CSCI 5525 Machine Learning: Analysis and Methods

Fall 2024

General Information

What are the state-of-the-art machine learning methods and why do they work? This graduate-level introductory course to machine learning focuses on the foundations of modern machine learning. We will cover selected topics from supervised learning, unsupervised learning, generative learning, and interactive learning. For each topic, we will describe key algorithmic ideas/intuitions and basic theoretical insights. By the end of the course, students will master main machine learning techniques, and apply/optimize/develop them for specific applications.

- **Prerequisite:** Introduction to machine learning or equivalent (CSCI5521 or 5523) are **required**. Maturity in linear algebra, multivariate calculus, and basic probability is assumed. Familiarity with Python (esp. Numpy, Scipy) is necessary to complete the homework sets and the final project.
- **When & Where:** Tue/Thur 2:30–3:45pm, Mechanical Engineering 108
- **Who:** Professor Ju Sun (Instructor) Email: jusun@umn.edu
  Leon Luo (TA) Email: luo00042@umn.edu
- **Office Hours:** Thur 4–6pm (Instructor) Mon 3–4pm (TA)
- **Course Website:** https://sunju.org/teach/ML-Fall-2024/ All course materials, including course schedule, lecture slides and notes, homework assignments, project description, will be posted on the course website. Enrolled students are encouraged to check the website on a regular basis.
- **Communication—minimizing emailing: Piazza** (linked from inside our Canvas website) is the preferred and most efficient way of communication. Please post all questions and discussions related to the course there instead of sending emails, and make them public if possible to minimize duplicate questions and answers. If you have to use emails, please begin the subject line with "CSCI 5525", and expect delay in response.

Tentative Topics

The list of topics is subject to change.

- Overview
- Linear prediction
- Support vector machines and kernel methods
- Elements of statistical learning theory
- Ensemble methods: boosting and random forests
- Dimension reduction

- Clustering
- Generative models
- Online/Reinforcement learning
- Basics of neural networks

Teaching Format

In-person only, but with UNITE option (for UNITE students).

Video streaming and recording UNITE live streaming will be available to UNITE students immediately (perhaps with seconds of delay). UNITE (recorded) videos will be available to **all registered students** with a **7-day** delay.

Recommended References

There is no required textbook. Most lectures will be whiteboard presentations, and you are strongly encouraged to attend lectures and take notes. Separate lecture notes will be distributed to aid in understanding. Recommended reference books are the following, all of which are freely available either online or through library e-access.

- Advanced machine learning textbooks
  - **Foundations of Machine Learning** (2e) by Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. https://cs.nyu.edu/~mohri/mlbook/
  - **Understanding Machine Learning: From Theory to Algorithms** by Shai Shalev-Shwartz and Shai Ben-David. https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/
  - **The Elements of Statistical Learning: Data Mining, Inference, and Prediction** by Trevor Hastie, Robert Tibshirani, and Jerome Friedman. https://web.stanford.edu/~hastie/ElemStatLearn/
  - The **Probabilistic Machine Learning** series by Kevin P. Murphy. https://probml.github.io/pml-book/
- Mathematics foundations
  - **Mathematics for Machine Learning** by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. https://mml-book.github.io/
  - **Linear Algebra and Optimization for Machine Learning** by Charu C. Aggarwal. https://rd.springer.com/book/10.1007/978-3-030-40344-7 (UMN library access; login required)
- Hand-on in Python
  - **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems** by Aurélien Géron (2ed). O'Reilly Media, 2019. click here (UMN library access; login required)

Assessment

- Homework 60%: 5 Homeworks (15% each), the worst score will be dropped automatically (i.e., you can skip one homework if needed)
- Course project 40%: proposal (10%) + progress report (10%) + final report (20%). The project can be a survey of a chosen topic not covered in detail in the class, a comparison of existing methods, or novel foundational or applied research.
- **Bonus points** 5%: 1% for 3 good questions and answers in Piazza (marked by the instructor and the TA)
- Final grades on a curve, but will be assigned *generously*

Homework

You have approximately 14 days (i.e., 2 weeks) to complete each homework (the exact due date will be specified in each assignment). **Late submissions will not be accepted**. All submissions *must* be electronic and uploaded via the Canvas system. The written part should be neatly written/scanned or typeset and *must* be submitted as PDF files. For students pursuing research in relevant areas, it is strongly encouraged that you type your solution using LaTeX. Computer programs *must* be submitted in the Python notebook format. Only Python 3 will be used and accepted in this course.

Collaboration on homework problems is strongly encouraged, but each student must ensure that the final submission is prepared individually. **Collaborators should be properly acknowledged in the final submission, at the problem level.** The same applies to computer programs. Plagiarism and cheating are not tolerated and are subject to disciplinary action. Please consult the student code of conduct for more information: https://regents.umn.edu/sites/regents.umn.edu/files/2019-09/policy_student_conduct_code.pdf

About the use of AI tools You are strongly encouraged to collaborate with AI tools—they are becoming our workspace friends, such as ChatGPT (https://chat.openai.com/) and Claude (https://claude.ai/chats), and Github Copilot (https://github.com/features/copilot), when trying to, e.g., solve homework problems and come up with project ideas. It takes a bit of practice to ask the right and effective questions/prompts to these tools; we highly recommend that you go through this popular free short course **ChatGPT Prompt Engineering for Developers** offered by https://learn.deeplearning.ai/ to get started.

Our catch-it-or-miss-it policy **If you use any AI tools for your homework problems, you are required to include screenshots of your prompting questions and their answers in your writeup**. The answers provided by such AI tools often contain factual errors and reasoning gaps. **So, if you only submit an AI answer with such bugs for any problem, you will obtain a zero score for that problem**. You obtain the scores only when you find the bugs and also correct them in your own writing. You can also choose not to use any of these AI tools, in which case we will grade based on the efforts you have made.

Course Project

The course project is to be carried out by teams of 3 students. All students in the same team will get the same score for their course project. We will give more details about the project later of the semester.

Programming and Computing

Our programming environment will be Python 3. For deep learning frameworks, PyTorch (≥ 2.0) is preferred, but Tensorflow (≥ 2.0) is also accepted and supported. For small-scale experiments, which will be the case for typical homework problems, Google Colab (`https://colab.research.google.com/`) and UMN MSI notebook service (`https://msi.umn.edu/about-msi-services/interactive-hpc/jupyter-notebooks`) will suffice. For power users, local installation of the relevant software packages may be a reasonable alternative. For large-scale course projects, we can use the Minnesota Supercomputing Institute (MSI) GPU computing queues, based on our class account (to be activated soon).

Other Policies

Please consult this policy link `https://policy.umn.edu/education/syllabusrequirements-appa`