CS265, Winter 2022

Class 12: Agenda and Questions

1 Announcements

- HW6 due Wednesday!
- You get a week off before HW7! (Due to President's day)

2 Questions?

Any questions from the minilectures and/or the quiz? (Constructive LLL)

3 You prove the constructive LLL for another problem!

Consider the following problem (which has featured on a quiz). You are coloring the integers $\{1, \ldots, n\}$ either blue or red. You are given as input a collection of sets $S_1, S_2, \ldots, S_m \subseteq \{1, \ldots, n\}$, so that:

- Each set S_i has size at most k.
- Each set S_i intersects at most d other sets S_j, for some $d > 1$.

Our goal is to color the points $\{1, \ldots, n\}$ so that there is no monochromatic set S_i. (A set S_i is *monochromatic* if every element of it is either red or blue).

Group Work

1. Mimic the proof of the constructive LLL that we saw for k-SAT to give a randomized algorithm that does the following.

   > Suppose that $k \geq \log_2 d + 10000$. (Here, 10000 is a stand-in for "some big enough constant.") Then there is a randomized algorithm that proceeds by re-randomizing the sets S_i, (that is, it will iteratively look at different sets S_i and randomly re-color all of the points in that set), so that:
   >
   > - If the algorithm terminates, then all of the numbers $\{1, \ldots, n\}$ will be colored so that there is no monochromatic set S_j.
   > - The expected number of times that the algorithm re-randomizes a set S_j is $\mathrm{poly}(m)$.

   Don't worry about giving a complete proof with all the details, just work it out with enough detail that you believe it. As we did in the minilecture video, you may use the (informal) fact that "there is no function $f : \{0,1\}^X \to \{0,1\}^Y$ so that (a) $Y \ll X$ and (b) with high probability over a uniformly random $x \in \{0,1\}^X$, it is possible to recover x given $f(x)$."

Hint*: To map this problem onto k-SAT, think of the S_j's as standing in for clauses, and the numbers $\{1,\ldots,n\}$ as standing in for variables.*

Hint*: It's not quite as straightforward as applying the mapping in the previous hint and calling it a day. In particular, can you still work backwards from the "print" statements in the k-SAT version to figure out the original random bits?*

2. What happens to your proof if the number of possible colors grows from two (blue and red) to some number t? In particular, can you get the same guarantee as above, but under a weaker guarantee (eg, $k \geq$ [something smaller than $\log d + 10000$]).

3. How does the answer that you got in the previous part compare to what Corollary 3 in the lecture notes would give you for this problem?

   As a reminder, that Corollary says:

   > Let V be a finite set of independent random variables. Let $\mathcal{A}$ be a finite set of events determined by the random variables in V. If for all $A \in \mathcal{A}$, $|\Gamma(A)| \leq d+1$, and $\Pr[A] \leq \frac{1}{e(d+1)}$, then the algorithm from the lecture notes (the general one, not just for k-SAT) will find an assignment to the variables V such that no event of $\mathcal{A}$ occurs. Additionally, the expected number of "re-randomizations" performed by the algorithm is bounded by $O(|\mathcal{A}|/(d+1))$.

4. [**This question is open-ended and may be difficult—think about it after you finish the others if you still have time.**] What happens to your proof if the sets can have variable size? (e.g., if all but a few of them have size k, and a few can be really small? Or if they have average size k? Or....?)

4 Using the asymmetric LLL

(We will (probably) not cover this in class; it's here in case there's extra time and/or you finish early)

Recall that the asymmetric LLL (that is, the more general statement that we had in the lecture notes, both for the algorithmic and non-algorithmic version) was:

Theorem 1. *Let V be a finite set of independent random variables. Let $\mathcal{A}$ be a finite set of events determined by the random variables in V. If there exists an assignment $x : \mathcal{A} \to (0,1)$ such that for all $A \in \mathcal{A}$,*

$$\Pr[A] \leq x(A) \prod_{B \in \Gamma(A) \setminus \{A\}} (1 - x(B)),$$

then [the algorithm from the mini-lecture] will find an assignment to the variables V such that *no event of $\mathcal{A}$ occurs. Additionally, the expected number of "re-randomizations" is bounded by* $\sum_{A \in \mathcal{A}} \frac{x(A)}{1-x(A)}$.

We saw in the mini-lecture how to derive the symmetric version from this (set $x(A) = 1/(d+1)$ for all A), but it might seem pretty unclear how to apply it in the general asymmetric case. In this exercise you'll work out an example.

Group Work

Suppose that n and ℓ are sufficiently large, and suppose that $\ell \gg \sqrt{n} \cdot \log(n)$. Show that there is an edge-coloring of the complete graph on n vertices by two colors (blue or red) that contains *neither* a blue triangle *nor* a red ℓ-clique. (And that the algorithm from class will find such a coloring).

(Don't worry too much about constants in your proof; in particular, it's fine to treat $e^x \approx 1 + x$ as an equality for small x).

Note: This gives a lower bound on the Ramsey number $R(3, \ell) \gtrsim \ell^2/\log^2(\ell)$, where $R(k, \ell)$ is the smallest n so that any coloring of K_n must contain either a red k-clique or blue ℓ-clique.

Hint: Consider coloring the vertices red with probability p and blue with probability $1 - p$, where $p = \Theta(1/\sqrt{n})$.

Hint: You may run into a situation where you want to count the number of sets over vertices of size ℓ that intersect a set T of size 3 in at least two places. Try trivially bounding this by $\binom{n}{\ell}$, which is just the number of sets of size ℓ.

Hint: If you would like an oracle to tell you how you should choose the $x(A)$, ask Mary during class...