Sublinear Algorithms for Big Data

Fall 2024

Instructor: Jasper Lee, `jasperlee@ucdavis.edu`

Time and Location: Tuesdays and Thursdays, 16:40–18:00 at Teaching and Learning Complex 3214

Collaboration Hours: TBD. Details on Piazza.

Website: `https://jasperchlee.github.io/courses/sublinear/F24`

Piazza: `https://piazza.com/class/m0obqsmqseloz/` (Accessible via Canvas)

Textbook: The course will not be following any particular textbooks. However, the following texts may be helpful for understanding course material (though possibly far too advanced for our scope) and as related reading:

- *Introduction to Property Testing* by Oded Goldreich, published by the Cambridge University Press and a draft pdf available on his webpage
- *Data Stream Algorithms* course lecture notes by Amit Chakrabarti, available at `https://www.cs.dartmouth.edu/~ac/Teach/data-streams-lecnotes.pdf`

Links to related courses, other relevant lecture notes and further reading will be posted to the course webpage.

Description

A huge quantity of data is worth little unless we can extract insights from it. Yet, the large quantities mean that classic algorithms (running in linear, quadratic or even more time) can be infeasible in practice. We must instead turn to new algorithmic approaches and paradigms, which allow us to answer valuable questions about our data in runtime that is still feasible even when the data set is Facebook-sized.

Surprisingly, to answer many computational and statistical questions, sometimes there is no need to read/store every piece of data! This course focuses on this exciting "sublinear" algorithmic regime. We will study practical algorithms, making clever use of randomness with strong theoretical guarantees, on the following (tentative and non-exhaustive) list of topics:

- Testing and learning information about structures such as lists (e.g. testing approximate sortedness/monotonicity) and massive graphs (e.g. testing approximate connectedness), using very few queries into the structure
- Learning information about a probability distribution from a small number of independent samples, e.g. deciding whether the distribution of users on your online platform has changed substantially since you introduced a UI change
- Computation on high-volume streams of data, whilst only maintaining a small memory buffer, e.g. approximately estimating the number of unique visitors to a webpage over a given time period without storing them one by one

Prerequisites

This is an advanced undergraduate level course that is also suitable for graduate students.

The key prerequisites are

1. Mathematical maturity, as we will prove most if not all of the covered results both in class and on homeworks
2. Familiarity with basic probability (you should be comfortable applying Markov's and Chebyshev's inequalities, and understand how a random walk can be characterized by a matrix)
3. Familiarity with basic analysis of algorithms

More concrete course prerequisites are

- ECS 20 or equivalent, and
- ECS 132 or MAT 135A or STA 130A or equivalent, and
- ECS 122A or ECS 222A or equivalent

Mathematical maturity is essential to succeed in (and necessary to pass) this course. We will prove almost all of the results in class, and most homework problems involve writing rigorous proofs.

Learning Goals

The primary goals of the course are to 1) introduce students to the models and problem formulations for sublinear algorithms and related topics, 2) give basic tools for students to read, understand and implement existing results, and 3) enable students to communicate these results effectively.

To achieve the first goal, this course surveys 3 main areas in sublinear algorithms: 1) combinatorial property testing and learning, 2) distributional property testing and learning, and 3) streaming algorithms. There are also lectures planned to cover related topics, such as dimensionality reduction and a brief introduction to robust statistics.

In surveying these algorithmic results, relevant design and analysis tools are introduced, which are applicable in general for understanding papers in the area.

The homeworks, lecture note scribing (as described below) and final project all complement the lectures to help students master the key concepts and tools. Furthermore, they are the primary venues for developing the students' communication skills for these sophisticated and technical results.

Lastly, for students intending to go onto graduate studies in theoretical computer science, the mathematical and paper-reading techniques from this course will hopefully form part of their research toolkit.

Course Outline

The following is a rough course outline. See the course webpage for the detailed lecture plan and schedule.

Introduction to Probability Tools (Week 1)

Sublinear algorithms on combinatorial structures (Weeks 2–3)

Sublinear algorithms on probability distributions (Weeks 4–6)

Sublinear-space streaming Algorithms (Weeks 7–9)

Miscellaneous Topics (Week 10)

Grading

Grade components may be curved by the instructor before a final grade is calculated.

Homeworks	50%
Participation+Scribing	20%
Final Project	30%
Extra Credit problems on homeworks	Maximum of 10%

Homework Assignments

Homeworks will be released weekly, with instructions on submission and deadlines. With the exception of Homework 0 which should be submitted but will not be formally graded, all homework submissions contribute to the homework component of the final grade as detailed in the previous section. Students will pair up in writing their homework submissions, again except only for Homework 0.

There are extra-credit homework problems, which students are encouraged to solve, for further mastery of the material. Please refer to the instructions on each homework.

Assignments should be typeset neatly in LaTeX or written in clearly legible handwriting. Unreadable submissions may not be graded.

Collaboration Policy and Academic Honesty

Collaboration on homework sets (with others in addition to your partner) is not only permitted, but also encouraged. To maximize learning, we suggest you first try solving the problems on your own, before exchanging and brainstorming ideas with your classmates. You must however write out all solutions only with your homework partner(s). On each homework, please state who you discussed the problems with.

You are also allowed to consult other sources, for example resources on the Internet, for alternative explanations of concepts and results covered in the course, as well as related reading materials. Inevitably, some of you will (intentionally or unintentionally) stumble upon solutions to homework problems. In order to have an enforceable collaboration policy, the bottom line is that you must

write out all the solutions in your own words, demonstrating that you at least understand the solution you have written. Under this policy, whilst you *are* allowed to just search for problem solutions online (if the problem is standard enough), and rewrite the solutions in your own words, it is of course heavily discouraged for the sake of *your* education.

If you do happen to base your answer (even if only partially) on outside sources, please also cite them. This is for the instructor's (and the class') benefit, to see what online sources may be useful.

For the purposes of this collaboration policy, treat Large Language Model (LLM) generated outputs as the same as other online sources. To cite an LLM output for its content, give the model name/version as well as the prompt used. If you used LLM to polish your writing (although also highly discouraged), you should cite only the model name/version.

Failure to comply with this lax collaboration policy, that is if you submit something that you plagiarized without demonstrating any understanding, results in a bad course grade and/or a report to OSSJA regardless of performance in the rest of the course.

Late Policy

Except with the prior approval of the instructor, any submission that is late for no more than 48 hours will get a 20% reduction on that homework's grades, and any submission more than 48 hours late will not be graded.

Permission for late submissions must be requested **at least 24 hours in advance**, and will only be granted in exceptional/extenuating circumstances or for religious observances. Contact the instructor directly for such requests.

Participation and Scribing

Participation and scribing constitutes 20% of the final grade. In addition to participating in class by joining in discussions and asking/answering questions, each student is expected to act as the lecture note scribe for at least 1 lecture. As with the homework assignments, scribing will also be done in pairs. Students may be asked to scribe more than once, depending on the enrolment numbers.

Lecture notes should be typeset using a supplied LaTeX template, and submitted to the instructor in a timely manner. Late submissions may have a 5% deduction (amongst the 20%), depending on the pacing of the course at the time and at the instructor's discretion, taking into account the reasons for lateness. Each submission will be awarded 10% for any non-trivial attempt that includes all components of the lecture. The other 10% is awarded for clear and readable notes. Extra credit may be given for notes that are particularly well-written and/or include materials supplementing the lecture, such as completing proofs of assumed results.

The instructor will give timely feedback on the submission, such that the students can revise and improve their notes for a higher grade, with *no* penalty.

Students should not see scribing as a burden, but an educational opportunity in line with the learning goals of the course. Scribing lecture notes is good practice for explaining technical ideas, with a mechanism to get feedback on writing, as well as a way to consolidate understanding before putting such understanding into words.

Final Project

As stated in an earlier section, the primary learning goals of the course include teaching students to read, understand, and communicate results in sublinear algorithms and related areas. To that end, the final project is to produce a written report/survey on one or more results in the area, allowing students both to 1) learn about new models/results and to 2) demonstrate their ability to understand and communicate these ideas.

Students are expected to read at least one sublinear algorithmic result and 1) implement it (if there is sufficient scope and complexity in the algorithm), reporting on the empirical findings, and/or 2) write up the analysis of the algorithm in their own words.

Graduate students and strong undergraduate students are also encouraged to read multiple research papers or a single sufficiently complex result, and produce a coherent survey on the analyses and relationship between the different results.

Projects are graded both on their scope and on the quality of the write-up. A high quality report/survey will have 1) clear writing style, 2) well-organized logical flow to explain technical results and 3) a distillation of the intuition behind the algorithm design and analysis.

The topic choices are subject to the instructor's approval. Students are expected to complete their projects individually, unless the students gain prior approval from the instructor for a project with significantly larger scope (e.g. a writeup of a proof of the PCP theorem). Further details will be communicated to the class during the second half of the course.

Tentatively, the final project is due on 9 Dec 2024 at 23:59 (Pacific time).

Accommodations (SDC and Religious Observance)

If you have any disabilities (of any form and type), or any existing or new medical conditions that could affect your learning and ability to complete the coursework, please contact SDC or a dean to discuss. Let us know as soon as possible if you require any accommodations, attaching a relevant Dean's note or SDC email. The staff will support you as best as we can.

Students with religious observance conflicting with the course schedule should endeavour to inform the instructor within the first 4 weeks of the semester, and no later than 1 week in advance, in order for us to make suitable arrangements.

Mental Health

Being a student can be very stressful. If you feel you are under too much pressure or there are psychological issues that are keeping you from performing well at UC Davis, we encourage you to contact Student Health and Counseling Services. They provide confidential counselling.

`https://shcs.ucdavis.edu/`

Coping with Unforeseen or Difficult Circumstances

If there are events that are upsetting to you, whether political, family-related, weather-related, etc., that affect your ability to do well in class, we are happy to take them into account with respect to our late policy. Please feel free to talk to the instructor about this. Additionally, the Office of Student Support and Judicial Affairs (`https://ossja.ucdavis.edu/`) can be a helpful resource for discussing current concerns and academic and personal plans.

Contacting the Staff

For any sensitive matters, you should feel free to contact the instructor or departmental staff directly, ignoring the instructions below.

- For technical questions about course content: post on Piazza, or better yet, go to hours
- For SDC accommodations: ask SDC to contact the instructor
- For requests for deadline extensions: email the instructor
- For grading concerns: email the instructor
- For concerns about the final grade: email the instructor
- For diversity and related concerns: email the instructor, or Professors Dipak Ghosal, Kurt Eiselt