} \leq \sum_{i=1}^n \frac{q_{\{i\}}}{q_\emptyset}$ [Harvey and Vondrak (2015)]. This implies that the expected number of resamplings is bounded by $O((\sum_{i=1}^n \frac{x_i}{1-x_i})^2)$, or $O((\sum_{i=1}^n \frac{q_{\{i\}}}{q_\emptyset})^2)$, at most a square of the known bounds in the Moser-Tardos model.

References

- [Moser and Tardos (2010)] Robin Moser and Gabor Tardos. A constructive proof of the general Lovász Local Lemma. *Journal of the ACM* 57(2), 2010.
- [Kolipaka and Szegedy (2011)] Kashyap Kolipaka and Mario Szegedy. Moser and Tardos meet Lovasz. In *Proceedings of ACM-SIAM STOC*, 2011.
- [Achlioptas and Iliopoulos (2014)] Dimitris Achlioptas and Fotis Iliopoulos. Random walks that find perfect objects and the Lovasz local lemma. In *Proceedings of IEEE FOCS*, 2014.
- [Harvey and Vondrak (2015)] Nicholas Harvey and Jan Vondrak. An algorithmic proof of the Lovász Local Lemma via resampling oracles. In *Proceedings of IEEE FOCS*, 2015.