Lecture 8 notes #37

A Anupam Gupta STAFF
2 weeks ago in General

PIN STAR WATCHING 36 VIEWS

The lecture video for today is online. Some notes:

- Towards the end of today's lecture, we talked about the intuition for why the simple randomized algorithm gives a discrepancy of sqrt(|S_i| log m), whereas the Lovett-Meka algorithm gives a discrepancy sqrt(|S_i| log (m/n)).

  Here's an attempt at some more intuition: The simple randomized algorithm is a one-shot thing, and wants to color all elements while violating none of the discrepancy targets. So the sqrt(log m) term ensures that none of the m bad events happen.

  The Lovett-Meka algorithm, on the other hand, colors things slowly, and ensures that we do not violate a stronger target. Indeed, it tries to balance the rate at which elements are colored (i.e., become frozen) with the rate at which sets reach their targets (i.e., become dangerous). Since there are m sets and n elements, choosing \sqrt{log (m/n)} ensures that the two processes happen at about the same rate. So by the time we have made n/2 constraints tight, half of the tight constraints are for elements --- i.e., we have managed to color n/4 elements. Now we can recurse.

- Faisal was correct in saying that if $X_i \,|Y_1, \ldots, Y_{i-1}$ is a Rademacher, then it must be independent of X_1, ..., X_{i-1}. Indeed, this is true if $X_i \,|Y_1, \ldots, Y_{i-1}$ is any random variable whose parameters do not depend on the past, e.g., if it is a standard Gaussian N(0,1). But if we just say that $X_i \,|Y_1, \ldots, Y_{i-1}$ is a $N(0, \sigma_i^2)$ r.v., where σ_i can depend on the past, then now it's a Martingale.

- Zhaozi asked: do we need $\Omega(s \log(n/s))$ queries to achieve recovery of s-sparse signals. The answer is no: there are approaches that give you recovery for s-sparse signals using $2s$ queries. See, e.g., Ankur Moitra's notes on Prony's method. The advantage of using the basis pursuit approach is that it also works in "noisy" settings. E.g., when the answers have small amounts of noise. Or when the signal itself is not s-sparse, but most of its mass is on some s-coordinates. In these settings, basis pursuit (the L_1-minimization linear programming based approach) is still applicable, whereas other approaches do not necessarily work.

  It would be interesting to see matching lower bounds are known for noisy settings. (There is a lot of work in this area, so digging through it should help.)