CS265, Fall 2023

Class 11: Agenda and Questions

1 Announcements

- HW5 due Friday!

2 Recap/Questions?

Any questions from the minilectures and/or the quiz (second moment method and LLL)?

3 Practice with the LLL

Recall the k-*SAT* problem. There are n variables $x_1, \ldots, x_n$. We consider clauses that looks like $(x_{i_1} \vee x_{i_2} \vee \overline{x_{i_3}} \vee \cdots \vee x_{i_k})$; that is, a clause is the OR of k literals. **For today, assume that each clause has k distinct variables that appear in it.** We have a formula φ that is the AND of m clauses. We would like to know: is φ satisfiable? That is, is there a way to assign values to the variables $x_1, x_2, \ldots$ so that φ evaluates to TRUE?

Group Work

Suppose that each variable x_i is in at most t clauses, for some parameter t that will depend on k and that you'll work out in this problem. Apply the LLL to get a statement like the following:

> Suppose that each variable is in at most t clauses of φ. Then φ is satisfiable.

(You should try to get t to be as large as possible. It's not hard to see that the statement above is true if, say, $t = 1$, but you should get a value of t that grows with k.)

***Hint**: Recall that to apply the LLL, you need to define a probability distribution and a set of "bad" events. We set up this example in the minilecture video, we just didn't work out the conclusion. In the set-up of the video, we considered the probability distribution to correspond to assigning TRUE/FALSE to each variable $x_1, \ldots, x_n$ independently with probability $1/2$ each, and we defined the bad event A_i to be the event that clause i is not satisfied.*

3.1 More Practice with LLL and Mutual Independence

Here's an example where the mutual independence requirement is a bit trickier to think about. Consider a set of m equations over variables $x_1, \ldots, x_n$:

$$\sum_{j=1}^{n} a_j^{(1)} x_j \equiv b^{(1)} \mod 17$$
$$\sum_{j=1}^{n} a_j^{(2)} x_j \equiv b^{(2)} \mod 17$$
$$\vdots$$
$$\sum_{j=1}^{n} a_j^{(m)} x_j \equiv b^{(m)} \mod 17$$

where:

- For all $j = 1, \ldots, n$ and all $r = 1, \ldots, m$, the coefficients $a_j^{(r)} \in \{0, 1, 2, \ldots, 16\}$ are not all zero; and
- for all $r = 1, \ldots, m$, $b^{(r)} \in \{0, 1, \ldots, 16\}$.

Suppose that each variable x_j appears in at most 4 of the m equations. (That is, for each j, $a_j^{(r)} = 0$ for all but four values of r.)

Group Work

With the setup above, prove that there exists an assignment to the variables such that *none* of the equations are satisfied.

Hint: *Recall that because 17 is prime, for any* $a \in \{1, \ldots, 16\}$ *and any* $b \in \{0, \ldots, 16\}$, *the equation* $ax \equiv b \mod 17$ *has a unique solution for* $x \in \{0, \ldots, 16\}$.

Hint: *It might be helpful to go back to the definition of mutual independence when arguing about the value of* d *when applying the LLL.*

4 Practice with derandomization via conditional expectation (from last class)

I don't expect we will get to this today in class, but if you finish the rest early, try this part that we didn't get to last week!

Group Work

1. (Bonus) Let φ be a 3-CNF formula with n variables and m clauses, and 3 distinct variables in each clause. Use the method of derandomization via conditional expectation to give an efficient (polynomial in n, m) deterministic algorithm to find an assignment to φ so that at least a 7/8-fraction of the clauses are satisfied.
2. (Even more bonus) There is also a natural greedy algorithm for this problem:
   - For $i = 1, 2, \dots, n$:
     - Assign x_i to be whichever value makes the most currently unsatisfied clauses true (breaking ties arbitrarily).

   In the previous example (maximizing the size of a cut), the algorithm we came up with was secretly the natural greedy algorithm. Is your algorithm from the previous part the same as this natural greedy algorithm? Is it better or worse?