CS265, Winter 2025

Class 10: Agenda and Questions

1 Announcements

- HW4 due Friday!

2 Warm-Up

Group Work

1. Show that, in any undirected, unweighted graph $G = (V, E)$ with no self-loops, there is a cut with at least $|E|/2$ edges that cross it. (Recall that a *cut* in G is just a partition of the vertices $V = S \cup \bar{S}$, and that an edge $\{u, v\}$ crosses the cut if $u \in S$ and $v \in \bar{S}$ or the other way around).
2. Let φ be a 3-CNF formula. That is, φ is the AND of a bunch of clauses that look like $(x \vee y \vee z)$ (or $(x \vee \bar{y} \vee \bar{z})$, or ..., where $\bar{x}$ means "not x"). For example, maybe

   $$\varphi = (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_2 \vee \overline{x_4} \vee x_5) \wedge \cdots \wedge (x_{23} \vee \overline{x_1} \vee \overline{x_5}).$$

   Suppose that each clause has three distinct literals that appear in it. (e.g., $(x_1 \vee x_1 \vee x_1)$ is not allowed).
   Given an *assignment* σ to the variables $x_1, x_2, \ldots$ (eg, $x_1 = TRUE, x_2 = FALSE$, etc), we say that a clause of φ is *satisfied* by σ if that clause evaluates to TRUE.
   Show that any 3-CNF formula φ has an assignment σ so that at least $7/8$ of the clauses are satisfied.

Group Work: Solutions

1. We choose a random cut. The expected number of edges that cross it is

   $$\mathbb{E}[\text{edges that cross}] = \sum_{e \in E} \mathbb{P}[\ e \text{ crosses cut }\ ] = |E|/2.$$

   Thus, there exists a cut with at least that many edges that cross it.

2. We choose a random assignment.

$$\mathbb{E}[\text{number of clauses satisfied}] = \sum_C \mathbb{P}[\ C \text{ is satisfied }] = (\text{num clauses}) \cdot (7/8)$$

because there's only one way out of 8 to fail to satisfy a clause (False OR false OR false). So there exists an assignment σ that satisfies at least a 7/8 fraction of clauses.

3 Recap/Questions

Any questions from the minilectures and/or the quiz? (The probabilistic method; Ramsey numbers; Independent sets)

4 Derandomization via conditional expectation

In class today, we'll explore a general way to turn an existence proof—like the ones from your warm-up exercise—into an algorithm. This is called "Derandomization via conditional expectation."

Group Work

Our goal in this group work is to find an efficient, deterministic algorithm to find a cut $(S, \bar{S})$ so that the number of edges crossing the cut is at least $|E|/2$. In general, finding a cut with the *maximum* number of edges crossing it is NP-hard; but this will at least find a large-ish cut.

Note: There is a straightforward deterministic greedy algorithm to do this. Here, we'll see a way to derive a deterministic algorithm using conditional expectations.

1. Let $G = (V, E)$ be as in warm-up question 1. Suppose the vertices are ordered $V = \{v_1, v_2, \ldots, v_n\}$.

   Suppose that $S \subseteq V$ is chosen uniformly at random (that is, each v_i is included in S indepedently with probability 1/2). Let X be the number of edges crossing the cut $(S, \bar{S})$.

   Convince yourself that $\mathbb{E}[X | v_1 \in S] = |E|/2$.

2. Suppose that you have made some choices for $v_1, v_2, \ldots, v_{t-1}$ (eg, $v_1 \in S, v_2 \notin S, v_3 \in S, \ldots, v_{t-1} \in S$), so that

$$\mathbb{E}[X | \text{ choices for } v_1, \ldots, v_{t-1}\ ] \geq \frac{|E|}{2}.$$

Show that **either**

$$\mathbb{E}[X \mid \text{choices for } v_1, \ldots, v_{t-1};\ \text{and } v_t \in S] \geq \frac{|E|}{2}$$

or

$$\mathbb{E}[X \mid \text{choices for } v_1, \ldots, v_{t-1};\ \text{and } v_t \notin S] \geq \frac{|E|}{2}$$

3. Again, suppose that you have made choices for $v_1, \ldots, v_{t-1}$ so that

$$\mathbb{E}[X| \text{ choices for } v_1, \ldots, v_{t-1} \,] \geq \frac{|E|}{2}.$$

Show how to *deterministically, efficiently* make a choice for v_t so that

$$\mathbb{E}[X| \text{ choices for } v_1, \ldots, v_{t-1};\ \text{and } v_t \,] \geq \frac{|E|}{2}.$$

***Hint**: Let $\mathcal{S}_{t-1}$ denote the choices already made for $v_1, \ldots, v_{t-1}$. Consider*

$$\mathbb{E}[X|\mathcal{S}_{t-1}, v_t \in S] - \mathbb{E}[X|\mathcal{S}_{t-1}, v_t \notin S].$$

Which edges $\{u, v\} \in E$ contribute to this difference and how?

4. Building on your method above, design an algorithm to make a choice for v_1, and then v_2, and then v_3, and so on, so that eventually you have (efficiently, deterministically) found a set S so that at least $|E|/2$ edges cross the cut $(S, \bar{S})$.

Group Work: Solutions

1. We have $\mathbb{E}[X|v_1 \in S] = \mathbb{E}[X|v_1 \notin S]$ by symmetry. Then

$$|E|/2 = \mathbb{E}[X] = \frac{1}{2}\mathbb{E}[X|v_1 \in S] + \frac{1}{2}\mathbb{E}[X|v_1 \notin S],$$

so both $\mathbb{E}[...]$ on the right hand side must both be equal to $|E|/2$, since they are the same.

2. Let $A = \mathbb{E}[X|\text{choices for } v_1, \ldots, v_{t-1}, v_t \in S]$ and let $B = \mathbb{E}[X|\text{choices for } v_1, \ldots, v_{t-1}, v_t \notin S]$. Then

$$\begin{aligned}\frac{|E|}{2} &\geq \mathbb{E}[X|\text{choices for } v_1, \ldots, v_{t-1}] \\ &= \frac{1}{2}A + \frac{1}{2}B,\end{aligned}$$

so at least one of A, B must be $\geq |E|/2$.

3. Let's use the same notation A, B from the previous part. We know that at least one of A, B is at least $|E|/2$, so we just want to find which one. Thus, it suffices to find which of A, B is larger, and make the corresponding decision. We can write

$$A = \sum_{e \in E} \Pr[e \text{ crosses} | v_1, \ldots, v_{t-1}; v_t \in S]$$

and similarly

$$B = \sum_{e \in E} \Pr[e \text{ crosses} | v_1, \ldots, v_{t-1}; v_t \notin S]$$

Consider

$$A - B = \sum_{e \in E} \left(\Pr[e \text{ crosses} | v_1, \ldots, v_{t-1}; v_t \in S] - \Pr[e \text{ crosses} | v_1, \ldots, v_{t-1}; v_t \notin S] \right).$$

For each edge e, if neither of e's endpoints are v_t, that term will cancel in this difference. Similarly, if one of e's endpoints is v_t and the other is some v_j for $j > t$ (e.g., v_j hasn't been placed yet), then that term is $1/2 - 1/2 = 0$ and cancels. Thus, the only terms that survive are those for $e = \{v_t, v_i\}$ for $i < t$. In that case,

$$\begin{aligned} &\Pr[e \text{ crosses} | v_1, \ldots, v_{t-1}; v_t \in S] - \Pr[e \text{ crosses} | v_1, \ldots, v_{t-1}; v_t \notin S] \\ &= \mathbf{1}[v_i \in \bar{S}] - \mathbf{1}[v_i \in S]. \end{aligned}$$

Thus, the difference can be written

$$\begin{aligned} A - B &= \sum_{i<t} \mathbf{1}[\{v_i, v_t\} \in E](\mathbf{1}[v_i \in \bar{S}] - \mathbf{1}[v_i \in S]) \\ &= \left(\#\text{ nbrs of } v_t \text{ already placed in } \bar{S}\right) - \left(\#\text{ nbrs of } v_t \text{ already placed in } S\right). \end{aligned}$$

If this is positive, then A is bigger; if it's negative, then B is bigger. Thus, if v_t has more neighbors already placed in $\bar{S}$, we should place v_t in S; if v_t has more neighbors already placed in S, we should place v_t in $\bar{S}$. This makes sense, and is actually just the straightforward greedy algorithm! (But by doing it this way, we get to illustrate the method of derandomization via conditional expectations).

4.
- Order the vertices $v_1, \ldots, v_n$.
- Initialize $T, S = \emptyset$
- For $t = 1, \ldots, n$:
  - If $|\Gamma(v_t) \cap S| > |\Gamma(v_t) \cap T|$: $T \leftarrow T \cup \{v_t\}$
  - Else $S \leftarrow S \cup \{v_t\}$
- Return (S, T)

[Solutions and discussion of the general paradigm of derandomization via conditional expectation; see lecture notes and/or slides]

Group Work

1. 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. (If time) There is also a natural greedy algorithm for this problem:
   - For $i = 1, 2, \ldots, n$:
     - 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?

Group Work: Solutions

1. We apply the method of deranomization by conditional expectation! We choose values for the variables $x_1, \ldots, x_n$ one at a time, and as before, provided that we have
$$\mathbb{E}[\text{number of sat clauses}|x_1, \ldots, x_{t-1}] \geq 7m/8$$
there is a choice for x_t so that
$$\mathbb{E}[\text{number of sat clauses}|x_1, \ldots, x_t] \geq 7m/8.$$
Thus, all our algorithm has to do is find it. Let X be the number of satisfied clauses. Then we can write
$$\mathbb{E}[X|x_1, \ldots, x_t] = \sum_C \Pr[C \text{ sat.}|x_1, \ldots, x_t].$$
Observe that for each clause C, we can actually compute
$$\Pr[C \text{ sat.}|x_1, \ldots, x_t]$$
in time $O(1)$: that's because there are at most 8 outcomes. Thus, we can compute the whole thing in time $O(m)$. So our algorithm is:
   - For $t = 1, \ldots, n$:

- Compute $\mathbb{E}[X|x_1, \ldots, x_{t-1}; x_t = T]$
- If that is $\geq 7m/8$, set $x_t \leftarrow T$; otherwise set $x_t \leftarrow F$.

• Return $(x_1, \ldots, x_n)$.

The whole thing takes time $O(nm)$.

2. The "natural greedy algorithm" is worse than the algorithm from part (a). To see this, we can consider an example with $n = 8$ and $m = 8$. The algorithm from part (a) will come up with an assignment that satisfies 7 of the 8 clauses. The clauses are $(x_i \vee x_7 \vee x_8)$ for $i = 1, 2, \ldots, 6$, along with $(\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_4} \vee \overline{x_5} \vee \overline{x_6})$. Each of $x_1, \ldots, x_6$ appears in exactly two clauses, once as-is and once negated, so the natural greedy algorithm can take an arbitrary choice for each of these variables. If we choose to set all of them to TRUE, then each of the last two clauses will be unsatisfied, so the greedy algorithm will satisfy only 6 out of the 8 clauses, worse than the algorithm from part (a). (If you don't like that this is based on breaking ties arbitrarily, consider repeating this example M times for some large M, and repeating its negation $M - 1$ times; as M grows, this will still approach satisfying only 3/4 of the clauses, instead of 7/8).