COEN 296 Topics in Computer Engineering

Introduction to Pattern Recognition and Data Mining

Instructor: Dr. Giovanni Seni
G.Seni@ieee.org

Department of Computer Engineering
Santa Clara University

Overview

- Course Goals & Syllabus
- Pattern Recognition Example
  - Features
  - Classification
  - Generalization
  - System components
- Related Fields: ML & DM
- Design Cycle
- Computational Complexity
- The R Language

Course Goals

- Convey excitement about an immensely useful field
  - Large increase in digital data (barcode scanners, e-commerce, etc.)
  - Moore's Law
- Provide foundation for further study/research
- Expose to real data
- Introduce you to toolbox of methods

Syllabus

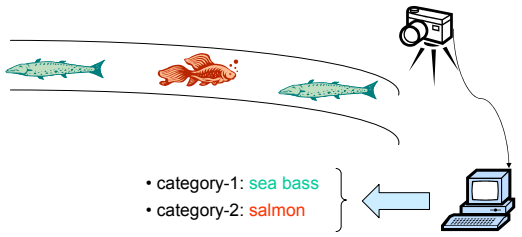
Jan 6	Introduction
Jan 13	Bayesian Decision Theory (2.1-2.6, 2.9)
Jan 20	Parameter Estimation (3.1-3.4; see also 4.5 HMS)
Jan 27	Linear Discriminant Functions (3.8.2, 5.1-5.8)
Feb 3	Neural Networks (6.1-6.5)
Feb 10	Neural Networks (6.6, 6.8)
Feb 17	Clustering (10.6, 10.7; see also 9.3-9.6 HMS)
Feb 24	Clustering (10.9)
Mar 2	Non-metric: Association Rules (5.3.2 HMS)
Mar 9	Text Retrieval (14.1-14.3 HMS)

Introduction
Pattern Recognition

- "The act of taking in raw data and taking an 'action' based on the 'category' of the pattern "
- Useful applications
  - Speech recognition
  - Word & Character Recognition
    - OCR (Optical Character Recognition)
  - Fingerprint identification ("biometrics")
  - DNA sequence identification ("bioinformatics")
  - Fraud detection
  - etc.

Introduction
Example

- Sorting incoming Fish on a conveyor according to species using optical sensing



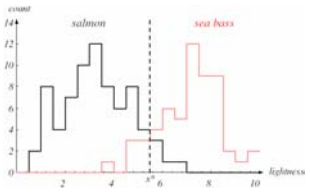
Introduction
Example

- Feature Extraction
  - *Representation* in which patterns that lead to same action are "close" to one another, yet "far" from those that demand a different action – i.e., discriminative
  - Data reduction
- Features to explore
  - Length, Lightness, Width, Number and shape of fins, Position of the mouth, etc…

Introduction
Example

- Initial *model*: sea bass is generally longer and lighter than salmon
  - Histograms on *training samples*

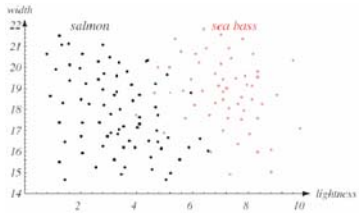
ID	Class	length	lightness
1		7.8	3.1
2		19.1	7.9
3		5.6	4.2
…			



Introduction
Example

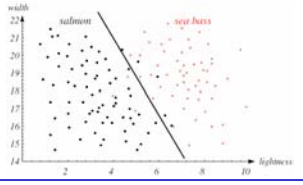
- Feature Space

$$\text{Fish} \Longrightarrow \mathbf{X} = \begin{pmatrix} x1 = \text{lightness} \\ x2 = \text{width} \end{pmatrix}$$



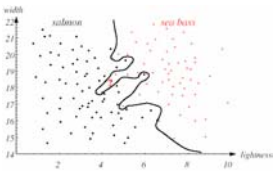
Introduction
Example

- Classification
  - Separate feature space into regions corresponding to the classes
  - The separating boundary is called the *decision boundary*
  - Perfect classification is often impossible… use probability framework
    - Easy to incorporate "priors" and misclassification "costs"

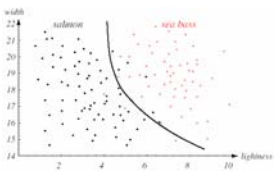


Introduction
Example

- Generalization
  - Ability to correctly classify novel input
  - Tradeoff between decision model complexity and generalization performance

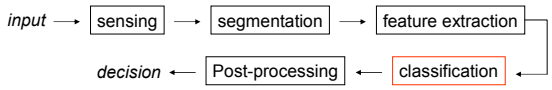


• complex
• lower training error
• higher test error



• simpler
• higher training error
• lower test error

Introduction
Pattern Recognition System



- *Sensing* – converts physical inputs into signal data
  - Bandwidth, resolution, sensitivity, distortion of transducer imposes limitations on system
- *Segmentation* - isolates objects from background or other objects
- *Post-processing* – account for "context" and cost of errors

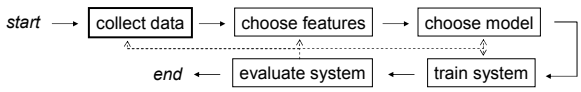
Introduction
Related Disciplines

- Data Mining – produce insight and understanding about the structure of *large* *observational* datasets – e.g.,
  - Find interesting relationships
  - Summarize the data in new ways that are understandable and actionable
- Machine Learning – how to construct computer programs that automatically improve with experience (Mitchell)
  - Theory and algorithms
- Other – Statistics, information theory, etc.

Introduction
Related Disciplines (2)

- Data Mining Algorithm Components
  - *Task*: visualization, classification, clustering, regression, rule discovery
  - *Structure*: functional form of the model we are fitting to the data (e.g., linear, hierarchical)
  - *Score function*: goodness-of-fit function we are using to judge the quality of our fitted model on observed data
  - *Search/optimization method*: computational procedure used to find the maximum (or minimum) of the score function for a particular model
  - *Data management technique*: location and manner in which data is accessed

Introduction
Design Cycle

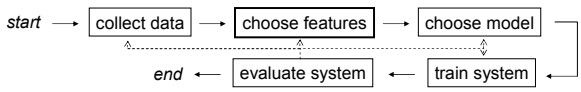


- Representative set of examples for training and testing the system
  - Can account for large part of the development cost
- Data matrix:

$n \times d$

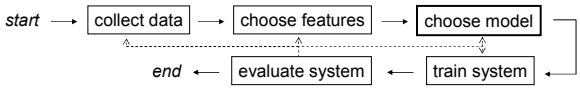
ID	Age	Sex	Marital Status	Education	Income
248	54	Male	Married	High school	100000
249	??	Female	Married	High school	12000
250	29	Male	Married	Some college	23000

Introduction
Design Cycle (2)



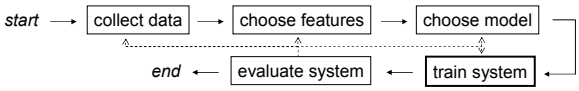
- *Feature choice* – useful for discriminating
  - Easy to extract
  - Invariant to irrelevant transformations
  - Insensitive to noise
- Type
  - Quantitative – measured on a numerical scale
  - Categorical: nominal and ordinal (possessing a natural order)

Introduction
Design Cycle (3)



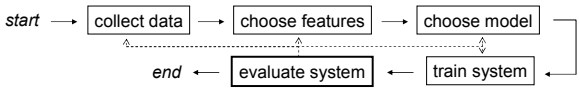
- *Predictive Modeling* – the value of one variable is predicted from the known values of other variables (classification, regression)
  - E.g., a *nonlinear* model $Y = aX^2 + bx + c$
- *Descriptive Modeling* – clustering and segmentation, dependency modeling, probability density estimation

Introduction
Design Cycle (4)

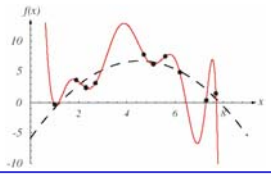


- *Training* – using training patterns to *learn* or estimate the parameters of the model (*supervised* or *unsupervised*)
  - Score Function: quantifies how well model fits a given data set
    - E.g., likelihood, sum of square errors, misclassification rate
  - Optimization (or Search) Method: determine the parameter values that achieve a minimum (or maximum) of the score function
    - E.g., gradient descent

Introduction
Design Cycle (5)

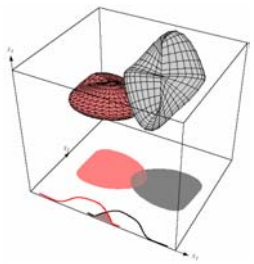


- *Evaluation* – measure performance and adjust components appropriately
- Train vs. Test Error
  - Overfitting
  - Bias-variance tradeoff



Introduction
Dimensionality

- Classification accuracy depends upon the dimensionality and the amount of training data
  - Theoretically, error rate can be reduced by introducing new, independent features
  - Need features that help separate the class pairs most frequently confused (e.g., distance between class means)



Introduction
Dimensionality (2)

- Practical *paradox*: beyond a certain point, the inclusion of additional features leads to worse performance
- Source of difficulty
  - Wrong model
    - E.g., Gaussian assumption
    - Independence assumption
  - Inadequate number of training samples
    - Distributions are not estimated accurately

Introduction
Computational Complexity

- Time/space considerations are of considerable practical importance at each stage
  - A table lookup might result in error-free recognition but impractical
- Scalability – as a function of:
  - Number of features (d)
  - Number of patterns (n)
  - Cumber of classes (c)
- Learning vs. decision-making time

Introduction
The R Language

- An open source version of "S" – a language and environment for data analysis
  - http://www.r-project.org/
  - Library provides many datasets
- Sample commands:

```
> x <- read.table("mydata.txt", header = TRUE)
> dim(x)
[1] 8192   18
> x[5, 7:9]
   P S  K
5 11 4 12
> hist(x[,7], breaks=100, xlab="Amount", main="P")
```


Introduction
The R Language (2)

- Other useful functions:
  - Input/Output: *read.table*, *read.delim*, *scan*, *write*, *write.table*
  - Extraction: *which*, *apply*
  - Names: *row.names*, *colnames*, *names*
  - Plots: *hist*, *plot*, *points*, *lines*, *pdf*, *dev.off*
  - Error catching: *stop*, *warning*
  - Sizes: *dim*, *nrow*, *ncol*, *length*
  - Math: *sum*, *mean*, *cor*, *log*, *max*, *min*, *range*
  - Casts: *as.matrix*, *as.vector*, *as.numeric*
  - Type test: *is.matrix*, *is.vector*, *is.numeric*, *is.data.frame*
  - Ordering: *sort*, *order*
  - Help: *?command*