Randomized Algorithms 2020-1
Lecture 9

The Lovasz Local Lemma *

Moni Naor

1 Recap

We stated by recalling several models of communication complexity (e.g. the simultaneous message model) and the complexity of the equality function in them. In general, with shared randomness (where the inputs are chosen without access to the shared randomness) the efficiency is much better ($O(1)$). Without shared randomness he complexity is $\Theta(\sqrt{n})$ in the simultaneous message model and $\Theta(\log n)$ in the one-round model.

We reviewed the structure of proofs by compression or encoding that are relevant for both lower bounds and upper bounds. In such a proof we consider a random string used in the system. This could be the random function that should be inverted in the example of function inversion or the randomness used by an algorithm (this will be the case with the Local Lemma). We then show that the event we want to bound implies an encoding of the randomness which compresses the representation. We know that the probability of compressing w bits (i.e. coming up with a representation that is w bit shorter than the length in a way that allows us to retrieve the original string) is at most 2^{-w}. This allows us to bound the probability of the bad event by 2^{-w}.

We also discussed Mirror Games and their relationship to card guessing [3, 2]. These are two player games where in the simplest case there is an even number of cards. The players take turns saying a name of a card and a player loses if this card was mentioned already by either one of the players. If all cards are finished, then it is a draw. The second player has a low memory strategy by mirroring the first player: fix a matching on the cards (e.g. the cards with the last bit flipped). For every move by the first player, respond by the matched card. You can view the matching as providing a subset in our setting and no bits need to be spent on it, since the current card indicates that the other has not arrived yet.

What Garg and Schnieder showed is that there is no deterministic sublinear strategy for the first player that is guaranteed to draw. The question is when can the first player have a reasonable chance of achieving a draw with little memory. Since it is a game, it is adversarial in nature, but on the other hand, half of the sequence itself is determined by the other player.

*These notes summarize the material covered in class, usually skipping proofs, details, examples and so forth, and possibly adding some remarks, or pointers. In the interest of brevity, most references and credits were omitted.

We mentioned that if there is a secret matching available to the first player, then this player can use the matching most of the time, but from time to time will need to find and unused pair (in case the second player hits the singleton). Te first player can apply the techniques of card guessing in order to find out a vacant pair. The expected number of times this happens is roughly $1/2H_{2n}$.

2 The Local Lemma

The Union Bound allow us to argue that certain events have a non-zero probability of not occurring simultaneously. Consider events $U = \{A_1.A_2, \ldots A_m\}$. For any event A_i the depdency on the other events is arbitrary. We know that

Theorem 1. *If there are* $0 \leq p_i \leq 1$ *s.t.* $\Pr[A_i] \leq p_i$ *and* $\sum_{i=1}^{m} p_i < 1$*, then the probability that no event* A_i *happens is positive.*

If all the $p_i = p$ then we need that $mp < 1$.

An example of an application is a CNF formula with m clauses where each clause has more than $\log m$ literals. Such a formula is necessarily satisfiable. Event A_i is clause i is not satisfied. $\Pr[A_i] < 1/m$.

The Local Lemma allows us to make such arguments based on local considerations only. You can read about in in either Alon-Spencer or Mitenmacher-Upfal.

We discussed two applications of the lemma. One was to argue that a CNF formula where every clause is of size k and every variable appears at most $2^k/ke$ clauses is necessarily satisfiable.

The proof we saw was non-constructive and for instance does not yield an algorithm that finds a satisfying assignment to a CNF. Next time we will see the algorithmic version of Moser and Tardos.

3 Hat Guessing

We saw an application of the lemma to hat guessing. Hat guessing puzzles are very common, and in recent years a variant related to graphs has become popular.

The following is taken almost verbatim from [4]. Suppose that n players are positioned on the vertices of a finite, simple graph G. An adversary puts a **colored** hat on each of their heads, in one of q colors. The players can only see the hats on their (immediate) neighbors' heads, and in particular no player sees their own hat. The players simultaneously guess the colors of their own hats, and they collectively win if any single one of them guesses correctly. The players may not communicate after the hat colors are assigned but may agree upon a strategy beforehand (and it could be a different strategy for every node). The hat guessing number $HG(G)$ of G is then the largest q for which the players have a winning strategy in the game with q colors.

If there is at least one edge in the graph, then $HG(G) \geq 2$. Why?

The classic case is when $G = K_n$, the complete graph on n vertices, and in this case $HG(K_n) = n$. A winning strategy is for the i-th player to guess the hat color (identifying colors with values

mod n) that would make the total of all the colors sum to $i \bmod n$. *Exactly one* player is correct in its sum and hence this player will be correct in its guess. Also for trees T it is known that $HG(T) = 2$, see Butler, Hajiaghayi, Kleinberg, and Leighton[1].

We showed that in any graph with maximum degree Δ we have that $HG(G) \leq e\Delta$. In other words, for every such graph, for every coloring strategy (which may specify for each node a different strategy) show that there is way to color the nodes with no more than $e\Delta$ colors such that the strategy fails: no node guesses its color correctly.

In more detail, let $G = (V, E)$ be a graph and let the neighborhood of a node $v \in V$ be $N(v)$. The strategy of the players is specified by a collection of functions, where for every node v there is a mapping from the colors of the neighbors to a color (that is the guess given that the neighbors are colored in certain way), $f_v : [q]^{|N(v)|} \mapsto [q]$. For any possible coloring with q colors of the nodes in V the strategist wants that there will be at lest one node v colored c with neighbors colored $c_1, c_2, \ldots c_{|N(v)|}$ s.t. $f_v(c_1, c_2, \ldots c_{|N(v)|}) = c$.

We showed that if $q > e\Delta$ then there is *no* such strategy by the players.

References

[1] S. Butler, M. T. Hajiaghayi, R. D. Kleinberg, and T. Leighton, *Hat guessing games*, SIAM J. Discrete Math.22(2008), 592–605.

[2] U. Feige, A randomized strategy in the mirror game, https://arxiv.org/pdf/1901.07809.pdf

[3] Sumegha Garg and Jon Schneider, The space complexity of mirror games, ITCS 2019.

[4] Hat Guessing Numbers of Degenerate Graphs, `https://arxiv.org/abs/2003.04990`