Advanced Topics in Machine Learning: CMSC 498V 18')

Instructor: Furong Huang (furongh@cs.umd.edu)
TA: Jingling Li (jingling@cs.umd.edu)
August 28, 18

http://www.cs.umd.edu/class/fall2018/cmsc498V/

Course Introduction
- Topics and Prerequisite

Course Intro

- Undergraduate level, research oriented course
- Read papers, discuss, brainstorm, collaborate
- Student expect:
  - Learn ML knowledge, scribing skills, presentation skills
  - Course project -> research project -> academic paper
- Instructor expect:
  - Students are highly motivated: read papers, discuss
  - Interesting research ideas, collaboration (student-student, student-instructor)

Topics

- The PAC Learning Framework
  - The PAC learning model
  - Guarantees for finite hypothesis sets - consistent/inconsistent case
  - Generalities
- Rademacher Complexity and VC-Dimension
  - Rademacher complexity
  - VC-dimension
  - Lower bounds
- Boosting and Margins Theory

Tentative syllabus to be adjusted to accommodate students needs

Topics

- Guaranteed Spectral Methods in Unsupervised Learning
    - Latent Variable Models
    - Dictionary Learning
    - Method of Moments
    - Tensor and Tensor Methods
    - Jennrich's Algorithm
    - Power Method

Tentative syllabus to be adjusted to accommodate students needs

Topics

- Generalization of Neural Networks
    - PAC learning framework
    - Compression based framework

Tentative syllabus to be adjusted to accommodate students needs

Topics

- Reinforcement Learning (if time permits)
  - Reinforcement Learning introduction
  - Evaluative Feedback
  - Dynamic Programming
  - Monte Carlo Methods
  - Temporal-Difference Learning
  - Deep Q learning (tentative)

Prerequisite Knowledge

- Basic machine learning concepts
  - Supervised/unsupervised/reinforcement learning
  - Classification, Regression, Cross validation, Overfitting, Generalization
  - Deep neural networks
  - See math4ml
- Basic calculus and linear algebra
  - Compute (by hand) gradients of multivariate functions
  - Conceptualize dot products and matrix multiplications as projections
  - Solve multivariate equations using, etc, matrix inversion, etc.
  - Understand basic matrix factorization
  - See linear algebra review, and advanced

Prerequisite Knowledge

- Basic optimization
  - Use techniques of Lagrange multipliers for constrained optimization problems
  - Understand and be able to use convexity
  - See convex analysis review, optimization review
- Basic probability and statistics
  - Understand: random variables, expectations and variance
  - Use chain rule, marginalization rule and Bayes' rule
  - Make use of conditional independence, and understand "explaining away"
  - Compute maximum likelihood solutions for Bernoulli and Gaussian distributions
  - See probability review

Course Format and Grading

Course Format

- Lecture Participation
  - Read pre-course material before class
  - **Submit a sheet of feedback** about last session at the beginning of each session
- Lecture Scribing
  - **Sign Up** [Here] **ASAP**: 28 sessions in total, <= 25 students, each student scribes at least 1 session
  - Bonus Scribing

Course Format

- Homeworks
  - Weekly homework: scan to upload to canvas
  - Due every **Thursday 15:00**
- Course Project
  - **Sign Up Here ASAP**: a list of projects, work with my grad. students
  - Final presentation and report
- Discussions on Piazza:
  - **Sign Up Here ASAP**: Be active, ask questions during sessions **in class, after class on Piazza**

Grading

- Lecture scribing (30%+10%)
- Homeworks (30%)
- Participation(10%)
- Course project(30%)

Lecture Scribing (30%) Logistics

- Within two days of class, you must post a detailed scribing of the class on Piazza
- You are also responsible for answering questions posted there (if you don't know the answer, talk to the presenter, TA or me)
- **Template** here. The basic commands are included in this template. I strongly prefer LaTex, talk to Jingling <jingling@cs.umd.edu> if you are not familiar with LaTex. (Lecture 1 scribing.)
- Bonus credit
  - One more scribing (10%)

Homeworks (30%)

- Complete Independently
- Typing(preferred) or clean handwriting (scan and submit pdf online)
- Due **every Thursday 3:00 pm**
  - **No Late submissions:**

Participation (10%) Logistics

- Hand in course feedback on the last session at the beginning of each session
- Complete course evaluation in a timely manner
- A maximum of 4 graceful sessions are allowed for unexpected emergencies

Course Project (30%) Logistics

- Your choice of topic
  - I will provide some ideas here (description will be added)
  - You are more than welcome to come up with new ideas
- Be done groups of **2** students
  - Write-ups must include the division of labor
  - Same grade for all team members unless there's a large discrepancy in labor
- Three items to hand in:
  - Proposal (**25 October**), includes *meeting with me (5%)*
  - Progress Report (**15 November**) (5%)
  - Final Presentation (**Last few lectures**) (10%)
  - Final Report (**6 Dec**) (10%)
- Goal: convince me that you learned something and can put that knowledge to use!

Tentative List of Course Projects

- **Tensornet**
- **Domain Adaptation**
- **Adversarial Example**
- **Differentially Private Topic Model**
- **Temporal Latent Variable Model**
- **Tensor ALS**
- **Generalization of Neural Networks**
- **Unsupervised Boosting**
- **Off-policy evaluation**

University Rules

Class Rules

- Nothing may be handed in late without:
  - prior arrangements (at least one week in advance), or
  - a doctor's note (or equivalent), modulo university regulations
- You are encouraged to ask questions in class or on Piazza
- Add/drop deadlines
  - University policy: http://registrar.umd.edu/current/registration/Schedule%20Adjustment.html
  - If you're going to drop, please please please do it now!
- Cheating: see academic integrity policy
- ADA/DSS:
  - Letter of accommodation to me within the first two week of class
  - Please also see ADA/DSS policy

Policy on class behavior

The open exchange of ideas, the freedom of thought and expression, and respectful scientific debate are central to the aims and goals of a this course. These require a community and an environment that recognizes the inherent worth of every person and group, that fosters dignity, understanding, and mutual respect, and that embraces diversity. **Harassment and hostile behavior are unwelcome in any part of this course.** This includes: speech or behavior that intimidates, creates discomfort, or interferes with a person's participation or opportunity for participation in the course.

We aim for this course to be an environment where harassment in any form does not happen, including but not limited to: harassment based on race, gender, religion, age, color, national origin, ancestry, disability, sexual orientation, or gender identity. Harassment includes degrading verbal comments, deliberate intimidation, stalking, harassing photography or recording, inappropriate physical contact, and unwelcome sexual attention. Please contact an instructor or CS staff member if you have questions or if you feel you are the victim of harassment (or otherwise witness harassment of others).

Edited from the NAACL Policy: http://naacl.org/policies/anti-harassment.html

Academic Integrity Policy

Any assignment or exam that is handed in must be your own work (unless otherwise stated). However, talking with one another to understand the material better is strongly encouraged. Recognizing the distinction between cheating and cooperation is very important. If you copy someone else's solution, you are cheating. If you let someone else copy your solution, you are cheating (this includes *posting solutions online in a public place)*. If someone dictates a solution to you, you are cheating.

Everything you hand in must be in your own words, and based on your own understanding of the solution. If someone helps you understand the problem during a high-level discussion, you are not cheating. We strongly encourage students to help one another understand the material presented in class, in the book, and general issues relevant to the assignments. When taking an exam, you must work independently. Any collaboration during an exam will be considered cheating. Any student who is caught cheating will be given an F in the course and referred to the University Office of Student Conduct. Please don't take that chance - if you're having trouble understanding the material, please let us know and we will be more than happy to help.

ADA/DSS Policy

Any student eligible for and requesting reasonable academic accommodations due to a disability is requested to provide, to the instructor in office hours, a letter of accommodation from the Office of Disability Support Services (DSS) within the first two weeks of the semester. You may reach them at 301-314-7682 or by visiting Susquehanna Hall on the 4th Floor.

The CS department does not consider requests for retroactive accommodation to be reasonable. In the same vein, we do not consider it reasonable to ask an instructor to create an alternate assignment of substance. The spirit of our accommodation is to help DSS-advised students find creative ways to meet the high standards we set for all our students.

Other Logistics

Other Logistics

- Instructor Office hours
  - Thursday 4:45 pm - 5:45pm AVW 3251
  - Or schedule an appointment

- TA: Jingling Li <jingling@cs.umd.edu>
  - Office hours: TBD

Other Logistics

- Announcements will be posted to Piazza
  - You are expected to read *regularly*
  - I will not necessarily repeat announcements in class

- Before asking a logistical question, check syllabus

Your task before next class

- Check out course webpage, Canvas, and Piazza
- Do the readings
- Register for Scribing, think about course projects
- Let me know dates of religious holiday you observe this semester
- Let me know if you will need DSS arrangements

Student Introduction

Ice breaking

Background + Research Interests + What do you expect from this course

Lecture Starts

Algorithmic Issues in Machine Learning

Algorithm design paradigm: give an algorithms that succeeds on all possible inputs

Difficulty: almost all of the optimization problems in modern ML are *computationally intractable*

Practitioners: use a wide variety of heuristics that are successful in practice

When/Why do these approaches work?

Q1: Which models in ML lead to tractable algorithmic problems?

Q1: Which models in ML lead to tractable algorithmic problems?

- Worst-case analysis
- Optimization problems that ML systems solve are hard in the worst-case
- Hard instances of ML problems are not the ones we would solve in practice anyways!
- Choose the right model that will lead us to discover new algorithms with provable guarantees.

Q2: Can new models be the inspiration for developing fundamentally new algos for ML problems?

Can we understand when and why widely used heuristics work?

White Board