Course ID: EECS 491

Course Title: Probabilistic Graphical Models

Credit Hours: 3

Class Times: MW 12:45 - 2:00 (Spring 2019)

Classroom: Olin 313

Instructor: Mike Lewicki, **msl88@case.edu**, Olin 508, hours: MW 2-3 (for the first few weeks I will be holding office hours in Olin 314) instead of my office.

Teaching Assistant: TBA

Prerequisites: EECS 233, EECS 391, MATH 408, MATH 207
The topics in this course assume you are familiar with basic concepts in Artificial Intelligence and Machine Learning (EECS 391 and EECS 440). The assignments will have a programming component that involves implement and using algorithms, in Matlab, Python or Julia. I also recommend that you have a basic course in algorithms and data structures (e.g. EECS 233). The mathematical basis of this course is probability theory, so I also recommend an introductory course on probability theory (MATH 380). Both probability theory and topics in this course draw heavily on univariate and multivariate calculus (e.g. MATH 121,122 and 223) and linear algebra (e.g. MATH 201 or 207). Having all these courses would make you very well prepared for this course. It is certainly possible to do well without having all the recommended background, but be prepared to spend more time in areas where your background is less solid. If your background is incomplete or missing, I would recommend against taking this course. Both books devote several chapters to foundational background material, so that is a good place to gauge your understanding.

Bibliography
Our main textbook is:

Bayesian Reasoning and Machine Learning by Barber

A pdf ebook for this is available online: http://www.cs.ucl.ac.uk/staff/d.barber/brml/.
I also recommend and will draw material from:

Machine Learning: A Probabilistic Perspective by Murphy

Pattern Recognition and Machine Learning by Bishop

The schedule of topics refers primarily to Barber and Murphy, because it's highly instructive to have explanations from multiple points of view. The course organization is more similar to Barber.

Course Website
The course has a standard canvas site (https://canvas.case.edu). If you are registered, you should have access. Check there periodically for announcements, assignments, lecture slides, etc.

Course Description

Probabilistic graphical models are widely used in Machine Learning and Artificial Intelligence (AI). It focuses on the fundamental theories, algorithms, techniques required to design adaptive, intelligent systems and devices that make optimal use of available information and time. Practical applications are covered throughout the course. Topics covered will include Bayesian inference and probabilistic reasoning in both discrete and continuous problem spaces, Bayesian belief networks, graphical models, algorithms for inference and learning, clustering, mixture models, principal and independent component analysis, sparse representation, non-linear dimensionality reduction, probabilistic linear models, discrete and continuous state hidden Markov models.

Course Outline (subject to revision)[1]

	Date	Topics	Readings	Notes
1	Mon, Jan 14	**Introduction and Overview** - course topics overview, applications, examples of probabilistic modeling and inference, class organization	B. chapter introductions; M.1	
		Probabilistic Reasoning (Barber Ch. 1)		
2	Wed, Jan 16	**Probabilistic Reasoning** - basic probability review, reasoning with Bayes' rule	B.1.1-2; M. 2.1-2	A1 out
	Mon, Jan 21	*(No class - Martin Luther King Jr. Holiday)*		
3	Wed, Jan 23	**Reasoning with Continuous Variables** - probability distribution functions, prior, likelihood, and posterior, model-based inference	B.1.3, B.8, B. 9.1; M.2.1-4	
4	Mon, Jan 28	**More Probabilistic Reasoning** - more complex examples	B.1.1-2; M. 2.1-2	
		Belief Networks & Graphical Models (Barber Ch. 3&4)		
5	Wed, Jan 30	**Belief Networks 1** - representing probabilistic relations with directed acyclic graphs, simple belief networks, modeling dependencies	B.3.1-2, (graph background B. 2); M.10.1-2	A1 due; A2 out
6	Mon, Feb 4	**Belief Networks 2** - inference, uncertainty and unreliability in evidence, independence, causality	B.3.1-4; M. 10.5	
7	Wed, Feb 6	**Graphical Models 1** - Markov random fields, expressiveness of graphical models, factor graphs	B.4; M.19.1-4	
8	Mon, Feb 11	**Graphical Models 2** - undirected graphical models, independence relationships, energy functions, Monte Carlo inference in MRFs	B.4; M10, 23, 24	
		Inference Techniques (Barber Ch. 5)		
9	Wed, Feb 13	**Introduction to Inference** - variable-elimination, Monte-Carlo methods, junction trees, loops	B.5 & 6; M.20, Bishop Ch.8	A2 due; A3 out
10	Mon, Feb 18	**Efficient Inference 1** - message passing, sum-product algorithm on factor graphs	B.5.1-2; M.20; Bishop Ch.8	

[1] In readings, B.x.y refers to Barber chapter x, section y, M.x.y refers to Murphy.

	Date	Topics	Readings	Notes
11	Wed, Feb 20	**Efficient Inference 2** - derivation of sum-product algorithm, expectation propagation, message passing for continuous distributions	B.5.2-4; M.20; Bishop Ch.8	
		Learning in Probabilistic Models (Barber Ch. 8-11)		
12	Mon, Feb 25	**Learning in Probabilistic Models** - represented data, statistics for learning, common probability distributions, learning distributions, maximum likelihood, Gaussian models	B.8.1-3,6-7; M.3.1	A4 out
13	Wed, Feb 27	**Representation and Probabilistic Models** - probabilistic models as belief networks, continuous parameters, training belief networks.	B.9.1-3	A3 due
14	Mon, Mar 4	**More on Generative Models** - Bayesian concept learning, Dirichlet multinomial model, bag of words model, Naïve Bayes, Bayesian model selection	drawn from B. 9, 10, 12; M. 3.1-4	
15	Wed, Mar 6	**Learning with Hidden Variables** - hierarchical models, missing data, expectation maximization (EM), EM.for belief nets, variational Bayes, gradient methods, deep belief nets	B. 9.4, 11.1-2; M.10.4	
	Mar 12-16	*Spring break - no class*		
		Clustering, PCA, and Dimensionality Reduction (Barber Ch. 12, 15, 20)		
16	Mon, Mar 18	**Gaussian Mixture Models** - clustering, nearest neighbor classification, k-means, latent variable models, mixture models, EM.for MMs	B.20; M.11	A5 out
17	Wed, Mar 20	**Bayesian Model Selection** - Occam's razor, Bayesian complexity penalization, Laplace approximation, Bayes information criterion, Bayes factors	B.12.1-5, M. 5.3, M.11.5	A4 due
18	Mon, Mar 25	**Principal Component Analysis** - dimensionality reduction, optimal linear reconstruction, whitening, latent semantic analysis	B.15.1-3; M. 12.2	
19	Wed, Mar 27	**Non-Linear Dimensionality Reduction** - ISOMAP, LLE, deep belief networks	handouts	
		Linear Models (Barber Ch. 17&18)		
20	Mon, Apr 1	**Optimal Linear Reconstruction** - derivation, matrix equation techniques, matrix calculus, Lagrangians for matrix constraints	B.15.1-3, 17	A6 out
21	Wed, Apr 3	**Bayesian Interpolation** - non-linear generative models for noisy data, regression, curve fitting, regularization	B.18	A5 due
21	Wed, Apr 3	**Sparse Linear Models** - sparse representation and coding, independent component analysis (ICA), compressed sensing	B.21.6, M.13	

	Date	Topics	Readings	Notes
22	Mon, Apr 8	**Sparse Linear Models** (continued) - blind source separation (BSS), ICA learning algorithm		
		Dynamic Models (Barber Ch. 23&24)		
23	Wed, Apr 10	**Dynamical Models** - discrete state Markov models, transition matrix, language modeling, MLE, mixture of Markov models, applications: Google PageRank algorithm, gene clustering	B.23.1, M. 17-1-2	A6 due; A7 out
24	Mon, Apr 15	**Hidden Markov Models (HMMs)** - classical inference problems, forward algorithm, forward-backwards algorithm, Viterbi algorithm, sampling, natural language models	B.23.2, M. 17.3-4	
25	Wed, Apr 17	**Learning HMSs** - EM.for HMMs (Baum-Welch algorithm), GMM.emission model, discriminative training, related models and generalizations, dynamic Bayes nets, applications: object tracking, speech recognition, bioinformatics, part-of-speech tagging	B.23.3-5, M. 17.5-6	
26	Mon, Apr 22	**Continuous-state Markov Models 1** - linear dynamical systems (LDS), stationary distributions, autoregressive models, latent linear dynamical systems, Kalman filtering, robotic SLAM	B.24.1-3, M. 18.1-3	
27	Wed, Apr 24	**Continuous-state Markov Models 2** - inference in CSMMs, filtering, smoothing, trajectory analysis, learning LDS, EM.for LDS, approximate online inference	B.24.4-7, M. 18.4-5	A7 due
28	Mon, Apr 29	**Retrospective**		
		No final exam. Grad student projects due Mon, Apr 29.		

Grading Policy

Students are required to attend lectures, read the assigned material in the textbooks prior to class, and expected master all the material covered in class. Classes missed due to reasons other than medical conditions cannot be made up.

Assignments consist of seven combined theory and programming assignments and one final project (for graduate students). There are no exams. Assignments are worth 78% of the final grade. The lowest score among the six assignments will be dropped, so each counting assignment is worth 13%. The final project is worth 22% of the final grade.

Note: Undergraduate students are not required to complete a final project. In this case, each counting project is worth 100%/6 = 16 2/3% of the final grade.

Academic Integrity Policy

Collaboration is encouraged, but students must turn in their own assignments. Any problem completed collaboratively should contain a statement that the students have contributed equally toward the completion of the assignment. Students are expect to understand all the material presented in the assignment.

I reuse some of the problems from previous years, because they are good problems and have been refined and improved over many years. *Referring to previous years' assignments is cheating*. It takes a lot of time and effort to develop good homework assignments, and we want you and future students to be able to continue to use them. We also welcome feedback for improvement. It is your responsibility to help protect the educational value of these assignments. Violations in any of the areas above will be handled in accordance with the University Policy on Cheating and Plagiarism.

Notes on Assignments

Assignments are the primary means by which to learn the mathematical material presented in class and will be coordinated with the lectures. Some assignments will depend on material completed in earlier assignments. Therefore, you are strongly advised to complete each assignment to stay current.

Some of the advanced methods discussed in class are not practical to cover in a homework because of their complexity. If you would like to study a particular topic in greater depth, it would be well worth considering it as a class project.

Class assignments are to be completed using either 1) jupyter notebook using python, julia, or R or 2) code (e.g. matlab) with latex and pdf writeup. All assignments and code must be submitted via Canvas and must include the notebook file, the .pdf file of either the notebook or the writeup. Do not submit any binary or large data files. Submitted files must be in pdf format and contain all graphics.

Notes on Class Projects

The class project is a project of your design. I will send out a request for project proposals in later in the semester and may request to meet with you. You are required to submit project proposal and turn in a project report.

Collaborations for joint projects are acceptable, but each person must make a unique contribution to the project, and each person must write up a report and give a presentation that describes their contribution to the project. Collaborations must be approved in advance and have a clear plan for the role of each student.

Your project writeup should clearly explain the ideas and results. To do this you will need to make complex ideas easy to follow and understand. The organization should have a clear, logical flow. An effective way to see if you can communicate effectively is to give your report to someone and see how many questions they ask. If they need to ask a lot, it means they are not understanding what you trying to say.

The best source I know of for how to write clearly is the book "Style: The Basics of Clarity and Grace" by Williams and Colomb. Here is the amazon link. Another good source of effective communication is Jean-luc Doumont. Here is a youtube talk of his.

The projects will be graded on the following factors:

1. Explanation and motivation (50%)

   What problem are you solving? What issue or topic are you investigating? Why is it interesting? What does success mean? How do we know if you are successful?

2. Approach and rationale (30%)

   What approach are you taking? Why does your this make sense? Why did you choose this approach. How did you have to simplify the problem or limit the scope of the project?

3. Results and/or analysis (20%)

   What are the take home messages from your project? Did your approach "work" ? Why or why not? How could it be improved? What did you learn from the project? What other approaches might you consider if did further work on it.

Note that a good grade does not depend on having a successful project with interesting results. It depends on you begging able to provide a clear explanation of what you did, why it was a good idea, and what the results were.

For the factors above, I have purposefully given more weight to points 1 and 2, because it's far more important to have a clear conception of the problem and well-reasoned rationale to your approach than it is to get results for uninteresting or poorly conceived problems.