525 Project Guidelines Instructor: Anna Karlin

This quarter you will be doing a project in CSE 525; 20% of your grade for this course will derive from this project. Ideally, this will be a project on a topic that is of great interest to you, and better yet, if it's relevant to your work/research in some way.

The plan is for you to pick a paper on randomized algorithms or probabilistic analysis from the theoretical computer science literature, something that appeared in a conference (such as FOCS, STOC, SODA, ICALP, ITCS, COLT, PODC, etc), or something from one of the main journals (e.g., SICOMP, JACM, Random Structures and Algorithms). Ideally, the paper should be relatively recent, say published in the last 20 or so years. And of course, the paper should not be something I will be covering in class.

You are encouraged to do the project in groups of **2 students** (but no more).

Project Options

You can choose from one of the following 3 types of project.

- Option 1: Write a summary of the paper, containing the following items:
  - Introduction: problem definition, motivation, applications, and a brief summary of prior work on the problem.
  - Main contributions: the main results of the paper, the algorithm or model and the implications of the results.
  - Overview of the proof: Give a high-level overview of the proof and the main ideas. What is the novelty of the paper which makes it different from the previous work?
  - Details of the proof: Go over the main steps and the main lemmas of the proof, explaining it in your own way. Make an effort to add value to the published version: more intuition, extra examples, details filled in, clarifying figures, clearer exposition, etc.
  - You may not copy/past lemmas/proofs from the paper.
- Option 2: Empirical study. In most theory papers, there is no empirical evaluation. If you choose this option, you should plan on implementing the algorithm proposed in the paper and running it against some test cases and possibly alternative algorithms. You may generate your own test cases or you may use actual data from real world applications. It would be ideal if you compare the performance of the algorithm with a "greedy" algorithm or any reasonable algorithm that comes to your mind. Please write a report describing the algorithm proposed in the paper, a description of your proposed algorithm and the details of your findings. Another version would be to explore a random process that has been studied theoretically via simulation, better yet attempting to determine the answer to a question that is difficult to solve theoretically. In both cases, you will be submitting your code and your test data.

- Option 3: Application to a different research area. In this case, the goal is for you to relate the paper to your own field of study. For example, you may modify the algorithm suggested in the paper and add a heuristic on top of it and run it against a real world problem that interests you. In your report, please describe the algorithm in the paper and your modifications. Then, explain how this can be used to solve a problem in your research area or to model a phenomenon that you are interested in. Then, describe the test data and the performance of your proposed solution for those test cases. Theoretical justification for your proposed solution would make this a great project.

In all three cases, you should make an effort to include a discussion of interesting open problems related to your topic (whether theoretical or applied).

Deadlines and evaluation

- Each group should turn in a 2-paragraph preproposal by **Monday, February 15** on gradescope. This pre-proposal must include the citation for the paper you will be working from. Your pre-proposal should also include a description of the type of project you will be doing from among the options listed above. If you are planning to do options 2 or 3, please provide some details on your plan.

- The final 7-10 page paper on your project will be due **Tuesday, March 16**. You will **post it to edstem.** The title of your edstem post should be your name(s) and the subject of your project.

- The paper will be evaluated on clarity, correctness, completeness, and depth. Your paper should be easy to read and not assume any background beyond what we've done in class (or what was assumed for class). Assume that the reader will have no knowledge about the topic and you are trying to give them intuition and a great entry point to it.

- By **Friday, March 19**, each student will read a writeup from one of the other groups (these will be assigned at random) and will provide a substantial public comment about it on the discussion board.

  Comments can include a discussion of points that you found unclear and why, questions about the results, suggestions for further related research, citations to relevant papers, etc. Comments need not be more than the equivalent of about a page long (and less is okay – we're looking for quality, not quantity.)

- The grading breakdown will be:

  - Preproposal: 5%
  - Paper: 75%
  - Comments on peer's paper: 20%

Possible topics

There are so many exciting topics that we do not have time to cover this quarter and every topic that we are covering is being done only at a superficial level. So on pretty much any topic we are touching upon, there are many interesting papers we are not covering that you could look into.

Resources - courses, etc

Here are some courses/lecture notes/surveys etc that might help you get started on a topic. Note that some of them are not specifically on randomized algorithms or probabilistic analysis, but contain many topics that are relevant.

- TCS+ talks: This is a collection of online seminars in theoretical computer science. Most of them involve probability and randomization. A great entree into reading a paper would be watching one of these talks and then diving into the associated paper.
- Beyond worst-case analysis (Tim Roughgarden)
- Algorithms for big data (Moses Charikar)
- Semidefinite programming and approximation algorithms (Lap Chi Lau)
- Algorithms for Massive Data (Alex Andoni)
- Introduction to Multi-armed Bandits (Aleksandrs Slivkins)
- Property Testing (Dana Ron)
- Metric embeddings and algorithmic applications (Moses Charikar)
- Machine Learning Theory (Avrim Blum)
- Spectral graph theory (David Williamson)
- Random graphs (Frieze and Karonski)
- Markov Chain Monte Carlo Methods (Eric Vigoda)

Resources - possible broad topics

Here are some topics you could explore (some of which are discussed in the above courses):

- sketching and streaming
- property testing
- algebraic methods, e.g. network coding
- more on concentration and tail bounds, e.g. negative association
- other types of randomized rounding algorithms (e.g. dependent rounding
- compressive sensing
- learning theory
- smoothed analysis
- randomized online algorithms

- spectral sparsification
- low distortion embeddings, sparsest cut, etc.
- discrepancy
- interactive proofs
- random graphs and sharp thresholds
- probabilistically checkable proofs
- approximate counting