Today

1) The Probabilistic Method (applied to k-SAT)
    - ↳ Via union bound
    - ↳ Via independence
    - ↳ via Lóvasz Local Lemma
2) From the Probabilistic Method to Algorithms

Probabilistic Method Framework (to show (*) is possible)

a) Define a random process

b) Define bad events $B_1, B_2, \ldots$ and $\bar{B}_1 \cap \bar{B}_2 \cap \ldots \rightarrow (*)$

c) Show $\Pr(\bar{B}_1 \cap \bar{B}_2 \ \ldots) > 0$

↳ Via union bound

↳ Via independence

↳ Via Lóvasz Local Lemma...

A <u>literal</u> is a boolean variable or its negation X or $\bar{X}$

A <u>Clause</u> is the "or" of distinct literals $\bar{X}_1 \vee X_2 \vee X_3$

↳ <u>k-Clause</u> if exactly k literals ↑ 3-clause

A <u>k-SAT formula</u> is the "and" of k-clauses (variables $X_1, \ldots, X_n$, m clauses)

2-SAT: $(X_1 \vee X_2) \wedge (\bar{X}_1 \vee X_2) \wedge (X_1 \vee \bar{X}_2)$ → satisfiable by $X_1 = X_2 = $ true

$(X_1 \vee X_2) \wedge (\bar{X}_1 \vee X_2) \wedge (X_1 \vee \bar{X}_2) \wedge (\bar{X}_1 \vee \bar{X}_2)$ → not satisfiable b/c ≥ 1 true, ≥ 1 false by ends but then a middle clause not satisfied

A k-SAT formula is <u>satisfiable</u> (*) if $\exists$ a truth assignment to its variables making it true

Intuition: many variables + few clauses = easy to satisfy

(a) [Independently assign each $X_i = \begin{cases} \text{true} & \text{w/ Pr } .5 \\ \text{False} & \text{w/ Pr } .5 \end{cases}$

(b) [Let $B_i :=$ ith clause not satisfied so if $\bar{B}_1 \cap \bar{B}_2 \cap \ldots$ then formula is satisfied

<u>Fact:</u> Any k-SAT clause w/ $m \leq \frac{2^k}{e} - 1$ clauses is satisfiable (see above eg.)

(c) [Have $\Pr(B_i) \leq \left(\frac{1}{2}\right)^k \ \forall i$ so $\Pr(B_1 \cup B_2 \cup \ldots) \leq \sum_i \Pr(B_i) < \frac{1}{2^k} \cdot \frac{2^k}{e} = \frac{1}{e} < 1$

so $\Pr(\bar{B}_1 \cap \bar{B}_2 \cap \ldots) = 1 - \Pr(B_1 \cup B_2 \cup \ldots) > 0$

A k-SAT formula has <u>overlap</u> α if each clause share variables w/ $\le \alpha$ other clauses

$$(x_1 \vee x_2) \wedge (x_3 \vee x_4) \rightarrow \text{overlap } 0$$

$$(x_1 \vee \bar{x}_2) \wedge (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (\bar{x}_3 \vee x_4) \rightarrow \text{overlap } 3$$

Intuition: smaller $\alpha \rightarrow$ easier to satisfy

<u>Fact</u>: any k-SAT formula w/ $\alpha = 0$ is satisfiable

Trivial to prove directly, but instructive to prove w/ probability

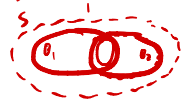
Why can't use union bound approach

Algebraically: Have $\Pr(B_i) \le \left(\frac{1}{2}\right)^k \ \forall i$ so $\Pr(B_1 \cup B_2 \cup \dots) \le \sum_i \Pr(B_i) \not< \frac{1}{2^k} \cdot 2k = 1$

Morally:

Union bound

Good upper bound if B_i are (mostly) disjoint



Independence

Implies not disjoint (assuming $\neq 0$ Pr ...)

Here B_i are independent $\rightarrow$ not disjoint so UB is bad

(c) $\Pr(\bar{B}_i) = 1 - \frac{1}{2^k} > 0 \ \forall i$ so $\Pr(\bar{B}_1 \cap \bar{B}_2 \cap \dots) = \prod_i \Pr(\bar{B}_i) > 0$

B_i independent so $\bar{B}_i$ independent

$\bar{B}_i > 0 \ \forall i$

Fact: Any k-SAT formula w/ $\alpha \le \frac{2^k}{e} - 1$ is satisfiable

Note: no dependence on # of clauses

If $\alpha > 0$ then B_i no longer independent

But small $\alpha \rightarrow$ "mostly independent"; need way to formalize

Event A is <u>mutually independent</u> of events $\mathcal{B} = \{B_1, B_2, \dots\}$ if $\forall$ partitions $\mathcal{B} = \mathcal{B}_1 \sqcup \mathcal{B}_2$

$$\Pr\left(A \,\middle|\, \bigcap_{B \in \mathcal{B}_1} B \cap \bigcap_{B \in \mathcal{B}_2} \bar{B}\right) = \Pr(A)$$

B_4 MI of $\{B_1, B_2\}$ in
$$\underbrace{(x_1 \vee \bar{x}_2)}_{B_1} \wedge \underbrace{(x_1 \vee x_2)}_{B_2} \wedge \underbrace{(x_1 \vee x_3)}_{B_3} \wedge \underbrace{(\bar{x}_3 \vee x_4)}_{B_4}$$

$(\mathcal{B}, E)$ is a <u>dependency graph</u> of events $\mathcal{B}$ if $\forall B \in \mathcal{B}$

B is MI from $\mathcal{B} \setminus \underbrace{\Gamma(B)}_{\text{neighbors of } B}$

$$\begin{tikzcd} B_1 \arrow[r, dash] \arrow[rr, dash, bend right] & B_2 \arrow[r, dash] & B_3 \arrow[r, dash] & B_4 \end{tikzcd}$$

<u>Symmetric LLL:</u> Given events $\mathcal{B}$ w/ dependency graph G of max-degree Δ

if $\exists P$ s.t.

1) $\Pr(B) \le P \quad \forall B \in \mathcal{B}$
2) $e \cdot P \cdot (\Delta + 1) \le 1$

then $\Pr\left(\bigcap_{B \in \mathcal{B}} \bar{B}\right) > 0$

(c)

<u>Proof of Fact using LLL</u>

Let $\mathcal{B} := \{B_1, B_2, \dots\}$ and $\{B_i, B_j\} \in E$ if B_i, B_j share variables, let $P = \frac{1}{2^k}$ and $\Delta = \alpha$

$(\mathcal{B}, E)$ is a dependency graph w/ max-degree Δ

But $\Pr(B_i) \le \frac{1}{2^k} = P \quad \forall i$

So $e \cdot P \cdot (\Delta + 1) \le e \cdot \frac{1}{2^k} \cdot \frac{2^k}{e} = 1$

So by LLL $\Pr\left(\bigcap_i \bar{B}_i\right) > 0$

LLL as a Union Bound Generalization

Suppose n events $B_1, B_2, \dots, B_n$ w/ $\Pr(B_i) \le p \ \forall i$

UB: If $p \cdot n < 1$ then $\Pr(\bigcup_i B_i) \le p \cdot n < 1$ so $\Pr(\bigcap_i \bar{B}_i) > 0$

LLL on complete graph: If $e \cdot p \cdot (\Delta+1) = e \cdot p \cdot n \le 1$ then $\Pr(\bigcap_i \bar{B}_i) > 0$

From Probabilistic Method to Algorithms

For UB: boosting

Suppose # clauses is $\le \frac{2^k}{e} - 1$ so $\Pr(1 \text{ random assignment not satisfying}) \le \frac{1}{e}$

Assign $x_1, x_2, \dots$ UAR

While $\exists$ unsatisfied clause

Resample all variables

Return $x_1, x_2, \dots$

Analysis: $\Pr(\le r \text{ iterations}) = 1 - \Pr(> r \text{ iterations}) \ge 1 - \left(\frac{1}{e}\right)^r$

so $\Pr(\le \ln n \text{ iterations}) \ge 1 - \frac{1}{n}$

For LLL: Moser-Tardos Algorithm → works for LLL in general

MT

Assign $x_1, x_2, \dots$ UAR

While $\exists$ unsatisfied clause c

Fix(c)

Return $x_1, x_2, \dots, x_n$

Fix(c)

Resample each $x_i \in c$

For each unsatisfied clause c' sharing variables w/ c → possibly c itself

Fix(c')

Analysis: "entropy compression" shows $O(m)$ iterations in $\mathbb{E}$