ECE 544: Pattern Classification

Instructor: Mark Hasegawa-Johnson

Syllabus, Fall 2007

Who, When, Where

Tuesdays and Thursdays, 11:30 AM - 1:00 PM, 106B3 Engineering Hall
Instructor: Mark Hasegawa-Johnson, jhasegaw@uiuc.edu.
Office hours: Wednesday 3:00-4:00, Beckman 2011.
Course Web Page: http://www.isle.uiuc.edu/544/index.html.

Course Summary

Pattern Recognition is concerned with recognition of an unknown given object as belonging to one of a number of classes. Classification is performed by discovering class specific "patterns" among a range of measurable object features and utilizing these class characteristic features for recognition of unknown objects. The design of a pattern recognition system requires development of four major modules: sensing, feature extraction, decision making, and system performance evaluation. This course will introduce the fundamentals of statistical pattern recognition with examples from several application areas. Techniques for handling multidimensional data of various types and scales along with classification/recognition algorithms will be explained. The course will present competing approaches to exploratory data analysis and classifier design so students can make judicious choices when confronted with real pattern recognition problems.

Pre-Requisites

Math 415 or equivalent (orthogonality, determinants, and eigenvectors), ECE 413 or equivalent (conditional, marginal, and joint PDFs; Gaussian PDF; cumulative density function, PDF, and expectations of functions of random variables), and programming experience.

Text, Problem Sets, Tests, Project, and Grading

The textbook is: Richard Duda, Peter Hart, and David Stork, *Pattern Classification* second edition, John Wiley and Sons, 2001. Relevant articles will occasionally be distributed in order to cover material inadequately explained in the text. Typically, one problem set question will be assigned on the subject of each article.

There will be about nine problem sets: one for each substantive chapter in the textbook (chapters 2-10). Each one will include typically four long and difficult written problems (mostly from the text), one literature review problem, one prescribed programming problem, and one open-ended programming problem.

Each literature review should review one journal article, one long conference paper ($\geq$ 8 pages), or two short conference papers (<8 pages). The literature review must include the following elements: (a) what was the application? (b) what methods have previously been used to address that application? (c) what specific problem with previous methods are the authors trying to solve? (d) what method did they propose? (e) how did they evaluate their proposal? (f) is the improvement statistically significant (i.e., not chance)? is the improvement worth the added complexity or cost?

There will be two tests, each covering half of the material in the course. One will take place during the 7th week, one during the 14th week of the course. Tests will cover the material that's on the problem sets.

Grades will be based on the homework (50%) and tests (50%).

Data

The open-ended part of each homework assignment will ask you to use a data set of your own choosing.

Data are plentiful on the web, on any subject from stock prices to star luminosities, but they are often not well organized. Good, well cleaned databases include the UC Irvine machine learning databases, the NetFlix movie preference database, the TIMIT phoneme classification database, and the CMU computer vision database; you may have other data of your own choosing.

Lecture Topics

Lectures will generally follow the order of chapters in the text. Here is a rough chronology.

1. Introduction and Definitions. Classification vs. detection vs. imaging. Supervised vs. unsupervised vs. semi-supervised. Training, testing, and development data. Training vs. testing algorithms.

2. Bayesian Decision Theory. Minimum-expected-risk classification; Discriminant functions; Error probabilities; Error bounds; Mutual information and conditional entropy; Bayesian networks.

3. Parameter Estimation. Maximum likelihood, maximum *a posteriori*, and minimum mean-squared error parameter estimation; Bayesian methods (no hard decisions about parameters); informative, uninformative, and conjugate priors. Optimization methods: linear and convex optimization, gradient descent. Sufficient statistics, Expectation-Maximization (EM); overfitting. PCA and LDA.

4. Non-Parametric Techniques. Histograms and Parzen windows; K-nearest neighbors; metrics.

5. Linear Discriminant Functions. Decision surfaces; minimum-squared-error training; support vector machines (SVM).

6. Multilayer Neural Networks. Universal approximation functions; optimality criteria; closed-form and iterative learning algorithms; error backpropagation; conjugate gradient learning; RBF networks; the kernel trick for converting linear to nonlinear problems.

7. Stochastic Methods. Monte Carlo methods; simulated annealing.

8. Nonmetric Methods. Classification and regression trees (CART), recognition with strings, grammatical inference.

9. Algorithm-Independent Learning. Parameter bias and variance. Empirical risk and expected risk; regularization criteria; minimum description length. Development test; cross-validation. Classifier combination; AdaBoost; log-linear combination.

10. Unsupervised Learning. Density estimation: Parzen windows and Gaussian mixtures. Clustering: K-means clustering, hierarchical clustering. Manifold estimation: self-organizing maps, locally linear embedding and ISOMAP.