CSC411/2515 Fall 2018 — Course Information
Machine Learning and Data Mining

Course web site: `http://www.cs.toronto.edu/~rgrosse/courses/csc411_f18/`

Overview

Machine learning is a set of techniques that allow machines to learn from data and experience, rather than requiring humans to specify the desired behavior by hand. Over the past two decades, machine learning techniques have become increasingly central both in AI as an academic field, and in the technology industry. This course provides a broad introduction to some of the most commonly used ML algorithms. It also serves to introduce key algorithmic principles which will serve as a foundation for more advanced courses, such as CSC412/2506 (Probabilistic Learning and Reasoning) and CSC421/2516 (Neural Networks and Deep Learning).

The first half of the course focuses on supervised learning. We begin with nearest neighbours, decision trees, and ensembles. Then we introduce parametric models, including linear regression, logistic and softmax regression, and neural networks. We then move on to unsupervised learning, focusing in particular on probabilistic models, but also principal components analysis and K-means. Finally, we cover the basics of reinforcement learning.

Schedule

See the course web page for information about lecture and tutorial times, and for the schedule of lecture and tutorial topics.

Prerequisites

Because we make heavy use of calculus, probability, and linear algebra, this course has some substantial prerequisites.

- **Calculus:** MAT(135H1,136H1)/MAT137Y1/MAT137Y1/MAT157Y1
- **Linear Algebra:** MAT221H1/MAT223H1/MAT240H1
- **Probability:** STA247H1/STA255H1/STA257H1
- **Data structures and analysis:** CSC263H1/CSC265H1

And some optional, but helpful, background:

- **Multivariable calculus (recommended):** MAT235Y1/MAT237Y1/MAT257Y1

- **Numerical methods (recommended):** CSC336H1/CSC350H1
- **Statistics (recommended):** STA248H1/STA250H1/STA261H1

Teaching Team

Instructors: Roger Grosse, Amir-massoud Farahmand, and Juan Carrasquilla
e-mail: `csc411.f18.instructors@cs.toronto.edu`
Office hours: Mon 10am–noon in Pratt 290F (That's the D. L. Pratt Building, not the E. J. Pratt Library!)

Teaching Assistants:
James Lucas (head TA)
Aryan Arbabi
Jesse Bettencourt
Elliot Creager
Chris Cremer
Mohammad Firouzi
Jonathan Lorraine
David Madras
Bret Nestor
Punit Shah
Guodong Zhang

Staff e-mail: `csc411.f18.staff@gmail.com` (goes to TAs and instructors)
TA office hours: TBA

Do NOT send course-related e-mails to our personal e-mail accounts. Use the official instructor and staff e-mails instead.

Please use Piazza for any questions about the course content which don't give away any homework hints. That way, everyone in the class can benefit from the answers.

Load

There are 24 hours of lectures and 10 hours of tutorials.

Textbook

There is no required textbook for the class. A few small readings may be assigned if the need arises. These required readings will all be available on the web, for free.

There are also some relevant resources which are freely available online. We will try to provide links on a lecture-by-lecture basis.

- The classic statistical learning textbook, *The Elements of Statistical Learning*, by Hastie, Tibshirani, and Friedman.
  `https://web.stanford.edu/~hastie/Papers/ESLII.pdf`
- David MacKay's excellent textbook, *Information Theory, Inference, and Learning Algorithms.* This isn't focused on neural nets per se, but it has some overlap with this course, especially the lectures on Bayesian models.
  `http://www.inference.phy.cam.ac.uk/mackay/itila/`
- *Metacademy*, an online website (which one of the instructors is involved with) which helps you construct personalized learning plans and which has links to lots of resources relevant to particular concepts. We'll post links to relevant Metacademy concepts as the course progresses.
  `http://www.metacademy.org`
- Video lectures for UofT Professor Geoffrey Hinton's Coursera course on deep learning. Professor Hinton is one of the fathers of the field, so think of these as the Feynman Lectures of neural nets.
  `https://www.youtube.com/playlist?list=PLoRl3Ht4JOcdU872GhiYWf6jwrk_SNhz9`
- *Deep Learning*, a textbook by Yoshua Bengio, Ian Goodfellow, and Aaron Courville.
  `http://www.deeplearningbook.org/`
- Andrej Karpathy's lecture notes on convolutional networks.
  `http://cs231n.github.io/`

Marking Scheme

The marking scheme for undergraduates is as follows:

- **45%:** 8-10 "weekly" Assignments, equally weighted
  - Combination of pencil & paper derivations and short programming exercises
  - Lowest homework mark is dropped
- **15%:** Midterm
  - One hour, time TBA
- **35%:** Final Exam

- Three hours

- **5%:** Read some classic papers.
  - Honor system: if you indicate to us that you read the papers, you will get the points.

- You must achieve a minimum mark of 30% on the final exam to pass the course.

Graduate students may elect to follow the undergraduate requirements and marking scheme. This is the “path of least resistance” for the course. However, graduate students may elect to do a final project, which will replace the homework assignments for the second half of the term. However, **everybody must take the final exam, including graduate students**. If this option is chosen, the marking scheme will be as follows:

- **20%:** First 4 “weekly” Assignments, equally weighted
  - Combination of pencil & paper derivations and short programming exercises
  - Lowest homework mark is dropped

- **25%:** Final project
  - See final project handout for details.

- **15%:** Midterm
  - One hour, time TBA

- **35%:** Final Exam
  - Three hours

- **5%:** Read some classic papers.
  - Honor system: if you indicate to us that you read the papers, you will get the points.

Academic Integrity

By the time you get to an advanced course like csc411 you’ve heard this lots of times, so we’ll keep it brief: avoid academic offenses (a.k.a. cheating). All graded work in this course is individual work.

Written Homeworks

In order to give you additional practice with the material, we assign written homeworks, which give you additional practice with the course content and encourage you to keep on top of the material. Roughly speaking, there will be one homework due each week that doesn't have another assignment or test. Each one consists of 2-3 conceptual questions and is meant to take a few hours.

Dates. Weekly homeworks will typically be due at 11:59pm on Wednesdays. See the course web page for particular deadlines. Each homework is due a minimum of one week after the relevant course material was introduced.

Format. Weekly homeworks must be submitted in PDF format through MarkUs. We encourage typesetting using LaTeX, but scans of handwritten solutions are also acceptable as long as they are legible.

Lateness. Weekly homeworks will be accepted up to 3 days late, but 10% will be deducted for each day late, rounded up to the nearest day.

Weighting. In aggregate, the homeworks count for 45% of the total mark for the course, so individually they count for roughly 5% each. The lowest homework grade for the term will be dropped.

Collaboration policy. You are expected to work on the homeworks by yourself. You should not discuss them with anyone except the tutors or the instructor. The report you hand in should be entirely your own work and you may be asked to demonstrate how you got any results that you report.

Remarks. Remark requests should be made through MarkUs, and will be considered by the same TA who marked the assignment. The deadline for requesting a remark is typically one week after the marked assignments are returned.

Tests

Exams are closed-book, and focus on the material introduced during lecture. More details will be provided during the term.

Missed tests. Missed tests will get a score of 0 except in the case of an official Student Medical Certificate or prior approval by the instructors. In the latter case, the request must be made at least one week in advance of the exam date.

Online forum

We'll use Piazza for the course forum. The URL will be sent out to the class mailing list.

Auditing

If you are not registered in the class, it is possible for you to audit it (sit in on the lectures). Here are the official university rules on auditors (taken from the Department of Computer Science instructor's advice page):

> To audit a course is to sit and listen to the lectures, and perhaps to the tutorials, without formally enrolling. Auditing is acceptable *if the auditor is a student at U of T, and no University resources are to be committed to the auditor.* The "must be a student" condition means that students of other universities, employees of outside organizations (or even of U of T itself!), or any other non-students, are not permitted to be auditors. (If we did not have this rule, the University would require us to collect auditing fees, and we are not willing to do that.)
>
> The "no resources used" condition means that auditors do not get computing accounts, cannot have term work marked, and cannot write exams. In other words, they cannot use instructors time, TA time, or administrative resources of any kind.
>
> An auditor may not attend class *unless there is an empty seat after the last regularly-enrolled student has sat down.* That sounds frivolous, but in fact it is an aspect of an important point: if enrollment in a course has been closed because the room size has been reached, then there may well be physical seats for auditors, because it is rare for every student to appear for a lecture, but auditors will not be allowed to enroll later on in the course, even if some students drop it. Neither instructors nor the department can waive this rule.
>
> Often these conditions are perfectly acceptable to auditors; we don't mean to ban the practice, but only to live within the University's rules.