Syllabus for Computational Machine Learning

Course name: Computational Machine Learning
Course number: CSCI-GA-3033-012
Course credits: 3
Room: CIWW 201
Time: 7:10pm-09:00pm

Course Description: Machine learning studies the question how can we build computer programs that automatically improve their performance through experience? This course primarily focuses on computationally-efficient machine learning, i.e. theoretically-grounded methods and algorithms to analyse large amounts of data with successful performance. Such large scales arise in applications such as information retrieval, social recommendation, computer vision, genomics, astrophysics. The course will cover both the theoretical and methodological aspects, and a variety of real-world applications.

Course Instructor: Zaid Harchaoui, zh13@nyu.edu
Office hours: Tuesday, 4-5pm; Thursday 11-1pm (please book an appointment here)
Office: Room #1208, 12th Floor, 715 Broadway. Take the elevator on the right (if asked, say you come to see me, and sign at the desk)

Academic Term in which course is given: Fall 2015

Contact Hours: 14-week semester. Each week comprises 2 hours of lectures, and 1 hour of tutoring session if enough students enrol. Course staff will be available for office hours for at least 2 hours per week. Course staff will also be available through the online Piazza page: https://piazza.com/nyu/fall2015/csciga3033012/home#

Course aims and objectives:
• Teach key concepts and tools of machine learning with algorithm runtime/scalability in mind
• Outline the fundamental dimensions and their order of magnitude in machine learning problems
• Provide hands-on experience in designing and programming machine learning algorithms
• Provide a basis for advanced study of machine learning and data science

Prerequisites:
• Fundamental Algorithms (CSCI-GA.1170)
• Mathematical Techniques for Computer Science Applications (CSCI-GA.1180)

General prerequisites:
• Solid mathematical background, equivalent to a 1-semester undergraduate course in each of the following: linear algebra, multi-variable calculus, probability, statistics
• Python/Matlab programming required for all homework assignments (not necessary for auditors as it will only be lightly mentioned in lectures)
• Recommended: Computer science background up to a course in data structures and algorithms
• Recommended: At least one advanced, proof-based mathematics course
• Desirable: undergraduate data science or machine learning class, or hands-on experience dealing with data for analysis/prediction/modelling (e.g. speech processing, computer vision, natural language processing, modelling of observational data, etc.). Students have to be familiar with "least-squares regression" and "1-nearest-neighbor classification".
Note that some prerequisites may be waived with permission of the instructor after discussion. Please attend the 1st course and discuss with the instructor.

Tentative List of Topics By Week:

- Week 1: Overview of Machine Learning, Evaluation Criterion, Loss Function, Empirical Risk
- Week 2: Perceptron, Neocognitron, Performance metrics, Linear Predictors, Capacity, Regularization
- Week 3: Ridge Regression, Logistic Regression, Linear Support Vector Machines, Boosting
- Week 4: Convex Learning Problems, Stochastic Gradient Descent
- Week 5: Faster Stochastic Gradient Descent, Model Selection, Validation
- Week 6: Kernel-based Methods, Boosting
- Week 7: Decision Trees, Random Forests, Ensemble Methods
- Week 8: Deep Neural Networks, I
- Week 9: Quantization, Clustering, Compression
- Week 10: Dimensionality Reduction
- Week 11: Latent Variable Models
- Week 12: Feature Selection, Feature Design
- Week 13: Deep Neural Networks, II
- Week 14: The Art of Machine Learning Modelling

Time permitting, we may be able to cover some of the following additional topics: gaussian processes, ranking problems, collaborative filtering, Bayesian inference, bandit problems (A/B testing, Thompson sampling, UCB methods), statistical generalization bounds. All of these are accessible topics for a class at this level. In any case, ambitious students are encouraged to seek my guidance in pursuing these topics on their own.

Method of assessment:

- There may be some short in-class quizzes (loosely graded) counted toward participation credit.
- There will be roughly 4-5 homework assignments with both written and programming parts. Some assignments may have extra credit opportunities in the form of optional questions or short surveys on special topics. Homeworks are due at noon on the date specified. Homeworks will still be accepted after this time-date but will have a 10% penalty per hour late.
- Midterm Exam: The exam will cover material from lectures, homework, and assigned readings up to the week before the exam.
- Final Project: Final projects will be done in groups of two students. The project will typically involve either a new data source, or doing something new with a well-known data source. More methodological or theoretical projects are also possible. In any case, the project must have some degree of “figuring out the approach”, rather than just implementing or comparing known methods.

Grading: The final numerical score will be the weighted average of assignment score, the midterm exam, and the final project.

Main bibliography:

Other useful resources:


Disclaimer: The professor reserves to right to make changes to the syllabus, including project due dates and test dates. These changes will be announced as early as possible.

Academic Integrity Policy: The course conforms to NYU’s policy on academic integrity for students: [http://www.nyu.edu/about/policies-guidelines-compliance/policies-and-guidelines/academic-integrity-for-students-at-nyu.html](http://www.nyu.edu/about/policies-guidelines-compliance/policies-and-guidelines/academic-integrity-for-students-at-nyu.html)

This policy prohibits plagiarism and cheating.

- Plagiarism: presenting others’ work without adequate acknowledgement of its source, as though it were one’s own. Plagiarism is a form of fraud. We all stand on the shoulders of others, and we must give credit to the creators of the works that we incorporate into products that we call our own. Some examples of plagiarism:
  - a sequence of words incorporated without quotation marks
  - an unacknowledged passage paraphrased from another’s work
  - the use of ideas, sound recordings, computer data or images created by others as though it were one’s own

- Cheating: deceiving a faculty member or other individual who assess student performance into believing that one’s mastery of a subject or discipline is greater than it is by a range of dishonest methods, including but not limited to:
  - bringing or accessing unauthorized materials during an examination (e.g., notes, books, or other information accessed via cell phones, computers, other technology or any other means)
  - providing assistance to acts of academic misconduct/dishonesty (e.g., sharing copies of exams via cell phones, computers, other technology or any other means, allowing others to copy answers on an exam)
  - submitting the same or substantially similar work in multiple courses, either in the same semester or in a different semester, without the express approval of all instructors
  - submitting work (papers, homework assignments, computer programs, experimental results, artwork, etc.) that was created by another, substantially or in whole, as one’s own
  - submitting answers on an exam that were obtained from the work of another person or providing answers or assistance to others during an exam when not explicitly permitted by the instructor
  - submitting evaluations of group members’ work for an assigned group project which misrepresent the work that was performed by another group member
  - altering or forging academic documents, including but not limited to admissions materials, academic records, grade reports, add/drop forms, course registration forms, etc.

Authors of Syllabus: Zaid Harchaoui, Yann LeCun, David Rosenberg, David Sontag