CS 599: Math for TCS, Spring 2022

Course Project Guidelines

The course project is an opportunity to perform an in-depth exploration of a course topic that interests you. The goals are to gain experience

- Independently reading and synthesizing research papers,
- Presenting research papers to an audience of your peers,
- Formulating and carrying out a short-term research project, and
- Critically evaluating written technical work.

You are not expected to produce publishable results by the end of the semester, but hopefully you will be able to lay the groundwork for something that can be further developed into publishable research. Projects may be done individually or in pairs, with paired projects strongly encouraged. Some examples of possible projects include:

- Making progress on an open theoretical question related to one of the course topics,
- Applying techniques from one fo the course topics toward a problem in an area that interests you,
- A survey synthesizing several related research results concerning one of the course topics or applications of those techniques,
- A self-contained exposition of a mathematical tool not discussed in class and its application to a problem in theoretical CS, or
- Something totally different! Be creative!

At the bottom of this document are references to some papers relevant to the course topics. These are only intended for inspiration. You are highly encouraged to find your own papers and topics, especially if they relate to your research interests. Directly related to the course is better than not directly related to the course. Helpful to your research is MUCH better than not helpful to your research. Please feel free to discuss your ideas with me and with each other throughout the semester, in office hours, by email, or on Piazza. Auditors are welcome to complete projects and collaborate with registered students.

To keep you on track with your project, there are several intermediate milestones. If you are working with a partner, you may submit a single set of any of these components.

Topic Ideas (Friday 3/25): Submit one to three possible topic ideas, with a paragraph describing each. For each topic, include the general question that you may like to address and list of relevant research papers that you plan to read. Submitting these ideas will encourage you start thinking about your project early, allow you to identify students with similar interests with whom you might like to collaborate, and let me give you preliminary suggestions for how to proceed.

Project Proposal (Friday 4/15): Submit a page or two outlining what your project will look like. At this stage, you should be able to clearly state your research questions, describe how your project relates to prior work on the topic, and describe your plan of attack for addressing these questions.

Project Report (Wednesday 5/11): Submit a paper (about 8–10 pages) describing your completed project. The paper should motivate your research questions and results, explain how the project fits into the context of previous work, and present the results in a clear and convincing manner.

Peer Feedback (Friday 5/13): I will randomly assign everyone a peer's anonymized report to provide a few paragraphs of respectful and constructive feedback. What did the author(s) do well? What could they improve? Are the proofs convincing? Is enough intuition provided to support the technical statements? Are there any places that are unclear or where special knowledge is assumed? Is there a a compelling outline for future work that could be done on this project?

These project and presentation guidelines are adapted from those used in other courses, including Jelani Nelson's "Algorithms for Big Data," Toni Pitassi's "Communication Complexity: Applications and New Directions," and especially Salil Vadhan/Jon Ullman's "Mathematical Approaches to Data Privacy."

Project/Presentation Topic Ideas

Below are a number of suggestions for possible project and presentation topics with a few representative references (with a bias toward recent) for each. These are just meant as inspiration to get you started thinking. You are in no way obligated to stick to these topics and are encouraged to explore topics that interest you. The references provided are not exhaustive. Many of these papers are just ones I happen to know for serendipitous reasons. You will want to do additional research to find the state-of-the-art on each topic or locate background references. Some good places to start are the references in the texts we use for assigned reading and the talks and proceedings of algorithms/theory conferences like STOC, FOCS, SODA, CCC, ITCS, and APPROX/RANDOM.

Boolean Fourier Analysis

- Towards a proof of the Fourier-Entropy conjecture? Kelman, Kindler, Lifshitz, Minzer, Safra, 2020.
- Properly learning decision trees in almost polynomial time. Blanc, Lange, Qiao, Tan, 2021.
- Learning low-degree functions from a logarithmic number of random queries. Eskenazis and Ivanisvili, 2022.
- Hypercontractivity on high-dimensional expanders. Gur, Lifshitz, Liu, 2022; Bafna, Hopkins, Kaufman, Lovett 2022.

Pseudorandomness

- Pseudorandom generators for read-once branching programs, in any order. Forbes, Kelley, 2018.
- Fooling polytopes. O'Donnell, Servedio, Tan, 2018.
- Better pseudodistributions and derandomization for space-bounded computation. Hoza, 2021.
- Fractional pseudorandom generators from any Fourier level. Chattopadhyay, Gaitonde, Lee, Lovett, Shetty, 2021.

Spectral Graph Theory

- Explicit, almost optimal, epsilon-balanced codes. Ta-Shma 2017.
- High-precision estimation of random walks in small space. Ahmadinejad, Kelner, Murtagh, Peebles, Sidford, Vadhan 2020.

- Spectral independence in high-dimensional expanders and applications to the hardcore model. Anari, Liu, Oveis Gharan, 2020.
- Spectral hypergraph sparsifiers of nearly linear size. Kapralov, Krauthgamer, Tardos, Yoshida, 2021.

Coding

- Explicit binary tree codes with polylogarithmic size alphabet. Cohen, Haeupler, Schulman, 2018.
- Arikan meets Shannon: polar codes with near-optimal convergence to channel capacity. Guruswami, Riazanov, Ye, 2020.
- LDPC codes achieve list-decoding capacity. Mosheiff, Resch, Ron-Zewi, Silas, Wooters, 2020.
- Fiber bundle codes: Breaking the $N^{1/2}\text{polylog}(N)$ barrier for quantum LDPC codes. Hastings, Haah, O'Donnell, 2020.
- Locally testable codes with constant rate, distance, and locality. Dinur, Evra, Livne, Lubotzky, Mozes, 2021.

Combinatorics

- Bill Gasarch's compendium: `http://www.cs.umd.edu/~gasarch/TOPICS/ramsey/ramsey.html`
- Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. Dell, van Melkebeek, 2010.
- Private PAC learning implies finite Littlestone dimension. Alon, Livni, Malliaris, Moran, 2019.
- Improved bounds for the sunflower lemma. Alweiss, Lovett, Wu, Zhang, 2020.
- Worst-case to average-case reductions via additive combinatorics. Asadi, Golovnev, Gur, Shinkar, 2022.

LP/SDP Hierarchies

- Sum-of-squares meets program obfuscation, revisited. Barak, Hopkins, Jain, Kothari, Sahai, 2018.
- Subexponential LPs approximate Max-Cut. Hopkins, Schramm, Trevisan, 2020.
- A stress-free sum-of-squares lower bound for coloring. Kothari, Manohar, 2021.

- Sum-of-squares lower bounds for sparse independent set. Jones, Potechin, Rajendran, Tulsiani, Xu, 2021.
- Robust linear regression: optimal rates in polynomial time. Bakshi, Prasad, 2021.
- On semi-algebraic proofs and algorithms. Fleming, Göös, Grosser, Robere, 2022.

Mathematical topics we (probably) won't cover Dimensionality reduction (e.g., Johnson-Lindenstrauss), interlacing families of polynomials, Sylvester-Gallai theorem, combinatorial quantities in learning theory (e.g., VC dimension), matrix rigidity, cap set problem (see `https://www.ams.org/journals/bull/2019-56-01/S0273-0979-2018-01648-0/S0273-0979-2018-01` `pdf`)