Randomness and Computation
or, “Randomized Algorithms”

Mary Cryan
School of Informatics
University of Edinburgh

The Lovász Local lemma

Definition (6.1)
A dependency graph for a set of events $E_1, \ldots, E_N$ is a graph $G = (V, E)$ such that $V = \{1, \ldots, N\}$ and for each $i = 1, \ldots, N$, the event $i$ is mutually independent with the events $\{E_j \mid (i, j) \notin E\}$. The degree of the dependency graph is the max degree vertex of $G$.

Theorem (6.11, Lovász Local Lemma)
Let $E_1, \ldots, E_N$ be a set of events and assume we know $p \in (0, 1)$, $d \in \mathbb{N}$ such that all the following conditions hold:

1. For all $i$, $\Pr[E_i] \leq p$;
2. The degree of the dependency graph on $\{E_1, \ldots, E_N\}$ is $\leq d$;
3. $4dp \leq 1$

Then

$$\Pr[\bigcap_{i=1}^N E_i] > 0.$$
SAT and k-SAT (standard probabilistic method)

Recall that in propositional logic, a Boolean variable $x_i$ can take on 0 or 1 values, a literal is either $x_i$ or $\overline{x}_i$, and for the set of variables $\{x_1, x_2, \ldots, x_n\}$ a SAT problem is any conjunction (AND) of a set of clauses, each individual clause being a disjunction (OR) of literals. For example,

$$\{x_4, x_7, \overline{x}_6\}, \{\overline{x}_1, x_6, x_5\}, \{x_1, x_2, \overline{x}_6\}, \{x_3, x_6, \overline{x}_2\}$$

is an instance of SAT. Since all clauses are of length 3, the one above is also an instance of 3-SAT.

Suppose we have $m$ clauses, with $k_i$ literals in the $i$th clause, $1 \leq i \leq m$. Then on a uniform random assignment of boolean values to the $n$ variables, the probability clause $i$ is satisfied is $(1 - 2^{-k_i})$.

$(2^k$ is the probability we would set all $k_i$ literals of this clause to be false.)

$k$-SAT with Lovász Local Lemma

Now consider the "bad events" $E_i$ to be the event where clause $i$ becomes unsatisfied, and consider the dependency graph.

**Theorem (6.13)**

If we have a $k$-SAT formula where no variable appears in more than $T = \frac{2^k}{4k}$ clauses, then that formula has some satisfying assignment.

**Proof** We assume a uniform random assignment to all the $x_i$ and hence $E_i$ is the event that all the $k$ variables get the "wrong" assignment. $Pr[E_i] \leq 2^{-k}$ for all $i$.

The event $E_i$ is mutually dependent of any $E_j$, where clause $i'$ shares no logical variables with clause $i$. For each of the variables in clause $i$, they may appear in $T = \frac{2^k}{4k}$ clauses, so taking all $k$ variables, there are at most $k \cdot T = \frac{2^k}{4} k$ clauses which share some variable(s) with clause $i$. So $d \leq \frac{2^k}{4}$.

Then $4d \cdot \frac{1}{2^k} \cdot 2^{-k} = 1$, and the LLL implies there is some assignment where none of the bad events occur (ie, all clauses are satisfiable).

LLL: can we de-randomize?

The only negative aspect of using LLL is that we don’t get an explicit randomized process linked to the existence result. So we don’t have a handle on how we might go about finding such an object.

There are ways to convert a LLL result into an explicit construction, but usually you need a lower dependency value.

We won’t cover this (see Sections 6.8, 6.10 of the book)
LLL: can we de-randomize?

The only negative aspect of using LLL is that we don't get an explicit randomized process linked to the existence result. So we don't have a handle on how we might go about finding such an object.

There are ways to convert a LLL result into an explicit construction, but usually you need a lower dependency value.

We won't cover this (see Sections 6.8, 6.10 of the book)

Notes

Reading
▶ Section 6.7 from the book.

Doing
▶ Tutorial sheet for 5th, 6th March (week 7).
▶ Coursework 2 specification has been released.