On the Algorithmic Lovász Local Lemma

Simon Grünbacher

11.4.2024

Motivation

- Let $\mathcal{A}$ be a finite collection of *bad* events in a probability space Ω.
- We want to prove that $\mathbb{P}(\bigcap_{A \in \mathcal{A}} \overline{A}) > 0$.
- **Case 1:** Events $\mathcal{A}$ independent and $\mathbb{P}(A) < 1$ for $A \in \mathcal{A}$
- $\implies \mathbb{P}(\bigcap_{A \in \mathcal{A}} \overline{A}) = \prod_{A \in \mathcal{A}} (1 - \mathbb{P}(A)) > 0$
- **Case 2:** Events not independent but $\sum_{A \in \mathcal{A}} \mathbb{P}(A) < 1$
- $\implies \mathbb{P}(\bigcap_{A \in \mathcal{A}} \overline{A}) \geq 1 - \sum_{A \in \mathcal{A}} \mathbb{P}(A) > 0$
- **Question:** What if we have something in between?
- **Assumption:** Every $A \in \mathcal{A}$ is determined by a finite set of independent random variables $\mathsf{vbl}(A) \subseteq \mathcal{V}$.
- Define $\Gamma(A) := \{B \in \mathcal{A} \mid B \neq A \wedge \mathsf{vbl}(B) \cap \mathsf{vbl}(A) \neq \emptyset\}$.

The Local Lemma

Theorem (László Lovász, Paul Erdős '75)

Let $\mathcal{A}$ be a finite collection of events as before. Assume that we have $x : \mathcal{A} \to [0,1)$ such that for all $A \in \mathcal{A}$ we have

$$\mathbb{P}(A) \leq x(A) \prod_{B \in \Gamma(A)} (1 - x(B)).$$

Then we have $\mathbb{P}(\bigcap_{A \in \mathcal{A}} \overline{A}) \geq \prod_{A \in \mathcal{A}} (1 - x(A)) > 0$.

Intuition: Easy to find $(x(A))_{A \in \mathcal{A}}$ if

- events *unlikely* ($\mathbb{P}(A)$ small) or
- events *close to independent* ($\#\Gamma(A)$ small).

Special Cases

- Union bounds:
- Assume $\sum_{A \in \mathcal{A}} \mathbb{P}(A) < \frac{1}{4}$.
- Choose $x(A) := 2\mathbb{P}(A)$.
- Then

$$\mathbb{P}(A) \leq \frac{1}{2} x(A) \leq x(A)(1 - \sum_{B \in \mathcal{A}} x(B)) \leq x(A) \prod_{B \in \mathcal{A}} (1 - x(B)).$$

- $\implies$ union bound can be almost reproduced
- Independence:
- Assume $\mathrm{vbl}(A) \cap \mathrm{vbl}(B) = \emptyset$ for $A \neq B$.
- Pick $x(A) = \mathbb{P}(A)$.
- Then $\mathbb{P}(A) \leq x(A)$.

Example

- Let $\Phi := C_1 \wedge \cdots \wedge C_n$ be a CNF Formula.
- $\mathcal{A} := \{\overline{C_1}, \ldots, \overline{C_n}\}$
- **Assumption:** Each clause depends on exactly 5 variables.
- **Assumption:** $|\Gamma(\overline{C_i})| \leq 11$.
- **Claim:** Φ is satisfiable.
- Let $x(\overline{C_i}) := \frac{1}{11}$ and assume $\mathbb{P}(X = \top) = \frac{1}{2}$ for $X \in \mathcal{V}$.
- We then have

$$2^{-5} = \mathbb{P}(\overline{C_i}) \leq \frac{1}{11}(1 - \frac{1}{11})^{11} \leq x(\overline{C_i}) \prod_{C \in \Gamma(\overline{C_i})} (1 - x(C)).$$

- **Question:** How can we find a satisfying assignment?

The Algorithm

```
for v ∈ 𝒱 do
    x_v ← a random sample of v
end for
while not all events avoided by x do
    A ← some event that holds under (x_v)_{v∈𝒱}
    for v ∈ vbl(A) do
        x_v ← a random sample of v
    end for
end while
return (x_v)_{v∈𝒱}
```

The Theorem

Theorem (Robin Moser, Gábor Tardos '09)

Let $\mathcal{A}$ be a collection of events that satisfies the conditions of the Lovász Local Lemma. Then the expected number of loop iterations is at most

$$\sum_{A \in \mathcal{A}} \frac{x(A)}{1 - x(A)}.$$

Note: Running twice as long, we succeed with probability $\frac{1}{2}$. Restarting k times, we succeed with probability $1 - 2^{-k}$.

A Proof Sketch

- Every resampling step happens for a unique *reason* τ, based on earlier random choices.
- We can use the parameters $(x(A))_{A \in \mathcal{A}}$ to define a *random reason generator* G_A.
- G_A returns a reason τ for resampling A with probability p_τ.
- $\mathbb{P}(A \text{ gets resampled for reason } \tau) \leq \frac{x(A)}{1-x(A)} p_\tau$
- We now have

$$\begin{aligned}
&\mathbb{E}(\# \text{ times } A \text{ gets resampled}) \\
&\quad = \sum_\tau \mathbb{P}(A \text{ gets resampled for reason } \tau) \\
&\quad \leq \frac{x(A)}{1-x(A)} \sum_\tau p_\tau \\
&\quad \leq \frac{x(A)}{1-x(A)}
\end{aligned}$$

Witness Trees

Definition

A *witness tree* τ is a rooted tree such that every vertex $v \in V(\tau)$ has a *label* $[v] \in \mathcal{A}$. We call τ *proper* if for all $v \in V(\tau)$, the children of v have distinct labels.

Witness Trees

Definition

A *witness tree* τ is a rooted tree such that every vertex $v \in V(\tau)$ has a *label* $[v] \in \mathcal{A}$. We call τ *proper* if for all $v \in V(\tau)$, the children of v have distinct labels.

- Let $A_1, \ldots$ be the events that get resampled by the algorithm.
- For every $n \in \mathbb{N}$, we define a witness tree τ_n:
- Let $\tau_n^{(n)}$ be a root node labelled A_n.
- For $i = n-1, \ldots, 1$ distinguish two cases:
- If there is $v \in V(\tau_n^{(i+1)})$ with $\mathrm{vbl}([v]) \cap \mathrm{vbl}(A_i) \neq \emptyset$:
- Add an A_i-labelled child to the maximum depth v to obtain $\tau_n^{(i)}$.
- Else, set $\tau_n^{(i)} := \tau_n^{(i+1)}$.
- Set $\tau_n := \tau_n^{(1)}$.

Witness Trees: Example

$$(a \vee b) \wedge (\neg a \vee c) \wedge (b \vee \neg c \vee d) \wedge \neg d \wedge e$$

a	b	c	d	e	C
0	0	0	1	0	e

e

Witness Trees: Example

$$(a \vee b) \wedge (\neg a \vee c) \wedge (b \vee \neg c \vee d) \wedge \neg d \wedge e$$

a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$

$\neg d$

Witness Trees: Example

$$(a \vee b) \wedge (\neg a \vee c) \wedge (b \vee \neg c \vee d) \wedge \neg d \wedge e$$

a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$
0	0	0	$\underline{0}$	0	e

$$\begin{array}{c} e \\ | \\ \mathrm{e} \end{array}$$

Witness Trees: Example

$$(a \vee b) \wedge (\neg a \vee c) \wedge (b \vee \neg c \vee d) \wedge \neg d \wedge e$$

a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$
0	0	0	$\underline{0}$	0	e
0	0	0	0	$\underline{1}$	$a \vee b$

$a \vee b$

Witness Trees: Example

$$(a \vee b) \wedge (\neg a \vee c) \wedge (b \vee \neg c \vee d) \wedge \neg d \wedge e$$

a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$
0	0	0	$\underline{0}$	0	e
0	0	0	0	$\underline{1}$	$a \vee b$
$\underline{1}$	$\underline{0}$	0	0	1	$\neg a \vee c$

$$\neg a \vee c$$

$$a \vee b$$

Witness Trees: Example

$$(a \vee b) \wedge (\neg a \vee c) \wedge (b \vee \neg c \vee d) \wedge \neg d \wedge e$$

a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$
0	0	0	$\underline{0}$	0	e
0	0	0	0	$\underline{1}$	$a \vee b$
$\underline{1}$	$\underline{0}$	0	0	1	$\neg a \vee c$
$\underline{0}$	0	$\underline{1}$	0	1	$a \vee b$

$$\begin{array}{c} a \vee b \\ | \\ \neg a \vee c \\ | \\ a \vee b \end{array}$$

Witness Trees: Example

$$(a \vee b) \wedge (\neg a \vee c) \wedge (b \vee \neg c \vee d) \wedge \neg d \wedge e$$

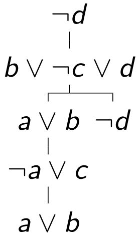
a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$
0	0	0	$\underline{0}$	0	e
0	0	0	0	$\underline{1}$	$a \vee b$
$\underline{1}$	$\underline{0}$	0	0	1	$\neg a \vee c$
$\underline{0}$	0	$\underline{1}$	0	1	$a \vee b$
$\underline{1}$	$\underline{0}$	1	0	1	$b \vee \neg c \vee d$

- $b \vee \neg c \vee d$
  - $a \vee b$
    - $\neg a \vee c$
      - $a \vee b$
  - $\neg d$

Witness Trees: Example

$$(a \lor b) \land (\neg a \lor c) \land (b \lor \neg c \lor d) \land \neg d \land e$$

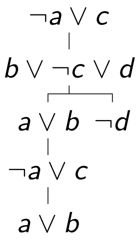
a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$
0	0	0	$\underline{0}$	0	e
0	0	0	0	$\underline{1}$	$a \lor b$
$\underline{1}$	$\underline{0}$	0	0	1	$\neg a \lor c$
$\underline{0}$	0	$\underline{1}$	0	1	$a \lor b$
$\underline{1}$	$\underline{0}$	1	0	1	$b \lor \neg c \lor d$
1	$\underline{1}$	$\underline{0}$	$\underline{1}$	1	$\neg d$



Witness Trees: Example

$$(a \vee b) \wedge (\neg a \vee c) \wedge (b \vee \neg c \vee d) \wedge \neg d \wedge e$$

a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$
0	0	0	$\underline{0}$	0	e
0	0	0	0	$\underline{1}$	$a \vee b$
$\underline{1}$	$\underline{0}$	0	0	1	$\neg a \vee c$
$\underline{0}$	0	$\underline{1}$	0	1	$a \vee b$
$\underline{1}$	$\underline{0}$	1	0	1	$b \vee \neg c \vee d$
1	$\underline{1}$	$\underline{0}$	$\underline{1}$	1	$\neg d$
1	1	0	$\underline{0}$	1	$\neg a \vee c$



Witness Trees: Example

$$(a \lor b) \land (\neg a \lor c) \land (b \lor \neg c \lor d) \land \neg d \land e$$

a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$
0	0	0	$\underline{0}$	0	e
0	0	0	0	$\underline{1}$	$a \lor b$
$\underline{1}$	$\underline{0}$	0	0	1	$\neg a \lor c$
$\underline{0}$	0	$\underline{1}$	0	1	$a \lor b$
$\underline{1}$	$\underline{0}$	1	0	1	$b \lor \neg c \lor d$
1	$\underline{1}$	$\underline{0}$	$\underline{1}$	1	$\neg d$
1	1	0	$\underline{0}$	1	$\neg a \lor c$
$\underline{0}$	1	$\underline{0}$	0	1	done

Witness Trees

Lemma

Let τ_n be a witness tree produced by the algorithm. Let $v, w \in V(\tau_n)$ be two distinct nodes at the same depth. Then $\mathrm{vbl}([v]) \cap \mathrm{vbl}([w]) = \emptyset$. In particular, τ_n is proper.

Proof.

- Choose i, j maximally such that $v \in V(\tau_n^{(i)}), w \in V(\tau_n^{(j)})$.
- Assume $i < j$ (i.e. v added after w) without loss.
- Seeking a contradiction, assume $\mathrm{vbl}([v]) \cap \mathrm{vbl}([w])$ is nonempty.
- Then the algorithm would have added v as a child of w at greater depth.

$\square$

Witness Trees

Lemma

Let τ be a proper witness tree. Then

$$\mathbb{P}(\tau = \tau_n \text{ for some } n \in \mathbb{N}) \leq \prod_{v \in V(\tau)} \mathbb{P}([v]).$$

Witness Trees

Lemma

Let τ be a proper witness tree. Then

$$\mathbb{P}(\tau = \tau_n \textit{ for some } n \in \mathbb{N}) \leq \prod_{v \in V(\tau)} \mathbb{P}([v]).$$

- Let $v_1, \ldots, v_k$ be the vertices of τ in depth-decreasing order.
- If we sample $[v_1], \ldots, [v_k]$ independently, then $\mathbb{P}([v_1] \cap \cdots \cap [v_k]) = \prod_{i=1}^{k} \mathbb{P}([v_i])$.
- **Claim:** When using the *same random source* as the resampling algorithm appropriately, then τ is produced $\Longrightarrow$ all $[v_1], \ldots, [v_k]$ hold

Witness Trees

a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$
0	0	0	$\underline{0}$	0	e
0	0	0	0	$\underline{1}$	$a \vee b$
$\underline{1}$	$\underline{0}$	0	0	1	$\neg a \vee c$
$\underline{0}$	0	$\underline{1}$	0	1	$a \vee b$
$\underline{1}$	$\underline{0}$	1	0	1	$b \vee \neg c \vee d$
1	$\underline{1}$	$\underline{0}$	$\underline{1}$	1	$\neg d$
1	1	0	$\underline{0}$	1	$\neg a \vee c$

$\neg a \vee c$

$b \vee \neg c \vee d$

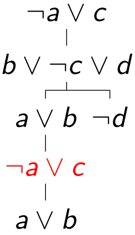
$a \vee b \quad \neg d$

$\neg a \vee c$

$a \vee b$

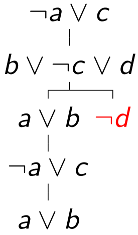
Witness Trees

a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$
0	0	0	$\underline{0}$	0	e
0	0	0	0	$\underline{1}$	$a \vee b$
$\underline{1}$	$\underline{0}$	0	0	1	$\neg a \vee c$
$\underline{0}$	0	$\underline{1}$	0	1	$a \vee b$
$\underline{1}$	$\underline{0}$	1	0	1	$b \vee \neg c \vee d$
1	$\underline{1}$	$\underline{0}$	$\underline{1}$	1	$\neg d$
1	1	0	$\underline{0}$	1	$\neg a \vee c$



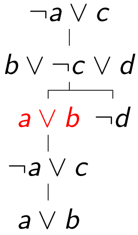
Witness Trees

a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$
0	0	0	$\underline{0}$	0	e
0	0	0	0	$\underline{1}$	$a \vee b$
$\underline{1}$	$\underline{0}$	0	0	1	$\neg a \vee c$
$\underline{0}$	0	$\underline{1}$	0	1	$a \vee b$
$\underline{1}$	$\underline{0}$	1	0	1	$b \vee \neg c \vee d$
1	$\underline{1}$	$\underline{0}$	$\underline{1}$	1	$\neg d$
1	1	0	$\underline{0}$	1	$\neg a \vee c$



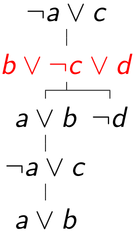
Witness Trees

a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$
0	0	0	$\underline{0}$	0	e
0	0	0	0	$\underline{1}$	$a \vee b$
$\underline{1}$	$\underline{0}$	0	0	1	$\neg a \vee c$
$\underline{0}$	0	$\underline{1}$	0	1	$a \vee b$
$\underline{1}$	$\underline{0}$	1	0	1	$b \vee \neg c \vee d$
1	$\underline{1}$	$\underline{0}$	$\underline{1}$	1	$\neg d$
1	1	0	$\underline{0}$	1	$\neg a \vee c$



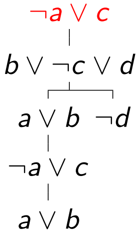
Witness Trees

a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$
0	0	0	$\underline{0}$	0	e
0	0	0	0	$\underline{1}$	$a \vee b$
$\underline{1}$	$\underline{0}$	0	0	1	$\neg a \vee c$
$\underline{0}$	0	$\underline{1}$	0	1	$a \vee b$
$\underline{1}$	$\underline{0}$	1	0	1	$b \vee \neg c \vee d$
1	$\underline{1}$	$\underline{0}$	$\underline{1}$	1	$\neg d$
1	1	0	$\underline{0}$	1	$\neg a \vee c$



Witness Trees

a	b	c	d	e	C
0	0	0	1	0	e
0	0	0	1	$\underline{0}$	$\neg d$
0	0	0	$\underline{0}$	0	e
0	0	0	0	$\underline{1}$	$a \vee b$
$\underline{1}$	$\underline{0}$	0	0	1	$\neg a \vee c$
$\underline{0}$	0	$\underline{1}$	0	1	$a \vee b$
$\underline{1}$	$\underline{0}$	1	0	1	$b \vee \neg c \vee d$
1	$\underline{1}$	$\underline{0}$	$\underline{1}$	1	$\neg d$
1	1	0	$\underline{0}$	1	$\neg a \vee c$



Witness Trees

Claim: When using the same random source as the resampling algorithm appropriately, then
τ is produced $\implies$ all $[v_1], \ldots, [v_k]$ hold.

Proof.

- For $P \in \mathcal{V}$ the algorithm uses independent samples $P_0, P_1, \ldots$.
- Assume the random source leads to τ being produced.
- Let $i \in \{1, \ldots, k\}$.
- For $P \in \mathrm{vbl}([v_i])$ let $S(P) := \{j \mid 1 \leq j < i, P \in \mathrm{vbl}([v_j])\}$
- When sampling $[v_i]$, we get the sample $P_{\#S(P)}$.
- At the step before v_i was resampled by the algorithm, x_P has value $P_{\#S(P)}$. Thus $[v_i]$ holds.
- This is because every time x_P was resampled corresponds to a unique $v \in V(\tau)$ with $P \in \mathrm{vbl}([v])$.
- This v has greater depth than v_i, thus $v \in \{v_j \mid j \in S(P)\}$.

$\square$

Random Reasons

Algorithm 1 Random Witness Tree Generator G_A

```
τ ← (A)
repeat
    L ← Leaves(τ)
    done ← true
    for v ∈ L do
        for B ∈ Γ⁺(v) do
            if rand() < x(B) then
                add (B) as child of v
                done ← false
            end if
        end for
    end for
until done
return τ
```

Random Reasons

Write $x'(A) := x(A) \prod_{B \in \Gamma(A)} (1 - x(B))$.

Lemma

Let τ be a proper witness tree with root labelled A. Then algorithm G_A produces τ with probability $p_\tau = \frac{1-x(A)}{x(A)} \prod_{v \in V(\tau)} x'([v])$.

Proof.

For $v \in V(\tau)$ let $W_v := \Gamma^+([v]) \setminus \{[c] \mid c \text{ child of } v\}$. Then

$$
\begin{aligned}
p_\tau &= \frac{1}{x(A)} \prod_{v \in V(\tau)} x([v]) \prod_{u \in W_v} (1 - x(u)) \\
&= \frac{1 - x(A)}{x(A)} \prod_{v \in V(\tau)} \frac{x([v])}{1 - x([v])} \prod_{u \in \Gamma^+([v])} (1 - x(u)) \\
&= \frac{1 - x(A)}{x(A)} \prod_{v \in V(\tau)} x'([v]).
\end{aligned}
$$

$\square$

The Proof

Proof.

Let $A \in \mathcal{A}$ and let T_A be the set of proper A-rooted witness trees.

$$\begin{aligned}
\mathbb{E}(\#\text{ times } A \text{ gets resampled}) &= \sum_{\tau \in T_A} \mathbb{P}(A \text{ gets resampled for reason } \tau) \\
&\leq \sum_{\tau \in T_A} \prod_{v \in V(\tau)} \mathbb{P}([v]) \\
&\leq \sum_{\tau \in T_A} \prod_{v \in V(\tau)} x'([v]) \\
&= \sum_{\tau \in T_A} \frac{x(A)}{1 - x(A)} p_\tau \\
&= \frac{x(A)}{1 - x(A)} \sum_{\tau \in T_A} p_\tau \\
&\leq \frac{x(A)}{1 - x(A)}
\end{aligned}$$

Another Example

- Let $n, k \in \mathbb{N}$ such that $4\binom{k}{2}\binom{n}{k-2}2^{1-\binom{k}{2}} \leq 1$
- **Claim:** We can 2-color the edges in the complete graph K_n such that no k vertices form a monochromatic subgraph
- Let $\mathcal{A} := \{C_I \mid I \in \binom{n}{k}\}$ be the events that $I \subseteq \underline{n}$ is monochromatic under a random coloring.
- We have $|\Gamma(C_I)| \leq \binom{k}{2}\binom{n}{k-2}$
- Choose $x(C_I) = (\binom{k}{2}\binom{n}{k-2})^{-1} =: N^{-1}$ and assume that all edges are colored red with probability $\frac{1}{2}$
- We then have

$$2^{1-\binom{k}{2}} = \mathbb{P}(C_I) \leq (4N)^{-1} \leq N^{-1}(1 - \frac{1}{N})^N$$

- **Note:** More efficient for lower bounds on offdiagonal ramsey numbers