CSE525: Randomized Algorithms and Probabilistic Analysis

Lecture 11

Lecturer: Anna Karlin *Scribe: Cyrus Rashtchian, Tyler Rigsby*

Lovász Local Lemma

The Lovász Local Lemma is a clever technique to lower bound the probability that no events in some set occur. In particular, it shows that under certain conditions (weaker than independence), there is a positive probability that a set of “bad” events all do not occur. Usually, the only way to bound such a probability is to hope that either (i) the sum of the probabilities of the bad events is less than one (i.e. the union bound), or (ii) that all the events are independent and at least one has probability less than one. The Lovász Local Lemma is a clever combination of these cases.

Let’s start with two relevant definitions about the dependency and mutual independence of events.

- An event A is *mutually independent* of events $B_1, B_2, \ldots, B_k$ if for any subset $I \subseteq [1, \ldots, k]$, we have that $Pr(A \mid \bigcap_{j \in I} B_j) = Pr(\text{A})$ (that is, the occurrence of any subset of events that A is mutually independent with does not affect the probability of A).

- A dependency graph for events $E_1, E_2, \ldots, E_n$ is an undirected graph $G = (V, E)$ where $V = \{1, \ldots, n\}$ and the set of edges is defined so that E_i is mutually independent of $\{E_j | (i, j) \notin E\}$ (that is, every event E_i is mutually independent with its set of non-neighbors).

Lovász Local Lemma: If there exists a real number $0 \leq p < 1$ so that a set of events $E_1, E_2, ..., E_n$ satisfies

1. $Pr(E_i) < p \ \ \forall i$,
2. the max degree in the dependency graph is d,
3. and, $4dp \leq 1$,

then, $Pr(\bigcap_{i=1}^{n} \overline{E_i}) > 0$.

Proof To prove the lemma, we’ll show that $Pr(\bigcap_{i \in S} \overline{E}_i) > 0$ and $Pr(E_k | \bigcap_{i \in S} \overline{E}_i) \leq 2p \ \forall k$ by induction on $|S|$.

Basis: $|S| = 1$. Then,

$$Pr(\overline{E}_i) = 1 - Pr(E_i) > 1 - p > 0.$$

Case 1 – there is no edge (k, i) in the dependency graph. Then:

$$Pr(E_k \mid \overline{E_i}) = Pr(E_k) \leq p$$

Case 2 – $\exists$ an edge (k, i) in the dependency graph. Then:

$$Pr(E_k \mid \overline{E_i}) = \frac{Pr(E_k \bigcap \overline{E_i})}{Pr(\overline{E_i})} \leq \frac{p}{1-p} < 2p \ (\forall p < \frac{1}{4})$$

Inductive step: Suppose we have some set T such that $|T| < s$, where $Pr(\bigcap_{i \in T} \overline{E_i}) > 0$ and $Pr(E_k | \bigcap_{i \in T} \overline{E_i}) \leq 2p \ \forall k$. First, we show that $Pr(\bigcap_{i=1}^{s} \overline{E_i}) > 0$:

$$\begin{aligned} Pr(\bigcap_{i=1}^{s} \overline{E_i}) &= Pr(\overline{E_1}) \cdot Pr(\overline{E_2} \mid \overline{E_1}) \cdot Pr(\overline{E_3} \mid \overline{E_1} \wedge \overline{E_2}) \cdots Pr(\overline{E_s} \mid \overline{E_1} \wedge \overline{E_2} \wedge ... \wedge \overline{E_{s-1}}) \\ &= \prod_{i=1}^{s} \left(1 - Pr(E_i \mid \bigcap_{j=1}^{i-1} \overline{E_j}) \right) \\ &\geq \prod_{i=1}^{s} (1 - 2p) \\ &> 0 \end{aligned}$$

Now, we just need to show that $Pr(E_k \mid \bigcap_{i \in S} \overline{E_i}) \leq 2p$. Consider splitting S into two subsets S_1 and S_2. If $S_2 = S$, then $Pr(E_k \mid \bigcap_{i \in S_2} \overline{E_i}) = Pr(E_k) \leq p$, by the inductive hypothesis. So, we'll assume that $S_2 \subset S$. We define three sets

$$F_S = \bigcap_{i \in S} \overline{E_i}, \qquad F_{S_1} = \bigcap_{i \in S_1} \overline{E_i}, \qquad F_{S_2} = \bigcap_{i \in S_2} \overline{E_i}.$$

Then, we see that

$$\begin{aligned} Pr(E_k | F_S) &= \frac{Pr(E_k \wedge F_S)}{Pr(F_S)} \\ &= \frac{Pr(E_k \wedge F_{S_1} | F_{S_2}) Pr(F_{S_2})}{Pr(F_{S_1} | F_{S_2}) Pr(F_{S_2})} \\ &= \frac{Pr(E_k \wedge F_{S_1} | F_{S_2})}{Pr(F_{S_1} | F_{S_2})} \end{aligned}$$

where we can compute

$$\begin{aligned} Pr(E_k \wedge F_{S_1} | F_{S_2}) &\leq Pr(E_k | F_{S_2}) \\ &\leq Pr(E_k) \\ &\leq p. \end{aligned}$$

Also,

$$\begin{aligned} Pr(F_{S_1} | F_{S_2}) &= 1 - Pr\left(\bigcup_{i \in S_1} E_i | F_{S_2} \right) \\ &\geq 1 - \sum_{i \in S_1} Pr(E_i | F_{S_2}) && \text{by the union bound} \\ &\geq 1 - \sum_{i \in S_1} 2p && \text{by the inductive hypothesis } (|S_2| < |S|) \\ &\geq 1 - 2dp \geq \frac{1}{2} && \text{by assumption} \end{aligned}$$

Thus, $Pr(E_k | E_S) \leq 2p$.

■

k-SAT

The first application of the Lovász Local Lemma that we will explore is k-SAT; given a k-SAT formula with n variables and m clauses, we will determine conditions under which we can guarantee that the formula has a satisfying assignment.

Theorem 1. *If no variable appears in more than $T = \frac{2^k}{4k}$ clauses, then the formula has a satisfying assignment.*

Proof Consider a random assignment where each variable is true with probability $\frac{1}{2}$ independently. Let E_i be the event that clause i is not satisfied.

- $p = Pr(E_i) = 2^{-k}$

- E_i is mutually independent of clauses that don't share variables

Thus, $d \leq kT = 2^{k-2}$, so $4dp \leq 2^2 \cdot 2^{k-2} \cdot 2^{-k} \leq 1$. Thus, by the Lovász Local Lemma, $Pr(\bigcap_{i=1}^{m} \overline{E_i}) > 0$– that is, there exists a satisfying assignment to the k-SAT instance.

■

Packet Routing

In this problem, we are given a undirected graph $G = (V, E)$, and also a set of n packets $p_1, \ldots, p_n$ that will be routed through G. Each packet p_i has a start vertex s_i, a destination vertex t_i, and a specific s_i, t_i path P_i. Vertices act as queues of packets, and packets move synchronously across edges. At every time step, at most one vertex can traverse each edge. Our goal is to develop a schedule dictating when each packet moves/waits, which minimizes the total time for all packets to get from their start to destination.

Let's define two parameters relevant to the efficiency of any schedule. The *dilation* is

$$d = \max_i \{ \text{ distance from } s_i \text{ to } t_i \ \}$$

and the *congestion* is

$$c = \max_e \{ \text{ number of paths that use edge } e \ \}.$$

Note that any schedule must take time $\Omega(\max(c, d))$ and we can easily find one with time $O(cd)$. We will show that there exists a schedule of time $O((c+d)(1+\alpha)^{\log^*(c+d)})$, where $\alpha > 1$ is a constant to be chosen, and $\log^*(c+d) \leq 6$ for all practical purposes (for all $c + d \leq 10000^{1000}$), so this is basically optimal.

Approach: To simplify the calculations, let's assume $d = \max(c, d)$, since our claimed solution is in terms of $(c + d) = O(\max(c, d))$. Our approach is that we will set an initial delay of at most αd for each packet, so that it will first wait this long, and then move along its path, one edge at each time step. We will simultaneously guarantee that if we pick the delays correctly, then each packet will not have to wait at all while moving along its path, and that all packets will be able to move concurrently.

Recursive solution: We will solve this problem recursively, by dividing the time steps $\{1, \ldots, (1+\alpha)d\}$ into phases of length $\log d$, and then solving the problems on each phase, and combining the solutions together. To pass to the recursive call, we will modify the packets' routes by supposing they start somewhere along their path, and then they move at most $\log d$ steps along their path. We then can easily piece together these intermediate start/destination vertices, since they will be defined so that the packet starts where it ended the phase before.

The remaining challenge then is to show that in any phase, there exists a schedule (i.e., a way to set the delays) so that the congestion along any edge is also at most $\log d$. If so, then we have a proper recursive call, because the total time, the dilation, and the congestion, have all become $\log d$. The base case of the recursion will be when there is only a single time step, and in that case since each edge has congestion 1, we just move the packets one edge ahead.

Schedule analysis: Before we prove the bound on the congestion in each phase, let's show how to prove the claimed result. At each step in the recursion, the total time needed grows by a factor of $(1+\alpha)$, and the initial time was at most $(c+d)$. To recurse down to a single time step, we need $\log^*(c+d)$ recursive calls. This proves our claim. Now back to the technical part.

Claim *In each phase of length* $\log d$*, there exists a set of delays for the packets so that the max congestion over any edge is at most* $\log d$*.*

Proof We proceed using the probabilistic method, namely showing that with positive probability under some distribution over the delays, the congestion won't be too big. We will sample a delay independently for each packet uniformly from $\{1, 2, \ldots, \alpha d\}$. Once the delays are set, we will recursively construct a schedule for each packet which has unit post-delay congestion, so we only have to worry about the total congestion of the phase.

Let's set up to use the Lovasz local lemma. Fix a phase and let A_e be the event that edge e has congestion greater than $\log d$. Since a packet's path is fixed post-delay, the chance that any p_i uses e at time any fixed time t is $1/(\alpha d)$. We bound $\mathbf{Pr}[A_e]$ using the Chernoff bound. The number of packets that use e is at most d, there at $\log d$ time steps in this phase, and the probability for each packet, in each time step, to use e is $1/(\alpha d)$, so

$$\mathbf{E}[\#\textit{edges using } e] \leq d \cdot \log d \cdot \frac{1}{\alpha d} = \frac{\log d}{\alpha}.$$

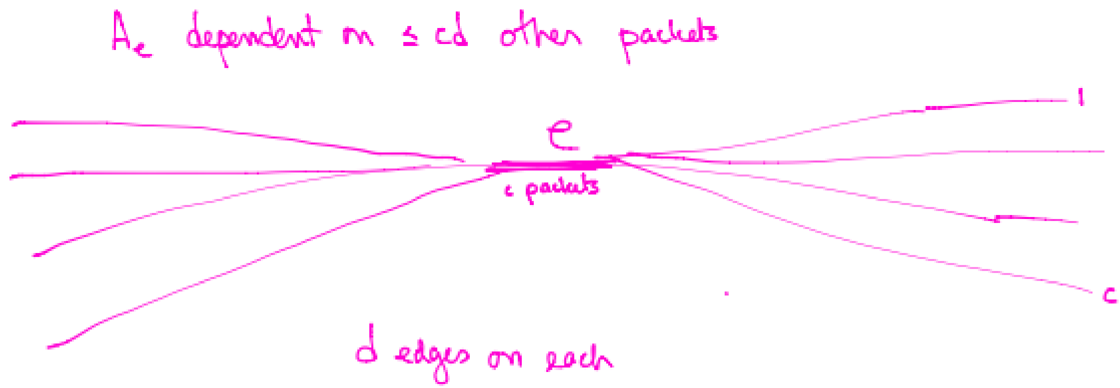
Then, by chernoff, with $\delta = \alpha - 1$ and $\mu = (\log d)/\alpha$,

$$\Pr[\#\textit{edges using } e \geq \log d] = \Pr[\#\textit{edges using } e \geq (1 + (\alpha - 1))\frac{\log d}{\alpha}] \leq \left(\frac{e}{\alpha}\right)^{\log d} \leq \frac{d^{-4}}{\alpha},$$

where the last inequality follows by setting $\alpha = e^{11}$ and assuming that $d \geq e^{11/6}$. Then, since there are at most αd phases, we have by the union bound that

$$\mathbf{Pr}[A_e] \leq d^{-3}.$$

Now the dependency graph. For edges e and f, the events A_e and A_f are independent unless the same packet traverses both e and f along its path. At most $cd \leq d^2$ other edges are dependent on A_e, since each path has length d, and the number of paths that overlap at e is at most c. Therefore, the conditions for the Lovasz local lemma are satisfied, and thus $\mathbf{Pr}[\bigcap_e \overline{A}_e] > 0$, that is there exists a set of initial delays so that each edge has congestion at most $\log d$. And as we argued above, this concludes the entire analysis.



$\blacksquare$

Exercises

1. **Generalized Lovasz Local Lemma.** Let $A_1, \ldots, A_n$ be a set of bad events, and let $D_i \subseteq \{A_1, \ldots, An\}$ denote the dependency set of A_i, i.e., A_i is mutually independent of all events not in D_i. If there exists a set of real numbers $x_1, \ldots, x_n \in [0, 1)$ such that

$$\mathbf{Pr}[A_i] \leq x_i \prod_{j \in D_i} (1 - x_j)$$

   for all i, then

$$\mathbf{Pr}[\bigcap_{i=1}^{n} \overline{A_i}] \geq \prod_{i=1}^{n} (1 - x_i) > 0.$$

   Note that if you set $x_i = \frac{1}{d+1}$ where d is the degree of the dependency graph, then you recover the symmetric version we proved in class. Also, note that the solution to this exercise is a fairly straightforward manipulation of the proof we saw in class.

2. **Structured bichromatic edge coloring.** Let K_n denote the complete (undirected) graph on n vertices. Show that if

$$4\binom{k}{2}\binom{n}{k-2}2^{1-\binom{k}{2}} \leq 1,$$

   then it is possible to color the *edges* of K_n with two colors so that it has no monochromatic K_k subgraph, that is, no clique of size k in the colored K_n with all $\binom{k}{2}$ edges assigned the same color. *Note that trying to color the graph in the case of $n = 6$ and $k = 3$ is an excellent puzzle.*