CS5330: Randomized Algorithms Spring 2020

Options for Reading Projects

Prof. Arnab Bhattacharyya

The following are the recommended options for reading projects. If you have something else in mind, please get in touch with me and we can discuss.

1. **(Randomized Decision Trees)** *Decision trees* are arguably the simplest non-trivial computational model, where the cost of an algorithm is measured by the maximum number of input bits it reads. In past lectures, we have looked at randomized decision trees for the AND/OR tree function, and we saw how they make asymptotically less queries than the best deterministic decision trees. Recent work in this model has uncovered functions for which randomization offers even more speedup, and it's also known what the optimal speedup is. The answers differ depending on whether you allow error or not (Monte Carlo vs Las Vegas).

   See the following papers and the references therein:

   - *Complexity measures and decision tree complexity: a survey.* Harry Buhrman and Ronald de Wolf, 2002. (Outdated but contains basic definitions)
   - *Probabilistic boolean decision trees and the complexity of evaluating game trees.* Michael Saks and Avi Wigderson, 1986. (Seminal paper which defines randomized decision trees and analyzed the AND/OR tree function)
   - *Separations in Query Complexity Based on Pointer Functions.* Andris Ambainis, Kaspars Balodis, Aleksandrs Belovs, Troy Lee, Miklos Santha, and Juris Smotrovs, 2016.
   - *Separations between deterministic and randomized query complexity.* Sagnik Mukhopadhyay, Jaikumar Radhakrishnan, and Swagato Sanyal, 2018.

2. **(Discrepancy)** In the lecture on Chernoff bounds, we saw that for any n sets $S_1, \dots, S_n \subseteq \{1, \dots, n\}$, there is a 2-coloring of $\{1, \dots, n\}$ with discrepancy $O(\sqrt{n \log n})$. It is in fact possible to reduce the maximum discrepancy to $O(\sqrt{n})$ which is optimal. The original argument was non-constructive but recent work has contributed beautiful insights yielding polynomial time algorithms.

   See the following papers and the references therein:

   - *The Discrepancy Method.* Bernard Chazelle, 2000. (Great book that describes the state-of-the-art as of 2000.)
   - *Six standard deviations suffice.* Joel Spencer, 1985. (Seminal paper which gives non-constructive argument.)
   - *Constructive algorithms for discrepancy minimization.* Nikhil Bansal, 2010. (Breakthrough paper which gives a randomized polynomial time algorithm for $O(\sqrt{n})$ discrepancy bound. Quite complicated.)
   - *Constructive discrepancy minimization by walking on the edges.* Shachar Lovett and Raghu Meka, 2012. (A different algorithm and proof that is also much simpler than Bansal's result.)

- *Constructive discrepancy minimization for convex sets.* Thomas Rothvoss, 2014. (A third algorithm and proof that is also very simple and applies to a more general setting.)
- `https://windowsontheory.org/author/emeritusl/`. Blog posts by Raghu Meka, 2014. (A useful set of blog posts giving a high-level overview of the geometric approach to discrepancy minimization.)

3. **(Network Coding)** Network coding is the problem of simultaneously sending multiple messages between a source and destination in a network in the presence of failures. A breakthrough work in 2006 gave a randomized algorithm. This has been used later to devise new graph algorithms.

   - `http://www.cs.cmu.edu/~haeupler/15859F15/schedule.html`, notes for lectures 5-7, 2015.
   - *Graph connectivities, network coding, and expander graphs*, Ho Yee Cheung, Lap Chi Lau, and Kai Man Leung, 2013.

4. **(k-means Clustering)** A classic approach to clustering data points in Euclidean space is to find a small set of centers that minimizes the sum of squared distances between each point and its closest center. A very popular algorithm is k-means++ which consists of a randomized algorithm to find an initial set of centers followed by an iterative process known as Lloyd's algorithm. More recently, an algorithm k-means|| has been proposed that is parallelizable and hence scalable.

   - *K-means++: The advantages of careful seeding.* D. Arthur and Sergey Vassilvitskii, 2007.
   - *Scalable k-means++.* B. Bahmani, B. Moseley, A. Vattani, R. Kumar and S. Vassilvitskii, 2012. (Introduced scalable k-means++, a.k.a. k-means||.)
   - *Simple and sharp analysis of k-means*||. Václav Rozhoň, 2020. (A beautiful, clean and very recent analysis of k-means||.)

5. **(Tabulation Hashing)** We have seen k-wise independent hash families in the course and how they can be constructed using polynomials of degree $k-1$. In practice however, such implementations are quite slow. It has been found that a very efficient scheme known as *simple tabulation* can be used for applications, both in practice and in theory.

   - *Fast and Powerful Hashing using Tabulation*, Mikkel Thorup, 2017. (A nice survey of tabulation hashing by its master.)
   - *The power of simple tabulation hashing*, Mihai Patrascu and Mikkel Thorup, 2011. (The first paper which could prove that simple tabulation hashing is more powerful than its independence suggests.)
   - *Simple Tabulation, Fast Expanders, Double Tabulation, and High Independence*, Mikkel Thorup, 2013. (Shows that two applications of simple tabulation also leads to high independence.)

6. **(L_p Sampling in Data Streams)** A very useful task in the data streaming model is to sample from the stream where each item is sampled with probability dependent on its frequency. In L_p sampling, i is sampled with probability $|f_i|^p/\|f\|_p^p$ where f is the frequency vector. L_p sampling has found a wide range of applications relating to many fundamental streaming problems.

- *1-pass relative-error L_p-sampling with applications.* M. Monemizadeh and D.P. Woodruff, 2010. (The first paper which showed that L_p sampling is possible, albeit approximately.)
- *Perfect L_p-sampling in a data stream.* Rajesh Jayaram and David Woodruff, 2018. (Showed that approximation is not necessary. Also see references therein.)

7. **(Density Estimation)** The following problem arises often in machine learning: given independent samples from an unknown distribution p, choose among a set of candidate distributions $q_1, \ldots, q_k$ the one that is is closest to p. For example, $q_1, \ldots, q_k$ could be the output of a machine learning algorithm with different hyperparameters. Amazingly, it turns out that there are efficient algorithms for this problem without making any assumptions on $p, q_1, \ldots, q_k$ (when the notion of distance is the ℓ_1-norm.)

    - *Combinatorial methods in density estimation.* Luc Devroye and Gabor Lugosi, 1997. (Awesome book whose Chapter 6 is all about this problem and its solution by Yatracos.)
    - *The optimal density factor in density estimation.* Olivier Bousquet, Daniel Kane and Shay Moran, 2019. (Recent paper which shows you can go beyond the Yatracos bound if you allow improper algorithms. See also the references herein.)

8. **(Algorithmic Stability & Adaptive Data Analysis)** Suppose you have n samples $x_1, \ldots, x_n$ from a distribution p over a domain $\mathcal{D}$. You wish to answer a *statistical query*, i.e., a query of the form $\Pr_{x \sim p}[x \in S]$ where $S \subseteq \mathcal{D}$. The Chernoff bound shows that with $n = O(\varepsilon^{-2})$ samples, the empirical mean $\frac{1}{n} \sum_{i=1}^{n} 1[x_i \in S]$ is within additive error ε. Now, suppose there is a fixed set of k statistical queries; then applying the union bound, we see that $n = O(\varepsilon^{-2} \log k)$ samples suffice. However, if the queries are adaptively chosen based on responses to previous ones, it can be shown that one needs $n = \Omega(k\varepsilon^{-2})$ in order to guarantee that the responses are all within additive error ε. And indeed, in many real world scenarios, the analyst issues queries that are not pre-determined but based on past responses.

    Surprisingly, a work of Dwork et al. showed that it is possible to improve the sample complexity by simply adding gaussian noise to the empirical estimate. They were initially motivated by results in *differential privacy* but it is also possible to do the analysis more directly.

    - *The reusable holdout: Preserving validity in adaptive data analysis*, Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth, 2015. (An article in the popular journal *Science* for a non-CS audience. `https://arxiv.org/abs/1411.2664` by the same authors is the more technical paper showing sample complexity $\tilde{O}(\sqrt{k}/\epsilon^2)$.)
    - http://people.seas.harvard.edu/∼madhusudan/courses/Spring2016/notes/thomas-notes-ada.pdf, Thomas Steinke, 2016. (Clean and tight analysis of the gaussian mechanism proposed by Dwork et al.)
    - *Calibrating noise to variance in adaptive data analysis*, Vitaly Feldman and Thomas Steinke, 2018. (Gives a better bound when the standard deviation of the query is much smaller than its range.)
    - *Interactive Fingerprinting Codes and the Hardness of Preventing False Discovery*, Thomas Steinke and Jonathan Ullman, 2015. (Showed that the square-root relationship between n and k is tight.)

9. **(Sampling & Connectivity)** A classic result of Benczur and Karger, following previous works by Karger, states that any graph can be sparsified to a graph with $O(n \log n)$ edges while preserving the weights of all the cuts. The algorithm simply samples each edge e independently with a probability p_e where p_e can be computed in polynomial time. While these results deal with edge connectivity, vertex connectivity is a tougher research challenge.

   - *Using randomized sparsification to approximate minimum cuts*, David Karger, 1993.
   - *Approximating $s - t$ minimum cuts in $\tilde{O}(n^2)$ time*, András Benczúr and David Karger, 1996.
   - *Tight Bounds on Vertex Connectivity Under Vertex Sampling*, Keren Censor-Hillel, Mohsen Ghaffari, George Giakkoupis, Bernhard Haeupler, and Fabian Kuhn, 2015.

10. **(Constructive Lovász Local Lemma)** Although we didn't get the opportunity to cover this in class, the Lovász Local Lemma (LLL) is a basic tool in probabilistic analysis. It allows you to argue that the probability of an event is nonzero given certain conditions. Unlike the Chernoff bound, where you can typically choose parameters to make the probability close to 1 allowing for an efficient randomized algorithm, the LLL as traditionally stated only asserted that the probability is positive. In some beautiful works over the last 15 years, it has been shown how to make the LLL a constructive tool.

    - *A constructive proof of the Lovasz local lemma.* Robin Moser, 2008.
    - `https://terrytao.wordpress.com/2009/08/05/mosers-entropy-compression-argument`. Terence Tao, 2009. (A nice exposition of Moser's proof)
    - *A constructive proof of the general Lovasz local lemma.* Robin Moser and Gábor Tardos, 2009.
    - *New constructive aspects of the Lovasz Local Lemma.* Bernhard Haeupler, Barna Saha and Aravind Srinivasan, 2010.
    - *Algorithmic and enumerative aspects of the Moser-Tardos distribution.* David G. Harris and Aravind Srinivasan, 2016.

11. **(High-dimensional Expanders)** Expanders, covered in the last lecture of this module, are graphs that show properties similar to random graphs. There has been a recent surge of interest in exploring generalizations of expanders to hypergraphs. This area is still in a very nascent stage, so there are several different definitions of high-dimensional expanders in the literature, but already there have been some amazing applications.

    Note: If you are doing this project, you need to have some familiarity with topology.

    - `https://simons.berkeley.edu/workshops/schedule/10588`. (The first three talks give a very good introduction to the area.)
    - *Log-Concave Polynomials II: High-Dimensional Walks and an FPRAS for Counting Bases of a Matroid.* Nima Anari, Kuikui Liu, Shayan Oveis Gharan and Cynthia Vinzant, 2018. (Uses high-dimensional expanders to give an amazing analysis of Monte Carlo Markov Chain algorithms. Results in an efficient algorithm to sample a random base of a matroid.)

12. **(Testing Bayesian Networks)** A Bayesian network is a popular high-dimensional graphical model that describes how variables depend causally on each other. The question studied in these works is to determine whether a distribution belongs to a given class of Bayes nets or is far from each one of them. The question differs depending on whether the underlying graph is given or not.

    - *Testing Bayesian Networks*, Clement Canonne, Ilias Diakonikolas, Daniel Kane, and Alistair Stewart, 2016.
    - *Square Hellinger Subadditivity for Bayesian Networks and its Applications to Identity Testing.* Constantinos Daskalakis and Qinxuan Pan, 2016.
    - *Efficient Distance Approximation for Structured High-Dimensional Distributions via Learning.* Arnab Bhattacharyya, Sutanu Gayen, Kuldeep S. Meel, and N. V. Vinodchandran, 2020.