Improved Algorithmic Versions of the Lovász Local Lemma

Aravind Srinivasan

University of Maryland

Speaker: Julián Mestre, MPII

The Local Lemma: A Powerful Tool

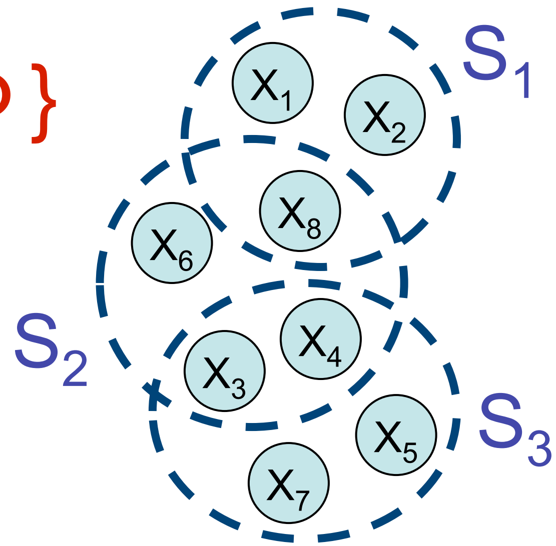
- **LLL (symm. Vers.)**
  - "Bad" events E_1, E_2, …, $E_{m,}$ want to avoid all
  - $Pr[E_i] \leq p$, "dependency" at most D
  - If $ep(D+1) \leq 1$ then $Pr[\text{all } E_i \text{ avoided}] \geq exp(-\theta \Sigma_i Pr[E_i])$

- [Beck, Alon, Molloy-Reed]: algorithmic versions
  - (super) polynomial time
  - gap with nonconstructive version

- Here: gap-reduction, fully poly-time (randomized)

Molloy & Reed: Set-System Framework

- Underlying **independent** R.V.s $X_1, \ldots, X_n$
- Subsets $S_1, \ldots, S_m$ of $\{ 1, \ldots, n \}$
- E_i determined by $\{ X_t : t \in S_i \}$
  depends on $\{ E_u : S_u \cap S_i \neq \Phi \}$



Want: assign't of X_t avoiding all E_i

Known results for [MR] framework

- **LLL:**
  - $Pr[E_i] \le p$, each S_i intersects at most D others
  - If *$ep(D+1) \le 1$, $Pr[\text{avoid all } E_i] \ge \exp(-\theta(\Sigma_i Pr[E_i]))$*
  - That is, "$pD \le O(1)$" suffices (*non-constructive*)

- [MR]:
  - "$pD^9 \le O(1)$" suffices (*algorithmic*)
  - Runtime depends on support-size of X_t and $|S_i|$
  - Often polynomial, but superpoly. in worst case

Example: Hypergraph 2-coloring

- Given a k-uniform hypergraph H = (V,E):
  - can V be 2-colored with no edge monochromatic?
- Natural rand. algorithm and [MR] set-system:
  - $p = Pr[E_i] = \mathbf{2^{1-k}} \rightarrow$ **$D \leq c\ 2^k$ suffices** by LLL
- [MR]'s "$pD^9 \leq O(1)$"
  - **$D \leq c'\ 2^{k/9}$ suffices** for efficient algorithm
- **Is it possible to approach "$D \leq c\ 2^k$ suffices" algorithmically?**

Color vertices randomly and idependently

binary X_j for each j ε V
E_i ≡ "edge #i monochromatic"
subset S_i (= edge #i) for edge #i
D = max. #edges overlapping any edge

Our first result

- Model:
  - R.V.s $X_1, X_2, \ldots, X_n$, events $E_1, E_2, \ldots, E_m$
  - each X_j samplable in poly(n,m) time
  - given the X_j, truth of any E_i checkable in poly(n,m) time

- [MR]:
  - "$pD^9 \le O(1)$" suffices ($D \le c' 2^{k/9}$ suffices for hyp. 2-col.)
  - runtime often polynomial, sometimes super-polynomial

- Here:
  - "$pD^4 \le O(1)$" suffices ($D \le c'' 2^{k/4}$ suffices for hyp. 2-col.)
  - runtime always polynomial in n and m

Two-phase algorithm

- Given an [MR] set-system with $pD^4 \leq c_0$, want to efficiently construct $\{X_j\}$ that avoid all E_i

- Phase I: assignment **A** of *some* of the X_j
  - ensure $\mathbf{Pr[E_i \mid A] \leq 1/(e(D+1))}$ for all E_i

- Phase II: problem restricted to the yet-unset X_j
  - break up events into *independent components*
  - randomly sample each yet-unset X_j from its distribution

What do we require of **A**?

- Analysis of component *C* in Phase 2:
  - Define $Pr'[E_i] = Pr[E_i \mid \mathbf{A}]$; recall $Pr'[E_i] \leq 1/(e(D+1))$
  - LLL → Pr[all E_i in comp. *C* avoided] ≥ $exp(-\theta(\Sigma_{i \in C} Pr'[E_i]))$
- Thus, requirement of (random) **A:**
  - $Pr'[E_i] \leq 1/(e(D+1))$ for all i (with prob. 1)
  - with high prob. (over the random choice of **A**), **all** comp.s *C* resulting from **A** have $\mathbf{\Sigma_{i \in C} Pr'[E_i] \leq O(\log m)}$
- Just "repeat" poly(m) many times for C in Phase II

Phase I Description

- Process in the order $X_1, \ldots, X_n$; initially none frozen
- Processing X_j:
  - if X_j currently frozen, skip over (handle in Phase II). **Else:**
  - generate (tentative) rand. value v_j for X_j from its distribution. Let $\mathbf{A_j}$ = set of value-assignments so far
  - for each E_i dependent on X_j do:

    if $\mathbf{Pr[E_i \mid A_j] > 1/(e(D+1))}$ /* call E_i "bad" */

    "undo" the setting of X_j

    freeze all yet-unset X_k (incl. X_j) on which E_i depends

- The setting $\mathbf{A}$ at the end has $\mathbf{Pr[E_i \mid A] \le 1/(e(D+1))}$

Remarks

- Recall: $Pr'[E_i] = Pr[E_i \mid \mathbf{A}]$; want (with high prob.) **all** comp.s *C* have $\mathbf{\Sigma_{i \in C} Pr'[E_i] \leq O(\log m)}$
- Idea: $Pr'[E_i] \leq Pr[E_i] / Pr[\mathbf{A}] \leq p / Pr[\mathbf{A}]$;
  so, $Pr'[E_i]$ "large" → $Pr[\mathbf{A}]$ is "small"
  i.e., unlikely to get many such E_i
  - use defn. of *bad*, enum. spanning trees *(2-3 trees*), …
- Our main contributions here:
  - augmentation of the 2-3 trees of [Beck, Alon, M-R]
  - explicitly using the LLL's success prob. in Phase II

Our second result

Parallel algorithm for hypergraph 2-coloring

- NC/RNC alg. for 2-coloring a given k-uniform hypergraph?

- [Alon '91]: $D \leq c' \, 2^{k / 500}$ suffices
  - did not attempt to optimize the constant "500"

- Here: $D \leq c'' \, 2^{k / 10.34}$ suffices
  - further augmentation of 2-3 trees
  - useful lemma on weighted indepedent sets

Open Questions

- Algorithmically approach "$pD \leq O(1)$ suffices"?
  - even "$pD^{3-\varepsilon} \leq O(1)$ suffices" seems challenging
- Derandomization?
  - Phase I : use *method of conditional probabilities*
  - Phase II : *inherently randomized*
- More powerful versions of the LLL (asymmetric, lopsided, …)? Parallel versions?

Thank you for
your Attention!