CO673/CS794: Optimization for Data Science

Fall 2020

TTh 10:00 – 11:20 (AOE)

Web: https://cs.uwaterloo.ca/~y328yu/mycourses/794

Piazza: https://piazza.com/uwaterloo.ca/fall2020/co673cs794/home

Syllabus

Instructor

Yao-Liang Yu
DC 3617
Assistant Professor, School of Computer Science, University of Waterloo
CIFAR AI Chair, Vector Institute

Web: https://cs.uwaterloo.ca/~y328yu
Email: yaoliang.yu@uwaterloo.ca
Office hours: TBD or by email appointment

Course Description

Techniques for formulating data science models as optimization problems. Algorithms for solving data science problems with an emphasis on scalability, efficiency and parallelizability including gradient-descent based algorithms, derivative-free algorithms, and randomized algorithms. Theoretical analyses of algorithms and approaches for recognizing the most suitable algorithm for solving a particular problem.

Course Objectives

Optimization algorithms has been an essential tool in many scientific and engineering fields, particularly so in data science where large amounts of data need to be processed quickly. This has caused a paradigm shift from conventional optimization algorithms such as interior-point methods to the old and classic first-order gradient methods, with many new extensions and analyses being derived/rediscovered recently. The main goal of this course is to introduce the students to these new/old algorithms (which are easy to implement and understand), to appreciate the corresponding theoretical analyses, and to understand which algorithm is good for solving what kind of data science problems.

The course will start from basic definitions and theorems in convex analysis, emphasizing the fundamental role of convex duality. Then, we are going to analyze an arsenal of gradient algorithms, each motivated using an appropriate machine learning application. Towards the end of the course, the students will be able to

- recognize and formulate convex, nonconvex, and structured optimization problems;
- understand the basic theory and algorithms for solving large-scale problems;
- gain some programming experience on numerical implementations;
- use the material in current or future research.

Course Overview

While the nature of this course is mostly algorithmic and theoretical, we complement each lecture, as shown in parenthesis, with current applications in machine learning and data science to motivate the algorithm and analysis.

- Introduction (ML applications)
- Optimization Basics
- Gradient Descent (flow)
- Projected Gradient (white-box adversarial attack)
- Subgradient (svm)
- Proximal Gradient (lasso)
- Conditional Gradient (recommendation system)
- Coordinate Gradient (graph clustering)
- Lower Bound I
- Acceleration (total variation denoising)
- Mirror Descent (reinforcement learning)
- Smoothing (robust svm)
- Proximal Average (federated learning)
- Splitting (robust PCA)
- Alternating (VAE)
- EM (mixture models)
- Minimax (adversarial training)
- De-acceleration (GANs)
- Lower Bound II
- Stochastic Gradient (Boltzmann machine)
- Derivative-free (black-box adversarial attack)
- Lower Bound III
- Variance Reduction (boosting)
- Backpropagation (neural nets)
- Maximum Principle (neural ode)

Prerequisites

Knowledge of linear algebra (vector space, eigenvalue, matrix multiplication, etc.), multivariate calculus (derivative, gradient, etc.), basic analysis (convergence, limits, etc.), and basic probability (common distributions, means, etc.). Exposure to numerical computing or machine learning is a plus but not required. Knowledge of programming in either Python or Julia (can learn in a few hours).

Textbooks

There is no required textbook. We will pose lecture notes or slides before class. You are encouraged to check out the following excellent books.

- *Lectures on Convex Optimization (2nd)*. Yurii Nesterov. Springer, 2018.
- *First-Order Methods in Optimization*. Amir Beck. SIAM, 2017.
- *Optimization for Machine Learning*. Edited by Suvrit Sra, Sebastian Nowozin and Stephen J. Wright. MIT, 2011.
- *Introduction to Optimization*. Boris Polyak. Optimization software, 1987.
- *Foundations of Optimization*. Osman Güler. Springer, 2010.

Grading

There will be 5 homework assignments, each worth 20% of your final grade. We will calculate the top 4 assignments into your grade hence in total 4 x 20% = 80%. Assignments will be posted on the course webpage and announced on course piazza page. Expect to have 1 assignment every 2 weeks (roughly). For programming questions, choose one of the following two languages: Python or Julia.

Completed assignments (and project proposal and report) will be submitted through LEARN. Submit early and often!

As usual, it is OK to seek for help, but you must write your solutions independently and individually, and you should always acknowledge any help you get (book, friend, internet, etc.).

Mark appeals should be requested within two weeks of receiving the mark. The appeal could go either ways, so request only if you truly believe something is wrong.

Late Policy

We do **NOT** accept any late homework submissions, unless you have a legitimate reason with formal proof (e.g. hospitalization, family urgency, etc.). Traveling, busy with other stuff, or simply forgetting to submit, are not considered legitimate.

Project (20% of the final grade)

Students are expected to conduct a research project: your project could be a survey of a class of optimization algorithms for data science and machine learning, or an empirical comparison of several related algorithms on an interesting dataset, or an application of the discussed algorithms to a new field/application, or designing a novel algorithm to address a need in machine learning, or theoretically analyzing the performance of an optimization algorithm (new or old). Some possible projects will be suggested as we progress in the course, but you are highly encouraged to choose your own project (that interests you the most).

You project should

- relate to the course (obviously)
- allow you to learn something new (and hopefully significant)
- be interesting and nontrivial, preferably publishable in a top conference/journal

The project proposal will be due immediately after the reading week (tentatively). Please concisely describe what your project is about, what are the related works, what is your execution plan, what do you expect to learn/contribute, and how are you going to evaluate your results. I expect the proposal to be less than **3 pages** (excluding references).

You are allowed to form a team of at most 3 members. However, you need to justify why this is necessary and clearly describe who is going to do what in your project proposal. Each team member will get the same mark for the project (irrespective of what is actually accomplished by each member). So, choose your teammate wisely! It is the team's responsibility to make sure each member is on track. There will be no extra credit for doing everything alone.

The project report will be due shortly after the last day of the term. Please summarize all your findings (empirical, algorithmic, theoretical) in a scientific report. I expect there is an introduction section, a background section, a main result section, and a conclusion section. Depending on your project, you may include an experimental section and/or discussion section. Please always give proper citations to prior work or results. Be precise and concise. I expect the report to be less than **8 pages** (excluding references).

Your project report will be evaluated by its clarity, significance, rigor, presentation, and completeness.

Academic Integrity

In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. Check the university website for more information.

Grievance

A student who believes that a decision affecting some aspect of hisher university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Student Petitions and Grievances, Section 4. When in doubt please be certain to contact the department's administrative assistant who will provide further assistance.

Discipline

A student is expected to know what constitutes academic integrity to avoid committing an academic offence, and to take responsibility for hisher actions. A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (e.g., plagiarism, cheating) or about "rules" for group work/collaboration should seek guidance from the course instructor, academic advisor, or the undergraduate Associate Dean. For information on categories of offences and types of penalties, students should refer to Policy 71, Student Discipline. For typical penalties check Guidelines for the Assessment of Penalties.

Avoiding Academic Offenses

Most students are unaware of the line between acceptable and unacceptable academic behaviour, especially when discussing assignments with classmates and using the work of other students. For information on commonly misunderstood academic offenses and how to avoid them, students should refer to the Faculty of Mathematics Cheating and Student Academic Discipline Policy.

Appeals

A decision made or penalty imposed under Policy 70 (Student Petitions and Grievances) (other than a petition) or Policy 71 (Student Discipline) may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72 (Student Appeals).

Note for Students with Disabilities

AccessAbility Services, located in the new addition to Needles Hall, Room 1401, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without compromising the academic integrity of the curriculum. If you require academic accommodations to lessen the impact of your disability, please register with the office at the beginning of each academic term.

Intellectual Property

Students should be aware that this course contains the intellectual property of their instructor, TA, and/or the University of Waterloo. Intellectual property includes items such as:

- Lecture content, spoken and written (and any audio/video recording thereof);
- Lecture handouts, presentations, and other materials prepared for the course (e.g., PowerPoint slides);
- Questions or solution sets from various types of assessments (e.g., assignments, quizzes, tests, final exams); and
- Work protected by copyright (e.g., any work authored by the instructor or TA or used by the instructor or TA with permission of the copyright owner).

Course materials and the intellectual property contained therein, are used to enhance a student's educational experience. However, sharing this intellectual property without the intellectual property owner's permission is a violation of intellectual property rights. For this reason, it is necessary to ask the instructor, TA and/or the University of Waterloo for permission before uploading and sharing the intellectual property of others online (e.g., to an online repository).

Permission from an instructor, TA or the University is also necessary before sharing the intellectual property of others from completed courses with students taking the same/similar courses in subsequent terms/years. In many cases, instructors might be happy to allow distribution of certain materials. However, doing so without expressed permission is considered a violation of intellectual property rights.

Please alert the instructor if you become aware of intellectual property belonging to others (past or present) circulating, either through the student body or online. The intellectual property rights owner deserves to know (and may have already given their consent).