STA 250: Theoretical Foundations for Machine Learning

Syllabus - Spring 2025 (as of March 30, 2025)

Basic Course Information

Instructor: Dogyoon Song

- *Email:* dgsong@ucdavis.edu
- *Office hours:* Mondays 4–5 PM, or by appointment (*Office:* MSB 4220)

Time and Location: Mondays & Wednesdays, 2:10–3:50 PM, Wellman Hall 109

Course Description

This course provides a rigorous introduction to the theoretical foundations of machine learning from both statistical and optimization perspectives. Primary topics to cover include generalization theory, optimization theory and methods, and theoretical aspects of deep learning. As time permits, we will also briefly survey selected additional topics and modern developments, such as causal machine learning and diffusion generative models. The overarching goal is to equip students with the analytical tools and frameworks necessary to develop and evaluate machine learning methods in their research.

Learning objectives

- Develop sufficient mathematical background to read and understand theoretical machine learning papers. Early course materials will cover core concepts and techniques from statistics and optimization, which students will apply through homework assignments and the final project.
- Learn to critically assess recent publications to identify impactful, novel future research directions. Students will practice this in paper discussion sessions and the project, drawing on both theoretical reasoning and/or experimental investigation.
- Gain experience formulating and outlining approaches to prove hypothesized theorems. This skill will be practiced in lectures, homework exercises, and potentially through the final project.
- Strengthen the ability to present problem motivations, insights, and results to a technical audience. This will be cultivated via paper discussions, the final presentation, written reports, and peer evaluations.

Prerequisites: Designed primarily for Ph.D. and master's students in Statistics, Mathematics, or Computer Science, this course assumes a strong mathematical background and enthusiasm for theoretical analysis. While there is no formal prerequisite, familiarity with basic probabilistic reasoning, proof-based methods, and quantitative analysis is essential. The course will be primarily proof-based but will include some programming in Python (e.g., PyTorch/JAX/TensorFlow). Students should also be comfortable with basic machine learning concepts and proof-based linear algebra and probability theory.

Texts and resources We have no official course textbooks, but you may find the following resources useful.

- Francis Bach, "Learning Theory from First Principles," MIT Press
- Martin Wainwright, "High-Dimensional Statistics," Cambridge University Press
- Stephen Boyd, "Convex Optimization," Cambridge University Press

- Percy Liang, Statistical Learning Theory, Stanford Lecture Notes
- Tengyu Ma, Machine Learning Theory, Stanford Lecture Notes
- Sanjeev Arora, "Theory of Deep Learning," Book Draft
- Matus Telgarsky, Deep Learning Theory, Lecture Notes (new version and old version)

Online platforms:

- The course webpage will host lecture notes, homework assignments, references, and additional materials.
- Canvas will be used for submitting all homework, paper discussion reports, and project slides/reports.
- Piazza will be used for discussion. Students are encouraged to ask and discuss each other on Piazza.

Course Topics and Tentative Schedule

Topics: We will cover the following subjects, which may change due to time constraints or student interests.

- Supervised learning and generalization theory
- Optimization theory and methods
- Deep learning theory
- Additional topics (tentative): Causal machine learning, (Deep) generative models, ...

Course schedule: The table below outlines a tentative schedule, which is subject to revision depending on time and student preferences.

Week	Topics	Homework	Reading group	Project
1	Supervised learning & ERM	HW 0 due		
2	Generalization theory			
3	Optimization	HW 1 due	Discussion 1	Milestone 1
4	Optimization			
5	Optimization	HW 2 due	Discussion 2	Milestone 2
6	Deep learning theory			
7	Deep learning theory	HW 3 due		
8	Additional topics		Discussion 3	Milestone 3
9	Additional topics			
10	Project presentation & report			Milestones 4 & 5

Reading group sessions (tentative):

- Discussion 1: Mon, April 14
- Discussion 2: Wed, April 30
- Discussion 3: Wed, May 21

Project milestones:

- Milestone 1 (pre-proposal): Fri, April 18
- Milestone 2 (proposal): Fri, May 2
- Milestone 3 (mid-term project meetings): sometime mid- or late-May
- Milestone 4 (presentation: Mon, June 2 & Wed, June 4
- Milestone 5 (final report): Mon, June 9

Assessments & Grading

Grading scheme: The students' performance in this course will be evaluated based on the following:

- **Homework** (30%)
- **Reading group discussions** (25%)
- **Term project** (45%)

Cutoffs for letter grades follow the standard UC Davis grading scheme (e.g., A $\geq$ 90%, B $\geq$ 80%, etc.), though these thresholds may be adjusted in stdents' favor at the end of the quarter.

Homework (30%): There will be three homework assignments (not including "Homework 0," which you need not submit). "Homework 0" is meant for self-assessment; no solutions will be provided. If you struggle with completing any of the non-programming part of "Homework 0," then please note the course may require significant extra effort on your part.

Each of the three graded homeworks will roughly align with each of the first three core units, offering essential practice for internalizing the concepts and methodologies covered in class and for exploring material beyond lectures. A random subset of problems will be graded, and clarity of writing counts alongside correctness.

You are welcome to collaborate with other students on your homework, but you must list the names of any collaborators at the top of your homework assignment. All final write-ups must be done individually, and submissions must be in LaTeX-produced PDFs via Gradescope (accessible through Canvas).

If you are unfamiliar with software packages like PyTorch/Jax/TensorFlow, you may find the following tutorials helpful:

- PyTorch: Learn the Basics
- Introduction to PyTorch - YouTube Series

All of the coding you will need to do for the course (and tutorials above) can be done on your personal laptop or in Google Colab. A GPU will not be needed, although using the Google Colab GPU may help if you want to do an experiment-heavy project.

Reading group discussions (25%): We will hold approximately three sessions to read and discuss an influential paper related to the course. The format follows a role-based discussion inspired by Colin Raffel and Alec Jacobson's role-playing student seminars, which is also used in Aditi Raghunathan's course).

- **Format.** Researchers in machine learning often attend reading groups to learn about recent research as well as to discover new problems to work on. In these discussions, we will provide additional structure whereby students adopt different "roles" that they take, taken from the following roles: positive reviewer, negative reviewer, archaeologist, academic researcher, industry practitioner, hacker, ...
- **Participation.** Every student must participate in at least one paper presentation, which will have 1-2 students per role for 2-4 roles depending upon enrollment. Further details will be provided in due time.
- **Evaluation.** The grade for the reading group discussions will be determined as follows:
  - Presentation (50%)
  - Non-presenter summaries (40%)
  - In-class participation (10%)

For more details on the reading group, please see the course webpage.

Term project (45%): Students will read 2 or more papers on a topic of their choice, then identify and formulate an interesting, concrete follow-up question for future research. Students are expected to make initial strides in solving these questions, via theory or experiments.

Process. At a high level:

- Students may work solo or in teams of up to 2 people.
- Select at least 2 papers of their interested topic, read them thoroughly, and formulate novel and significant follow-up questions to the paper.
- Make initial attempts toward solving these research questions, possibly by breaking them down into subproblems, via theory or experiments.

Project grading rubric. Your grade on the project will be evaluated by the following rubric.

- **Milestone 1: Pre-proposal (5%).** Submit a **one-page pre-proposal** listing team members, your initial topic, chosen papers, and preliminary research questions or conjectures. The aim of this pre-proposal is to provide you early feedback and help in your exploration.
- **Milestone 2: Proposal (15%).** Submit a **proposal up to 3 pages** that includes:
    1. A brief summary of the chosen papers, highlighting their significance, strengths, and weaknesses
    2. The motivation for your proposed follow-up work, including its relevance and tractability.

  The proposal should be understandable to someone else taking the class, and should also have logistical information about how you plan to work on the formulated research questions. An outstanding proposal might also contain some preliminary research efforts (a theoretical approach or supporting experiments).
- **Milestone 3: Mid-term project meeting (10%).** Each group will meet with me in mid or late May to discuss progress and receive feedback if needed.
- **Milestone 4: Project presentation (20%).** Present slides for about 20 minutes (15 min presentation + 5 min Q&A). It is typically more effective to highlight key ideas—e.g., by simplifying the math and omitting less important contents—rather than crowding slides with exhaustive technical details.
- **Milestone 5: Project report (50%).** Submit a written project report in LaTeX-generated PDF format, using the NeurIPS 2025 style files. The report should be **at most 9 pages** in the main section, with an unlimited number of pages for references and the appendix; however, appendix may not be carefully read. A concise/succinct report which clearly communicates your work and findings is better than a long-winded one.
- **Attendance and participation.** Attendance on presentation days is mandatory. For each missed presentation day, you will lose 5 percentage points from the project.

For additional details, please see the course webpage.

Course Policies

Attendance & participation

Although attendance is not mandatory, it is crucial for your success to attend each lecture and actively study on your own. Classes will be held in person and will not be recorded unless otherwise specified. If you miss a lecture, you are responsible for any material covered and announcements made in your absence; in that case, studying on your own and attending office hours is recommended if you have further questions.

Code of conduct & academic integrity

All students are expected to follow the UC Davis Code of Academic Conduct. Violations include (but are not limited to) collaborating or communicating during exams, copying or allowing someone to copy graded assignments, doing someone else's homework/exam (or having someone do yours), sharing assignments/exams, and submitting work that is not your own. The fact that the violation did not benefit you directly does not diminish its seriousness.

Under the UC Davis Code of Academic Conduct, faculty are required to report suspected academic misconduct to the Office of Student Support and Judicial Affairs (OSSJA). Any violation will be reported, and students found guilty will receive an F, regardless of the extent or type of violation. A first offense leads to a failing grade on the relevant assignment or exam; a second instance results in a failing grade in the course. Any student who cheats on an assignment or exam will be referred to the Office of Student Support and Judicial Affairs. More information on academic dishonesty and UC Davis policy is available on the OSSJA website. Please do not violate the code of conduct—remain vigilant and avoid any misconduct.

Use of large language models (LLMs)

You may use LLMs such as ChatGPT, Grok, etc. for your learning purposes (e.g., to generate examples and debug your code) but you are responsible for any errors in the outputs. Moreover, ***their direct use in preparing your submissions, such as homework, paper discussion reports, and project-related submissions, is not permitted.*** You are also required to acknowledge in your submission how you used LLMs, along with the prompts you used, if you indirectly benefited from them. Any suspected violation may be reported to OSSJA.

Distribution of course materials

Please do not distribute any course materials outside of this class, as doing so results in an infringement of copyright as per UC policy. Use of sites like Course Hero and Chegg is not permitted. You may take notes, make copies of course materials for your own use, and share them with other students who are auditing this course. However, you may not reproduce, distribute, or display (post/upload) lecture notes or course materials in any other way—whether or not a fee is charged—without the instructor's express prior written consent. You also may not allow others to do so. Violations may result in student conduct proceedings under the UC Davis Code of Academic Conduct.

Accommodations for students with disabilities

UC Davis is committed to educational equity in the academic setting, and in serving a diverse student body. All students who are interested in learning more about the Student Disability Center (SDC) are encouraged to contact them directly at `https://sdc.ucdavis.edu`, `sdc@ucdavis.edu` or (530) 752-3184. If you are a student who requires academic accommodations, please submit your SDC Letter of Accommodation as soon as possible, ideally within the first two weeks of class.

Changes to syllabus

This version of the syllabus is current as of March 30, 2025. The instructor may update it based on student progress or shifting instructional priorities. Any updates will be communicated promptly.