Machine Learning: Data to Models 601.476/676

(Subtitle: Probabilistic Graphical Models)

Instructor: Daniel Malinsky (malinsky@jhu.edu)
Assistants: Noam Finkelstein (noam.finkelstein@jhu.edu)
Younsu Kim (ykim99@jhu.edu)
Razieh Nabi (rnabiab1@jhu.edu)

DESCRIPTION:

This is a second course in machine learning, focusing on probabilistic graphical models (PGMs). Most contemporary machine learning methods are probabilistic/statistical, i.e., they use the mathematical machinery of probability & statistics to represent uncertainty. In complex systems with hundreds or thousands of variables, the formalism of graphical models can make representation more compact, inference more tractable, and intelligent data-driven decision-making more feasible. We will focus on representational schemes based on directed and undirected graphical models and discuss probabilistic inference for prediction as well as structure learning. Applications of PGM-based methods include healthcare, genetics, economics, natural language processing, robotics, image analysis, neuroscience, and more. We will draw connections in lecture between theory and these application areas. The final project will be entirely "hands on," where students will apply techniques discussed in class to a real data set, and write up the results.

OBJECTIVES:

At the end of this course you will be able to

- Make intelligent choices about the model class / representation appropriate to a given data problem
- Understand the semantics and limitations of different model choices
- Perform inference and learning tasks for multiple model classes
- Analyze real data using probabilistic graphical methods

TEXTBOOK: *Probabilistic Graphical Models* by Daphne Koller and Nir Friedman (2009)
(Note: you are not required to purchase any textbook. Relevant sections will be posted online.)

SUPPLEMENTS: *Graphical Models* by Steffen L. Lauritzen (1996)
and *Machine Learning: A Probabilistic Perspective* by Kevin Murphy (2012)

GRADING OVERVIEW:

Problem sets: 45% (3x15%)
Project proposal: 5%
Final research project: 50%
(No midterm or final exams.)

FINAL PROJECT:

This will be a research project that requires students to apply methods learned in the class to real data. Several public (and relatively "clean") data sets will be made available, spanning multiple areas: biology, neuroscience, social science, and intelligent systems.* Students will write a report in the style of a NeurIPS/ICML paper, about 4-7 pages. The project proposal, worth 5% of the grade, is a brief write-up which will be due several weeks before the final deadline, describing the data set, the methods, and software involved. It will be graded on 0 or 100 scale: if the student hands in a (reasonable) proposal, they will be award a full 100% on that assignment. If the student fails to hand in a proposal, a 0% on that assignment, worth 5% of their grade. The point of the proposal is to incentivize planning, and to identify and potential problems or pitfalls ahead of time. Details regarding the expectations for this project will be made available in class.

*Graduate students may instead elect to use a data set relevant to their own research, pending approval by the instructor. Undergraduate students are required to use one of the data sets which will be made available.

PROGRAMMING:

Homework assignments will require some programming, and the final project will require real data analysis which may either require novel programming or the use of available software. You may use whatever programming languages you like, though the official languages of this class are R and Python. That means the instructor and TAs will only answer questions (to the best of their ability) about programming-related problems in R and Python — for any other language you are on your own.

COURSE SCHEDULE:

Meetings: Tuesdays and Thursdays, 12:00-1:15pm in Gilman 132

First day of class: Tues Jan 29th
Last day of class: Thurs May 2nd
Spring Break (no class): March 18-24
Note: April 16 & 18 class is cancelled (conference travel)

Week 1 (1/29, 1/31):
Introduction: overview of applications of graphical models, conditional independence

Week 2 (2/5, 2/7):
Bayesian networks / directed graphical models

Week 3 (2/12, 2/14):
Markov random fields / undirected graphical models

Week 4 (2/19, 2/21):
Exact inference: sum-product variable elimination, belief propagation on junction trees

Week 5 (2/26, 2/28):
Approximate inference (part 1): variational methods, mean-field approx, loopy BP

Week 6 (3/5, 3/7):
Approximate inference (part 2): sampling methods, Markov Chain Monte Carlo, Gibbs sampling

Week 7 (3/12, 3/14):
Structure learning: constraint-based & score-based learning of Bayesian networks and MRFs

SPRING BREAK

Week 8 (3/26, 3/28):
Structure learning: learning in settings with latent variables, semi-parametric assumptions

Week 9 (4/2, 4/4):
Parameter learning: maximum likelihood estimation, Bayesian estimation, confidence/credal intervals, EM algorithm

Week 10 (4/9, 4/11):
Real messy data problems: missing data (multiple imputation and EM), selecting tuning parameters, a look at some applications

Week 11 (4/23, 4/25):
Latent variable models: PCA, factor analysis, learning via tetrad/rank constraints

Week 12 (4/30, 5/2):
Temporal models: dynamic Bayesian networks, the Kalman filter, nonstationarity, Poisson and Hawkes processes, local independence for counting processes

CLASS POLICIES:

The strength of the university depends on academic and personal integrity. In this course, you must be honest and truthful, abiding by the *Computer Science Academic Integrity Policy*:

> Cheating is wrong. Cheating hurts our community by undermining academic integrity, creating mistrust, and fostering unfair competition. The university will punish cheaters with failure on an assignment, failure in a course, permanent transcript notation, suspension, and/or expulsion. Offenses may be reported to medical, law or other professional or graduate schools when a cheater applies.

Violations can include cheating on exams, plagiarism, reuse of assignments without permission, improper use of the Internet and electronic devices, unauthorized collaboration, alteration of graded assignments, forgery and falsification, lying, facilitating academic dishonesty, and unfair competition. Ignorance of these rules is not an excuse.

Academic honesty is required in all work you submit to be graded. Except where the instructor specifies group work, you must solve all homework and programming assignments without the help of others. For example, you must not look at anyone else's solutions (including program code) to your homework problems. However, you may discuss assignment specifications (not solutions) with others to be sure you understand what is required by the assignment.

If your instructor permits using fragments of source code from outside sources, such as your textbook or on-line resources, you must properly cite the source. Not citing it constitutes plagiarism. Similarly, your group projects must list everyone who participated.

Falsifying program output or results is prohibited.

Your instructor is free to override parts of this policy for particular assignments. To protect yourself: (1) Ask the instructor if you are not sure what is permissible. (2) Seek help from the instructor, TA or CAs, as you are always encouraged to do, rather than from other students. (3) Cite any questionable sources of help you may have received.

On every exam, you will sign the following pledge: "I agree to complete this exam without unauthorized assistance from any person, materials or device. [Signed and dated]".

Your course instructors will let you know where to find copies of old exams, if they are available.

Please report any violations you witness to the instructor.

You can find more information about university misconduct policies on the web at these urls:

- *Undergraduates: https://studentaffairs.jhu.edu/policies-guidelines/undergrad-ethics/*
- *Graduate students: http://e-catalog.jhu.edu/grad-students/graduate-specific-policies/*

ON GROUP/INDIVIDUAL WORK:

All assignments in this course are individual, not group, assignments. You may freely discuss homework assignments with your fellow classmates. The final solutions, however, must be written entirely on your own. This includes programming: you must implement any programming task on your own. Copying someone else's code (and then subsequently making minor changes) constitutes plagiarism. So, if you need to discuss programming assignments, you may discuss general strategy but should write the code by yourself.

STUDENTS WITH DISABILITIES:

Any student with a disability who may need accommodations in this class should obtain an accommodation letter from Student Disability Services: 385 Garland, (410) 516–4720, studentdisabilityservices@jhu.edu.