Course Syllabus, Fall 2014
4 graduate hours. 4 undergraduate hours.

ECE 544 is a special topics course: lectures and discussions related to advanced topics and new areas of interest in speech, image, and multidimensional processing. ECE544NA is the section of this course dedicated to special topics in pattern recognition. Content varies every year, but usually includes error metrics (e.g., information-theoretic and perceptron-based) and optimization (e.g., neural network, Bayesian, stochastic, and convex programming techniques) for the supervised, semi-supervised, and unsupervised estimation of probability densities, feature selection, regression and classification.

In fall 2014, the course will focus on problems relevant to big data, including parametric learning (MAP-REDUCE), non-parametric learning (GMM and MLP), methods for accurately and rapidly training neural networks, methods for estimating how much data you need (PAC learning, cross-validation), and methods for acquiring those data (semi-supervised learning, active learning, crowdsourcing, and Markov decision processes).

Pre-requisites: Vector spaces and probability. For example, it is sufficient to have taken (ECE 313 and ECE 310 or equivalent) or (STAT 542 or equivalent) or (CS 446 or equivalent).
Text, fall 2014: Pattern Recognition, Duda, Hart and Stork
The text will be supplemented occasionally with articles from the professional literature, e.g., covering
the error exponent, covering Boltzmann pre-training, and covering some of the Bayesian techniques.
Problem sets will not be drawn from the text, so students can use other texts if desired, but notation in
lecture will be drawn primarily from the Bishop text.

<table>
<thead>
<tr>
<th>Lecture Topics</th>
<th>Contact hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification/Regression and Training: ML, MAP, MPE, and MMSE</td>
<td>3</td>
</tr>
<tr>
<td>Parametric Models, Sufficient Statistics, and MAP-REDUCE</td>
<td>3</td>
</tr>
<tr>
<td>Kernel Estimators, GMM, and Sigmoidal Neural Nets</td>
<td>3</td>
</tr>
<tr>
<td>MMSE and Cross-entropy training criteria</td>
<td>3</td>
</tr>
<tr>
<td>Stochastic gradient descent, L-BFGS</td>
<td>3</td>
</tr>
<tr>
<td>Restricted Boltzmann machines</td>
<td>3</td>
</tr>
<tr>
<td>Sample complexity: Probably approximately correct</td>
<td>3</td>
</tr>
<tr>
<td>Semi-supervised learning</td>
<td>3</td>
</tr>
<tr>
<td>Active learning</td>
<td>3</td>
</tr>
<tr>
<td>Crowdsourcing</td>
<td>3</td>
</tr>
<tr>
<td>Markov Decision Process: Baum-Welch, Q-learning</td>
<td>3</td>
</tr>
<tr>
<td>Special Topics</td>
<td>10</td>
</tr>
<tr>
<td>Total Contact Hours</td>
<td>43</td>
</tr>
</tbody>
</table>

Homework and Exams
Written homework assignments are performed until mastery, as follows.
1. Each written homework assignment includes three concepts.
2. Each concept is exemplified by at least three written problems, available on the
course web page.
3. Each student needs to choose just one problem from each concept, thus each
student should hand in answers to three problems per week.
4. If a student receives less than a perfect score on any problem, she can try to raise
her grade by working one of the other problems associated with the same concept.
If the second problem is scored more highly than the first, it replaces the grade of
the first.
5. In-class written exams will be problems generated by the same algorithm that
generates homework problems, and will be drawn from a subset of the written
homework concepts.

Matlab Homework
1. Each week during weeks 1-10 of the semester, students will be assigned a matlab
programming assignment, usually based on simulated toy data.
2. Matlab assignments are due one week after assigned.
3. Graded Matlab assignments may be re-done, as often as you like, in order to fix
erors in the first assignment. Fixed assignments replace the original grade.
Final Project
Every week during the semester, the matlab assignment will compare two different solutions to the same problem (GMM vs. mixture multinomial, RBF vs. sigmoidal nodes, cross entropy vs. MMSE training, etc.). The goal of the final project is to select one of those contrasts, and:

1. Add a third option to the contrast (a third method for solving the same problem)
2. Prove that there is some quality criterion that is optimized by the proposed third option (consistency, computational complexity, sample complexity, etc.; if you want to prove that the algorithm optimizes one of these criteria under certain assumptions, then make a text argument in support of your assumptions).
3. Apply the three algorithms to a dataset of interest to you. The new dataset should be either:
   a. Real-world data, drawn from a task of research or commercial interest, or
   b. Toy data that you generate yourself, according to a generative model that emphasizes the difference among the three tested algorithms
4. Measure performance of the algorithms in at least three ways (e.g., computational complexity, sample complexity, asymptotic accuracy, recall vs. precision, cross-entropy vs. MMSE), and discuss interpretation of the results in light of the provable properties of the algorithms.

Grading Policy:
Written Homework 25%
Matlab Homework 25%
In-class Quizzes 25%
Final Project 25%

Prepared By: Mark Hasegawa-Johnson