Uniform Sampling Through The Lovasz Local Lemma

Yusuf Hakan Kalayci

Bogazici University

yhkalayci@gmail.com

October 6, 2018
Overview

1. Introduction
   - Motivation
   - The Lovasz Local Lemma
   - Algorithmic Lovasz Local Lemma

2. Non-uniform Example

3. Making The Algorithm Uniform
   - Restricted Framework
   - Generalization of the Algorithm
Powerful combinatorial tool for concluding existence

Goal: concluding existence or finding a combinatorial object devoid of undesired properties.

\( A_1, A_2 \cdots A_N \) are independent bad events and \( P[A_i] \in (0,1) \)

Obviously in the previous case, avoiding all the bad events is possible.

Question: Can we generalize this notion for limited or structured dependency?

Answer: Yes, in 1975, Laszlo Lovasz and Paul Erdos answered this question by the Lovasz Local Lemma.
The Lovász Local Lemma

Notations:

- Let $\mathcal{A} = \{A_1, A_2, \ldots, A_N\}$ be the set of $N$ bad events in arbitrary probability space where $\mathbb{P}[A_i] \in (0, 1)$.

- Let $G = (V, E)$ be a directed graph (digraph) where $V = \{1, 2, \ldots, N\}$ and $E$ is the set of edges.

- $A_1, A_2, \ldots, A_N$ are bad events and $\mathbb{P}[A_i] \in (0, 1)$.

- Here there is a one-to-one correspondence between $V$ and $\mathcal{A}$.

- We call graph $G$ as **dependency digraph** or simply **dependency graph** of event set $\mathcal{A}$ if for each $1 \leq i \leq N$ the event $A_i$ is mutually independent of all the events $\{A_j : (i, j) \notin E\}$. 

Let $A$ be the set of bad events and $G$ be the dependency graph. Also, suppose that there are real numbers $x_1, x_2, \cdots x_N$ such that $0 \leq x_i < 1$ satisfying

$$\mathbb{P}[A_i] \leq x_i \prod_{(i,j) \in E} (1 - x_j) \text{ for all } 1 \leq i \leq N. \text{ Then}$$

$$\mathbb{P}\left[\bigcap_{i=1}^{N} A_i^c\right] \geq \prod_{i=1}^{N}(1 - x_i).$$

In particular, with positive probability no event $A_i$ holds.
Corollary (The Lovasz Local Lemma: Symmetric Case)

Let $\mathcal{A}$ be the set of bad events. Suppose that each event $A_i$ is mutually independent of a set of all the other events $A_j$ but at most $d$, and $\mathbb{P}[A_i] \leq p$ for all $1 \leq i \leq N$. If

$$ep(d + 1) \leq 1$$

then $\mathbb{P}[\cap_{i=1}^{N} A_i^c] > 0$. 
Question: Can we find an instance of desired object?

Answer: For a subset of problems, the answer is ”Yes”!

From early 1990s by Beck, Alon, Moser, Tardos and many others

In 2010, groundbreaking work of Moser and Tardos give constructive proof for a large class of problems.
Algorithmic Lovasz Local Lemma

Notation:

- Let \( \{X_1, X_2 \cdots X_M\} \) be set of independent random variables, call them variable.
- Let \( \mathcal{A} = \{A_1, A_2 \cdots A_N\} \) be set of bad events.
- Let each \( A_i \) be determined by a subset \( \text{var}(A_i) \) of variables.
- For the dependency graph \( G = (V, E) \), \( V = \{1, 2 \cdots N\} \) will be the set of nodes and \((i, j) \in E \) if \( \text{var}(A_i) \cap \text{var}(A_j) \neq \emptyset \).
- In other words, event \( A_i \) and \( A_j \) are dependent if they share at least one variable.
Algorithm 1 Moser-Tardos Algorithm

1: Draw independent samples of all variables $X_1, \ldots, X_M$ from their respective distributions.
2: While at least one $A_i$ holds, uniformly at random pick one of such $A_i$ and resample all variables in $\text{var}(A_i)$.
3: Output the current assignment.

Theorem (Main Result of Moser Tardos)

Under the condition of the LLL, the expected number of resampling steps in Algorithm 1 is at most $\sum_{i=1}^{N} \frac{x_i}{1-x_i}$. 
Algorithm 2 Parallel Moser-Tardos Algorithm

1: Draw independent samples of all variables $X_1, \cdots, X_M$ from their respective distributions.
2: While at least one $A_i$ holds, find the independent set $I$ of occurring $A_i$’s and independently resample all variables in $\bigcup_{i \in I} var(A_i)$.
3: Output the current assignment.

It is nothing but resampling multiple non-interfering events at the same time.
Suppose we have a hypergraph $H = (V, E)$ where $V = \{1, 2, 3, 4\}$ and $E = \{E_1 = \{1, 2\}, E_2 = \{2, 3, 4\}\}$. Our aim is to find a 2-coloring of nodes where none of the hyperedges are monochromatic.
Consider the behavior of the algorithm as a Markov Chain.

States:
- 0 : 0000 corresponds $X_1 = X_2 = X_3 = X_4 = 0$.
- 1 : 0001 corresponds $X_1 = 1$ and $X_2 = X_3 = X_4 = 0$. and so on.

States 0010, 0101, 0110, 1001, 1010, 1101 are absorption states since none of them have monochromatic hyperedge.

States 0011, 0111, 0100, 1000, 1011, 1100 makes $E_1$ monochromatic $\rightarrow$ resample the variables $X_1, X_2$.

States 0001, 1110 makes $E_2$ monochromatic $\rightarrow$ resample the variables $X_2, X_3, X_4$.

The states 0000 and 1111 makes both of the bad events satisfied then the algorithm uniformly select one of those events and sample variables associated with it.
Non-uniform Example

\[
P = \begin{bmatrix}
\frac{3}{16} & \frac{1}{8} & \frac{3}{16} & \frac{1}{8} & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0

0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0

0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0

\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0

0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0

0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0

0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0

0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0

0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0

0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0

0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0

\frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & 0

0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0
\end{bmatrix}
\]
Non-uniform Example

Theorem

Let \( h^k = (h^k_i : i \in S) \) be the hitting probabilities to state \( k \) when starting from state \( i \) where \( S \) is the state space of Markov Chain. Then, \( h^k \) is the minimal non-negative solution to the system of linear equations

\[
h^k_k = 1
\]
\[
h^k_i = \sum_{j \in S} P_{ij} h^k_j \quad \text{for} \quad i \neq k
\]

Approximated solution to the linear equation system:

\[
\begin{bmatrix}
0 & 0 & 0.142 & 0 & 0 & 0.178 & 0.178 & 0 & 0 & 0.178 & 0.178 & 0 & 0 & 0.142 & 0 & 0
\end{bmatrix}
\]

In other words, chain is more likely to end up with states 0101, 0110, 1001, 1010 than 0010, 1101.
Recent study of Guo, Jerrum and Liu showed that when some restrictions imposed on the probability space, the algorithm can produce output uniformly.

**Condition**

\[ \text{If } (i,j) \in E, \text{ then } \mathbb{P}[A_i \cap A_j] = 0; \text{ namely the two events } A_i \text{ and } A_j \text{ are disjoint if they are dependent.} \]
Definitions:

- Execution log: \( l_1, l_2 \cdots l_t \)
- Resampling table: \( \{X_{i,1}, X_{i,2} \cdots \} \)

\[
\begin{array}{c|cccc}
X_1 & X_{1,1} & X_{1,2} & X_{1,3} & \cdots \\
X_2 & X_{2,1} & X_{2,2} & X_{2,3} & \cdots \\
X_3 & X_{3,1} & X_{3,2} & X_{3,3} & \cdots \\
X_4 & X_{4,1} & X_{4,2} & X_{4,3} & \cdots \\
X_5 & X_{5,1} & X_{5,2} & X_{5,3} & \cdots \\
\end{array}
\]

- Value pointers:

\[
\hat{J}_{i,t+1} = \begin{cases} 
\hat{J}_{i,t} + 1 & \text{if } \exists k \in I_t \text{ such that } X_i \in A_k \\
\hat{J}_{i,t} & \text{otherwise}
\end{cases}
\]

- Active values vector: \( \sigma_t : \{X_{i,j_i,t} : 1 \leq i \leq M \} \)
- Included neighbours: \( \Gamma^+(.) \) denotes the set of all neighbours of \( I \) unioned with \( I \) itself.
Lemma

Suppose given condition holds. Given any log \( S = S_1, S_2, \cdots, S_k \) of length \( k \geq 1 \), \( \sigma_{k+1} \) is a random sample from the product distribution conditioned on none of the bad events \( A_i \) occurring where \( i \notin \Gamma^+(S_k) \).

- Suppose the algorithm runs in two different sampling tables \( M \) and \( M' \) where the only difference is active values vector at time \( t + 1 \) namely \( \sigma_t \) and \( \sigma'_t \).
- Aim: Showing the two runs produces the same execution log until time \( t \).
- Result will follow from here since probability of having any active values vector at time \( t + 1 \) is proportional to its probability in original space.
- **Condition** allows us to conclude that in the history none of the resamplings interfere each other and log is preserved for two running.
Definitions:

- Let $Bad(\sigma)$ be the set of occurring bad events when the values of the variables are $\sigma$.
- Let $\partial S$ be the boundary of $S$; that is, $\partial S = \{i : i \notin S \text{ and } \exists j \in S \text{ such that } (i, j) \in E\}$.
- $var(S) := \bigcup_{i \in S} var(A_i)$.
- Let $\sigma_S$ be the partial assignment of $\sigma$ restricted to $var(S)$.
- We write $A_i \cap \sigma_S = \emptyset$ if $var(A_i) \cap var(S) = \emptyset$ or $A_i$ is disjoint from the partial assignment $\sigma_S$. 
Algorithm 3 Select the resampling set $\text{Res}(\sigma)$ under an assignment $\sigma$

1: Let $R = \text{Bad}(\sigma)$, which is the set of events that will be resampled. Let $N = \emptyset$, which is the set of events that will not be resampled.
2: While $\partial R \setminus N \neq \emptyset$, go through $i \in \partial R \setminus N$; if $A_i \cap \sigma_R \neq \emptyset$, add $i$ into $R$, otherwise add $i$ into $N$.
3: Output $R$ and call $\text{Res}(\sigma)$.

Algorithm 4 General Partial Rejection Sampling of Guo, Jerrum and Liu

1: Draw independent samples of all variables $X_1, \cdots, X_n$ from their respective distributions.
2: While at least one bad event occurs under the current assignment $\sigma$, use Algorithm 5 to find $\text{Res}(\sigma)$. Resample independently all variables in $\text{var}(\text{Res}(\sigma))$.
3: When none of the bad events holds, output the current assignment.
References

Robin A. Moser and Gabor Tardos (2010)
A constructive proof of the general Lovasz Local Lemma.
*J. ACM 57*(2).

Heng Guo, Mark Jerrum, Jingcheng Liue
Uniform Sampling Through The Lovasz Local Lemma
https://arxiv.org/abs/1611.01647

Noga Alon, Joel H. Spencer
The Probabilistic Method
The End