**CS6840: Advanced Complexity Theory** 24th Feb 2011

# Lecture 07 : Lovász Local Lemma

*Lecturer: Narayanaswamy N.S.* *Scribe: Princy Lunawat*

THEME: The focus of this lecture is the proof and applications of the Lovász Local Lemma which provides a useful tool to analyse the dependencies among events under which the probabilty of occurence of none of events is non zero.
LECTURE PLAN:In this lecture, we define the notion of a Dependency Graph for a given set of dependent events.We state and prove the Lovász Local Lemma which proves a lower bound on the probability of occurence of none of events where each event may have a different probability bound.We then state the symmetric version of the lemma for events with the same probability bound.We map the `KCNFSAT` problem to the Dependency Graph framework and apply the lemma to analyse the characteristics of the clauses of the problem which give a non-zero probability of satisfiability.Finally, we give the constructive version of the lemmausing randomization.

## 1 KCNFSAT

Given a boolean formula in $k$ variables and $m$ clauses in `CNF` , that is, $C_1 \wedge C_2 \wedge C_3 \ldots C_m$, we define a *bad* event $\mathcal{E}_i$ to be the event that $C_i$ is not satisfiable. Then,

$$\Pr[\mathcal{E}_i] = \Pr[C_i \text{ is not satisfiable}] = \frac{1}{2^k}$$

Our interest is to analyse the dependencies among the clauses of the formula in order to be able to come up with the probability that the formula is satisfiable. For example, if no two clauses $C_i$ and $C_j$ have a variable in common, that is, the clauses are mutually disjoint, then it is easy to observe that there is a unique satisfiable assignment for the formula. Essentially, we determine the conditions under which $\Pr\left[\bigwedge_{i=1}^{m} \bar{\mathcal{E}}_i\right] > 0$ , that is, all the clauses are satisfiable. Now, we define Dependency Graphs for the given set of events $\mathcal{E}_i$ for our analysis.

## 2 Dependency Graph

An undirected Graph $D(V, E)$ is the Dependency Graph for the set of events $\mathcal{E}_i$ defined above, when $V = \{\mathcal{E}_1, \mathcal{E}_2, \ldots \mathcal{E}_m\}$. The edges $E$ are defined implicitly. Let $N(i)$ (neighbour-

hood of $\mathcal{E}_i$ ) is the set of vertices adjacent to $\mathcal{E}_i$. That is,

$$\forall \mathcal{E}_j \in N(i), \{\mathcal{E}_i, \mathcal{E}_j\} \in E$$

Then, the graph $D$ satisifies the following:

$$\forall i, \forall S \subseteq V - N(i), \Pr\left[\mathcal{E}_i | \bigcap_{J \in S} \mathcal{E}_J\right] = \Pr\left[\mathcal{E}_i\right]$$

That is, the event $\mathcal{E}_i$ is independent of the events corresponding to the vertices of any subset $J$ of the set of vertices $S$ which are not in the neighbourhood of the vertex $\mathcal{E}_i$.

## 3 Lovász Local Lemma

We first state the assymetric version of the lemma where in, different probability bounds are allowed for different events.
Let $\{\mathcal{E}_1, \mathcal{E}_2, \ldots \mathcal{E}_m\}$ be a set of events in a probability space.If there exists an assignment of reals $\{x_1, x_2 \ldots x_m\}, x_i \in [0, 1)$ ,such that

$$\Pr\left[\mathcal{E}_i\right] \leq x_i \prod_{(i,j)} (1 - x_j) \tag{1}$$

where $\forall j, \mathcal{E}_j \in N(i)$, that is $\{\mathcal{E}_i, \mathcal{E}_j\} \in E$ where $E$ is the set of edges in the corresponding dependency graph $D(V, E)$. Then,

$$Pr\left[\bigwedge_{\mathrm{i=1}}^{m} \bar{\mathcal{E}}_i\right] > \prod_{\mathrm{i=1}}^{m} (1 - x_i) \tag{2}$$

*Proof. Claim:* $\forall S \subseteq \{\mathcal{E}_1, \mathcal{E}_2, \ldots \mathcal{E}_m\}, \forall i,\ \Pr\left[\mathcal{E}_i | \bigcap_{J \in S} \bar{\mathcal{E}_J}\right] \leq x_i$ .

*Proof.* For $|S| = 0$,

$$\Pr\left[\mathcal{E}_i | \bigcap_{J \in S} \bar{\mathcal{E}_J}\right] = \Pr\left[\mathcal{E}_i\right] \leq x_i$$

by equation 1. By induction hypothesis, let the claim be true for all $S'$ such that $|S'| < r$. Let $|S| = r$, we partition $S$ into two subsets $S_1$ and $S_2$ such that in the dependency graph,

for a given vertex $\mathcal{E}_i$, $S_1 = S \cap N(i)$ and $S2 = S \setminus N(i)$.Then,

$$\begin{aligned}
& \Pr\left[\mathcal{E}_i | \bigcap_{J\in S} \bar{\mathcal{E}_J}\right] \\
= & \Pr\left[\mathcal{E}_i | \bigcap_{J\in S_1} \bar{\mathcal{E}_J} \bigcap_{J\in S_2} \bar{\mathcal{E}_J}\right] \\
= & \Pr\left[\mathcal{E}_i \bigcap_{J\in S_1} \bar{\mathcal{E}_J} | \bigcap_{J\in S_2} \bar{\mathcal{E}_J}\right] / \Pr\left[\bigcap_{J\in S_1} \bar{\mathcal{E}_J} | \bigcap_{J\in S_2} \bar{\mathcal{E}_J}\right] \text{[1]} \\
\leq & \Pr\left[\mathcal{E}_i | \bigcap_{J\in S_2} \bar{\mathcal{E}_J}\right] / \Pr\left[\bigcap_{J\in S_1} \bar{\mathcal{E}_J} | \bigcap_{J\in S_2} \bar{\mathcal{E}_J}\right] \\
= & \Pr\left[\mathcal{E}_i\right] / \Pr\left[\bigcap_{J\in S_1} \bar{\mathcal{E}_J} | \bigcap_{J\in S_2} \bar{\mathcal{E}_J}\right] \text{[2]} \\
\leq & x_i \prod_{(i,j)} (1 - x_j) / \Pr\left[\bigcap_{J\in S_1} \bar{\mathcal{E}_J} | \bigcap_{J\in S_2} \bar{\mathcal{E}_J}\right] \ \ldots \text{ from equation 1}
\end{aligned}$$

So we have,

$$\Pr\left[\mathcal{E}_i | \bigcap_{J\in S} \bar{\mathcal{E}_J}\right] \leq x_i \prod_{(i,j)} (1 - x_j) / \Pr\left[\bigcap_{J\in S_1} \bar{\mathcal{E}_J} | \bigcap_{J\in S_2} \bar{\mathcal{E}_J}\right] \tag{3}$$

Let $|S_1| = t, t \leq |S|$. Now consider the denominator of equation 3,

$$\begin{aligned}
& = \Pr\left[\bigcap_{J\in S_1} \bar{\mathcal{E}_J} | \bigcap_{J\in S_2} \bar{\mathcal{E}_J}\right] \\
& = \Pr\left[\bigcap_{i=1}^{t} \bar{\mathcal{E}_{J_i}} | \bigcap_{J\in S_2} \bar{\mathcal{E}_J}\right] \\
& = \Pr\left[\bigcap_{i=1}^{t-1} \bar{\mathcal{E}_{J_i}} | \bigcap_{J\in S_2} \bar{\mathcal{E}_J} \cap \bar{\mathcal{E}_{J_t}}\right] \times \Pr\left[\bar{\mathcal{E}_{J_t}} | \bigcap_{J\in S_2} \bar{\mathcal{E}_J}\right] \times \left(1 - \Pr\left[\bar{\mathcal{E}_{J_t}} | \bigcap_{J\in S_2} \bar{\mathcal{E}_J}\right]\right)
\end{aligned}$$

By induction hypothesis on the first term of the above product,$(t - 1 < |S|)$

$$\begin{aligned}
& \geq (1 - x_{J_1}) \times (1 - x_{J_2}) \ldots \times (1 - x_{J_t}) \\
& = \prod_{i=1}^{t} (1 - x_{J_i})
\end{aligned}$$

Plugging the above value in equation 3, we have,

$$\Pr\left[\mathcal{E}_i | \bigcap_{J\in S} \bar{\mathcal{E}_J}\right] \leq x_i \prod_{(i,j)} (1 - x_j) / = \prod_{i=1}^{t} (1 - x_{J_i})$$

[2] $\Pr[A|B \cap C] = \frac{Pr[A \cap B \cap C]}{Pr[B \cap C]} = \frac{Pr[A \cap B|C] \times Pr[C]}{Pr[B \cap C]} = \frac{Pr[A \cap B|C]}{Pr[B|C]}$

[2] All events is set $S_2$ are independent of $\mathcal{E}_i$($S_2$ is the set of non-neighbours of $\mathcal{E}_i$).

Observe that both $x_j$ in the numerator and $x_{J_i}$ in the denominator represent an event in the neighbourhood of $\mathcal{E}_i$. Hence,

$$\Pr\left[\mathcal{E}_i | \bigcap_{J\in S} \bar{\mathcal{E}_J}\right] \leq x_i$$

Hence the claim. □

We now prove the lemma using the above claim.

$$\begin{aligned}
&\Pr\left[\bigwedge_{i=1}^{m} \bar{\mathcal{E}_i}\right] \\
&= \Pr\left[\bar{\mathcal{E}_m} | \bigwedge_{i=1}^{m-1} \bar{\mathcal{E}_i}\right] \times \Pr\left[\bigwedge_{i=1}^{m-1} \bar{\mathcal{E}_i}\right] \\
&= \left(1 - \Pr\left[\mathcal{E}_m | \bigwedge_{i=1}^{m-1} \bar{\mathcal{E}_i}\right]\right) \times \Pr\left[\bigwedge_{i=1}^{m-1} \bar{\mathcal{E}_i}\right]
\end{aligned}$$

From the above claim,

$$\begin{aligned}
&\geq (1 - x_m) \Pr\left[\bigwedge_{i=1}^{m-1} \bar{\mathcal{E}_i}\right] \\
&\geq \prod_{i=1}^{m} (1 - x_i)
\end{aligned}$$

(By repeated application of the same steps as above with $m = m - 1$.)

□

# 4 Symmetric Version of the Lemma

We now consider a symmetric version of the lemma in which each event has the same probability bound $p$,that is, each event $\mathcal{E}_i$ occurs with probabilty $p$.Then the lemma states that if each event is independent of all the other events except for at most $\Delta$ of them, that is, if in the corresponding dependency graph $D(V, E)$,
$\forall i, deg(\mathcal{E}_i) \leq \Delta$
and the following relation holds,
$ep(\Delta + 1) \leq 1$
Then there is a non-zero probability that none of the events will occur. That is,
$\Pr\left[\bigwedge_{i=1}^{m} \bar{\mathcal{E}_i}\right] > 0$

## 5 Dependency Graph for KCNFSAT

We now define the dependency graph for the KCNFSAT problem discussed before, and use Lovász Local Lemma to analyse the satisfiability of a given formula.

Given a formula in $k$ variables, of the form $C_1 \wedge C_2 \wedge \ldots \wedge C_n$, let $\mathcal{E}_i$ be the event that the clause $C_i$ is not satisfied.The dependency graph $D(V, E)$ is defined as follows:

$V = \mathcal{E}_1, \mathcal{E}_2 \ldots \mathcal{E}_m$
$\{\mathcal{E}_i, \mathcal{E}_j\} \in E$ iff $C_i$ and $C_j$ have atleast one variable in common.

If there is no edge between two vertices in $D$, we conclude that the corresponding events are independent of each other.Mathematically,

$$\Pr\left[\mathcal{E}_i | \bigcap_{J \neq N(i)} \bar{\mathcal{E}}_i\right] = \Pr\left[\mathcal{E}_i\right]$$

We know that each event $\mathcal{E}_i$ is bounded by a probability $p$, such that,

$$p = \Pr\left[\mathcal{E}_i\right] = \Pr\left[C_i \text{ is not satisfiable}\right] = \frac{1}{2^k}$$

The formula is satisfiable iff all of the bad events are avoided, that is , $\Pr\left[\bigcap_{i=1}^{m} \bar{\mathcal{E}}_i\right] > 0$.By Lovász Local Lemma, the following condition has to be fullfilled for the formula to be satisfiable,

$$ep\left(\Delta + 1\right) \leq 1$$

For $p = \frac{1}{2^k}$,

$$\begin{aligned} \frac{e\left(\Delta + 1\right)}{2^k} &\leq 1 \\ \left(\Delta + 1\right) &\leq \frac{2^k}{e} \\ \Delta &\leq \frac{2^k}{e} - 1 \end{aligned}$$

Hence, for a formula to be satisfiable, the degree of each vertex in the dependency graph has to be atmost $\frac{2^k}{e} - 1$. therefore, each variable can occur in atmost $\frac{1}{k}\frac{2^k}{e} - 1$ clauses.

## 6 Constructive Version of Lovász Local Lemma

Observe that this lemma is nonconstructive in the sense that it gives no method of determining an explicit element of the probability space in which no bad event occurs. We give a

randomized algorithm that finds such an element. Before that, Let us first introduce some concepts that are used in the algorithm.
Each $\mathcal{E}_i$ is a subset of a universe of bad events, each is given by $p_1, p_2 \dots p_n$ , $p_i \in \{0, 1\}$. When a assignment of these is 1, then a bad event $\mathcal{E}_i$ occurs.An assignment is said to avoid a bad event $\mathcal{E}_i$ if $\mathcal{E}_i$ is true (or occurs) under this assignment. The following randomized algorithm provides us with a constructive version of the lemma.

- Assign random values to $p_1, p_2 \dots p_n$. If it avoids each of the bad events $\mathcal{E}_i, \forall i$. Output the assignment and Halt.
- Else resample $p_1, p_2 \dots p_m$ and repeat from the above step.

Expected number of resamples to find a satisfying assignment(that is, the one which avoids all the bad events) is given by $\sum_{i=1}^{m} \frac{x_i}{1-x_i}$ where $m$ is the number of events. section3SAT A problem instance of (3SAT) consists of a formula in `3++CNF`. We give a randomized algorithm to check the satisfiability of the formula and the corresponding analysis.

- Assign random values from $0, 1$ to $x_1, x_2 \dots x_n$.
- If the chosen assignment satisifies the formula, then HALT.
- Else, for a clause which is violated in the formula,pick one of the variables in the clause with probability $\frac{1}{3}$ and flip the assignment.
- Repeat the above steps $3n$ times.

*Analysis:*

Here we present the intuition to analyse the above algorithm by considering a *Random Walk* on the chain of assignments. Let our random choice is $x_1, x_2 \dots x_n$ which is at a hamming distance(no. of bit flips) of $j$.

The following chain pictorially represents the hamming distance of the random solution form the hypothetical satisfying solution.

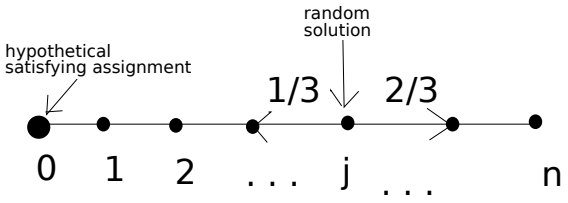


**Figure 1**: Random Walk Chain

With a probability of over $\frac{1}{3}$, we flip the right variable's value in a violated clause and move 1 unit closer to the hypothetical satisfying solution, that is the hamming distance between the two decreases by 1.Similarly, with probability atmost $\frac{2}{3}$, we flip the wrong variable's value in a violated clause and move 1 unit away from the hypothetical satisfying solution.A seemingly correct(but wrong!) analysis would be to solve for the following recursion:

$$\Pr(J) = \frac{1}{3}\Pr(J-1) + \frac{2}{3}\Pr(J-1)$$

Here, it is crucial to observe that the algorithm allows you to take a total of $3n$ steps (towards or away from the satisfying solution).Without loss of generality, let us assume that a 0corresponds to a left move from $j$ and a 1corresponds to a right move from $j$. The task now is simply to count the no. of $3n$ lenght binary strings that take you from position $j$ on the chain to position 0.Let this number be $k$, then probability of success is $\frac{k}{2^n}$ which turns out to be $\frac{1}{2^j}$.