Electrical and Computer Engineering Course Syllabus

Instructor:	Prof. Henry Pfister	**E-mail:**	`henry.pfister@duke.edu`
Office / Hour:	305 Gross Hall / T 1:30 - 2:30 PM	**Phone:**	(919) 660-5288
Class Room:	318 Gross Hall	**Class Time:**	MW 3:05 - 4:20 PM

Course Name:	**ECEN 590.10**
Course Title:	**Graphical Models and Inference**
Prerequisite(s):	graduate level course in applied probability
Required Text(s):	Course reader handouts
Other Text(s):	*Modern Coding Theory* by Richardson and Urbanke (MCT)
	Information, Physics, and Computation by Mezard and Montanari (IPC)
	Information Theory, Inference and Learning Algorithms by MacKay (ITILA)

Course Objectives:

In this course, graphical models will be introduced with an emphasis on general principles that can be applied to a wide variety of problems. The described techniques will be applied to a range of problems in communications, computer science, and statistics. In particular, their application to error-correcting codes and compressed sensing will be covered in some detail.

Many real-world systems can be modeled by a large number of dependent random variables. For such systems, inference problems often require computing the marginal distribution of a small subset of the random variables. Graphical models provide a unifying framework for these systems and provide efficient algorithms for both exact and approximate inference. Moreover, the best schemes currently known for error-correction and compressed sensing both rely on graphical models.

Belief propagation is a well-known algorithm for efficient inference on graphical models that is exact when the graphical model is a tree. Sparse graphical models are particularly well suited for approximate inference via belief propagation. For example, low-density parity-check codes provide state-of-the-art error correction and are typically decoded by belief propagation on a sparse graphical model. Also, K-SAT problems give rise to sparse graphical models when K is fixed, the number of variables is large, and the number of clauses is proportional to the number of variables. Some dense graphical models also allow efficient inference and analysis. In particular, the AMP algorithm provides efficient reconstruction in compressed sensing for Gaussian measurement matrices. In statistics, the AMP algorithm also enables efficient and robust M-estimation for some high-dimensional statistical models.

At the end of the course, the student should be able to:

1. Describe basic properties of directed and undirected graphical models.
2. Define maximum-likelihood (ML), maximum-a-posteriori (MAP), and a-posteriori-probability (APP) estimation for inference problems.
3. Construct factor-graph representations for standard inference problems (e.g., the decoding of parity-check codes)
4. Implement message-passing algorithms for ML, MAP, and APP estimation and understand why they are optimal on tree factor graphs.
5. Define the marginal polytope and the standard linear programming (LP) relaxation of ML estimation for factor graphs (e.g., solving Sudoku via LP).
6. Understand how and why the LP relaxation fails (e.g., define pseudocodewords for LDPC codes)
7. Describe the connection between message-passing algorithms and the Bethe free entropy on general factor graphs.

8. Understand the statistics of random instances of graphical models (e.g., random coding and random code ensembles).

9. Analyze the performance of message-passing decoding on the binary erasure channel (BEC) for random ensembles and individual codes.

10. Explain why locally-optimal algorithms (e.g., message passing) are not globally optimal (e.g., the gap between MAP decoding and message-passing for LDPC codes on the BEC).

11. Describe the factor graph associated with the K-SAT problem and explain why constraint satisfaction problems can be more challenging.

12. Implement belief-propagation guided decimation to solve random instance of the K-SAT problem.

13. Construct the factor graph associated with the compressed sensing (CS) problem.

14. Derive the AMP algorithm as an asymptotically exact approximation of belief-propagation for CS.

15. Implement the AMP algorithm for random instances of some CS problems.

16. Identify situations in engineering, computer science, and statistics where where factor graphs can be used to obtain useful results.

Student Evaluation:

Midterm	25%	Homework	25%
Final	25%	Project	25%

- Homework will include programming assignments.

Rules and Guidelines:

The class shall follow all established policies of Duke University.

Course Topics and Hours: (2.5 hours/week)

Unit	Topics	Hours
1	Introduction to Graphical Models	2.5
2	Factor Graphs and Inference	2.5
3	Belief Propagation (BP) and Message-Passing Algorithms for Inference	2.5
4	Low-Density Parity-Check Code Construction and Simulation	2.5
5	Inference via Linear Programming (LP) and Pseudocodewords	2.5
6	The Bethe Free Entropy and Belief Propagation	2.5
7	Analysis of Message-Passing Algorithms via Density Evolution	2.5
8	Locally Optimal Algorithms and Global Optimality	2.5
9	Factor Graphs, Polar Codes, and Channel Polarization	2.5
10	Factor Graphs and the K-SAT Problem	2.5
11	Belief-Propagation Guided Decimation for Constraint Satisfaction Problems	2.5
12	Factor Graphs and the Compressed Sensing Problem	2.5
13	Derivation of the AMP Algorithm	2.5
14	Connections to High-Dimensional Statistics	2.5
	Total Hours	35