CS385: Machine Learning

Introduction

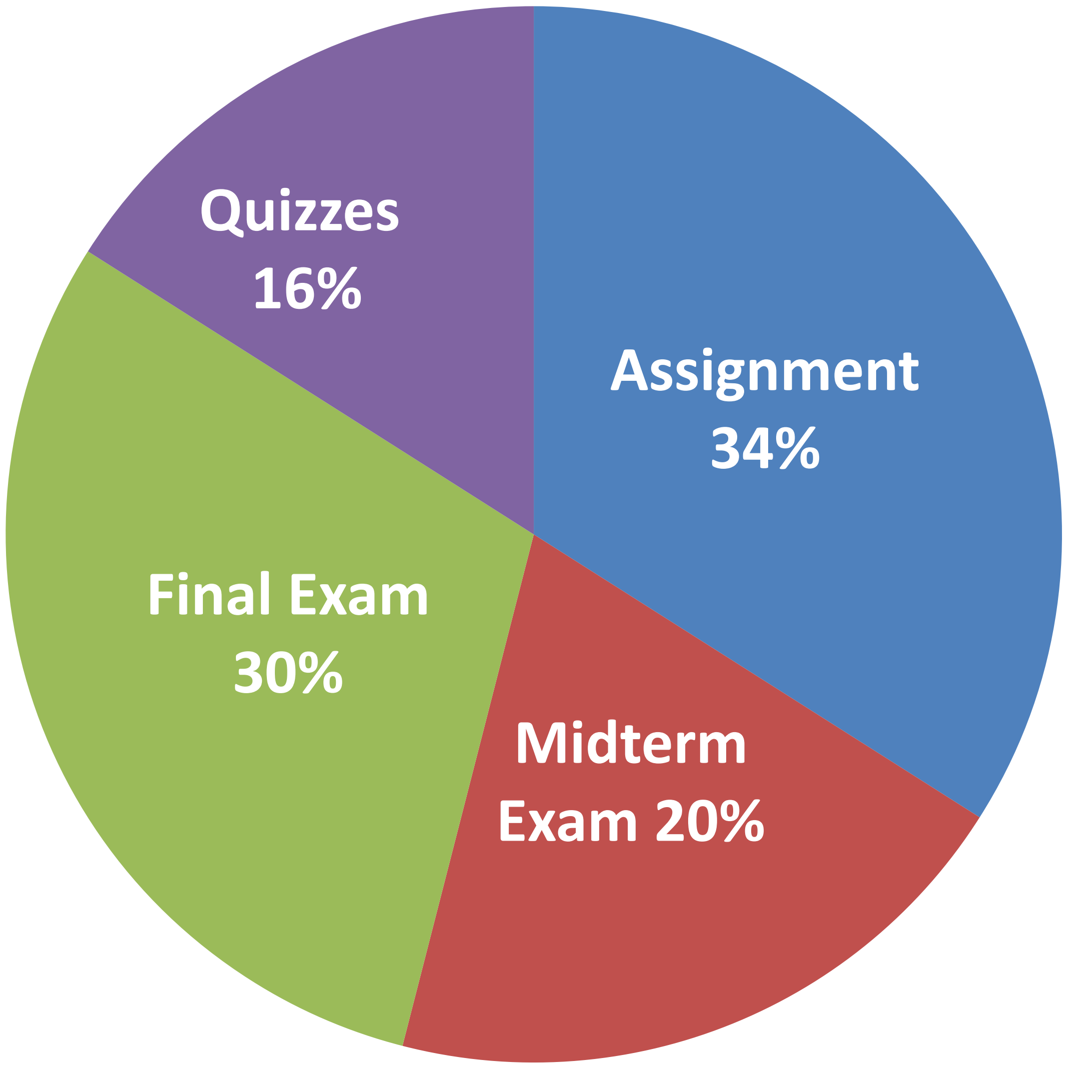
Course Logistics

- Class time: Tue & Thurs: 4 – 5.20PM @ Avery
- Instructors: Dr. Fang Liu & Dr. Prabhu Natarajan
- Email: fang.liu@digipen.edu
  prabhu.n@digipen.edu
- Include "[CS385]" in email subject
- Office Hours: Wednesday: 12:00 - 2:00 pm or by appointment

Textbooks

- None!
- Material and notes are to be provided when necessary.

Assessment



Assessment Distribution

- Quizzes: 16%
- Programming Assignments: 34%
- Midterm Exam: 20%
- Final Exam : 30%

- **You must receive an average score of 60% on both the midterm and final exams to pass this course, regardless of your quiz/assignment scores.**

Interaction

- **Moodle** : main interaction media
- All materials will be uploaded to Weekly Outline after the lectures
- https://distance.sg.digipen.edu

Assignments

- **Six assignments**:

  Assignment 1: Familiarization with Python

  Assignment 2: KNN Algorithm

  Assignment 3: Linear Regression

  Assignment 4: Logistic Regression

  Assignment 5: K-means Clustering

  Assignment 6: Artificial Neural Network
- **Warning: Please follow instructions exactly for assignment submissions, as is described in syllabus and assignment handout.**
- Assignment report is important – either latex or MS Word

Temporary Outline

Week No.	Topics
1	Introduction to Machine Learning, Fundamentals of Machine learning
2	K-Nearest Neighbor, Simple Linear Regression
3	Naïve Bayes and Text Processing
4	Multi-Variable Linear Regression, Gradient Descent Algorithm
5	Polynomial Linear Regression, Stochastic Gradient Descent Algorithm
6	Logistic Regression
7	Midterm Exam, Paper review
8	Clustering, Partitioning Around Medoids (PAM), [CLARANS]
9	[Hierarchical], K-Means, PCA
10	EM Algorithm, K-Means vs EM
11	Density-based clustering: DBSCAN, Supervised Learning Review
12	Linear Regression and Artificial Neural Network (A.N.N.).
13	Feed Forward Shallow A.N.N.
14	Recurrent Hopfield A.N.N.
15	Review & Final Exam

Machine Learning

- Humans – Learn from experiences
- Computers – Follow the instructions
- Can computers learn from experiences?
- Can computers make decision from experiences?
  - Yes, through machine learning.
  - Using the previous data

Machine Learning

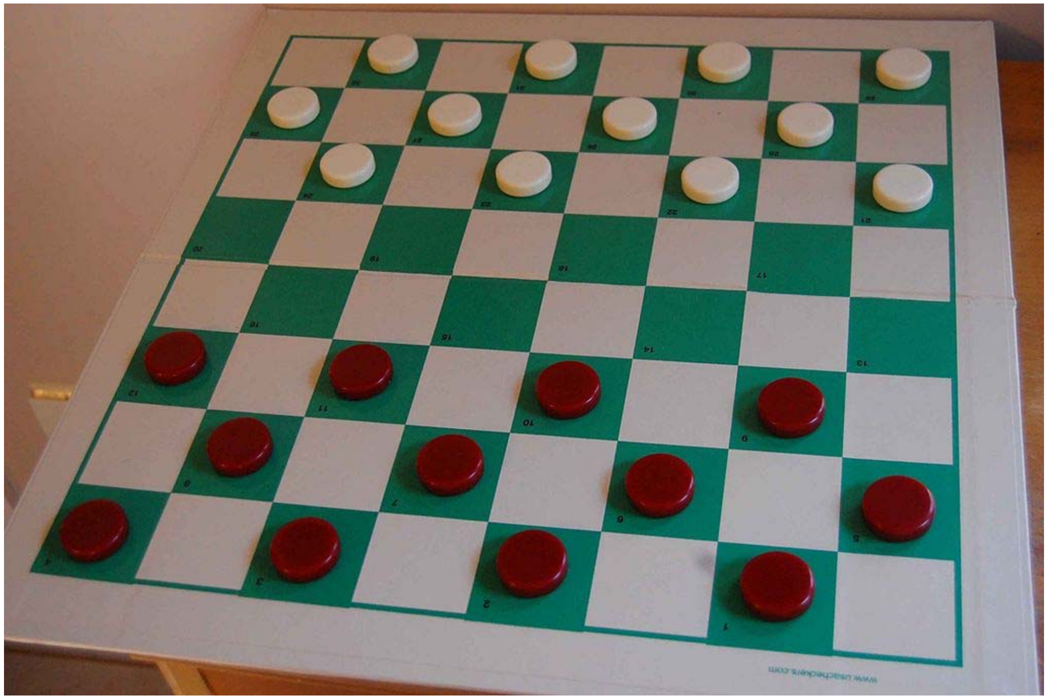
- Improve automatically with experience
- Imitating human learning
  - Human learning
  - Fast recognition and classification of complex classes of objects and concepts and fast adaptation
  - Example: neural networks
- Some techniques assume statistical source
  - Select a statistical model to model the source
- Other techniques are based on reasoning or inductive inference (e.g. Decision tree)

Disciplines relevant to Machine Learning

- Artificial intelligence
- Bayesian methods
- Control theory
- Information theory
- Computational complexity theory
- Philosophy
- Psychology and neurobiology
- Statistics

Machine Learning Definition

- Arthur Samuel (1959). Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.



Machine Learning Definition

- Arthur Samuel (1959). Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.

- Tom Mitchell (1998). Well-posed learning problem: A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

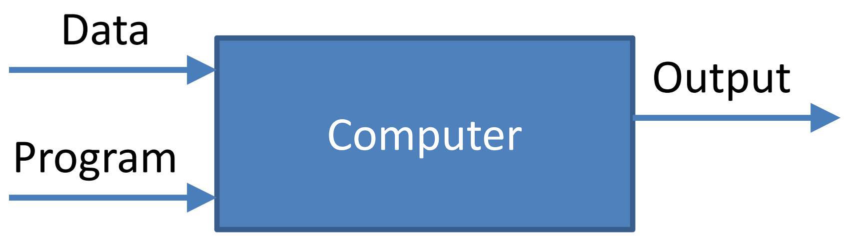
Examples of Learning Problems

- Example 1: Handwriting Recognition
  - T: Recognizing and classifying handwritten words within images
  - P: percentage of words correctly classified
  - E: a database of handwritten words with given classification.

- Example 2: Learn to play checkers:
  - T: play checkers
  - P: percentage of games won in a tournament
  - E: opportunity to play against itself

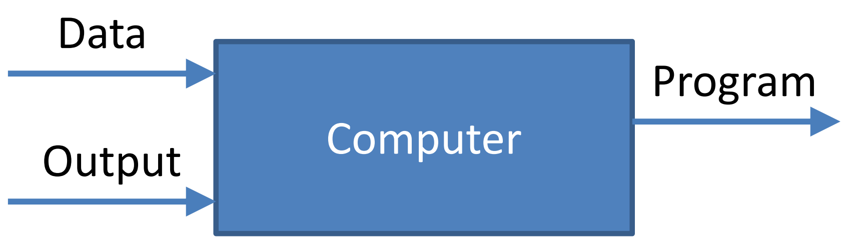
Traditional Programming vs Machine Learning Paradigm

Traditional Programming:



Finding a Square root of a number

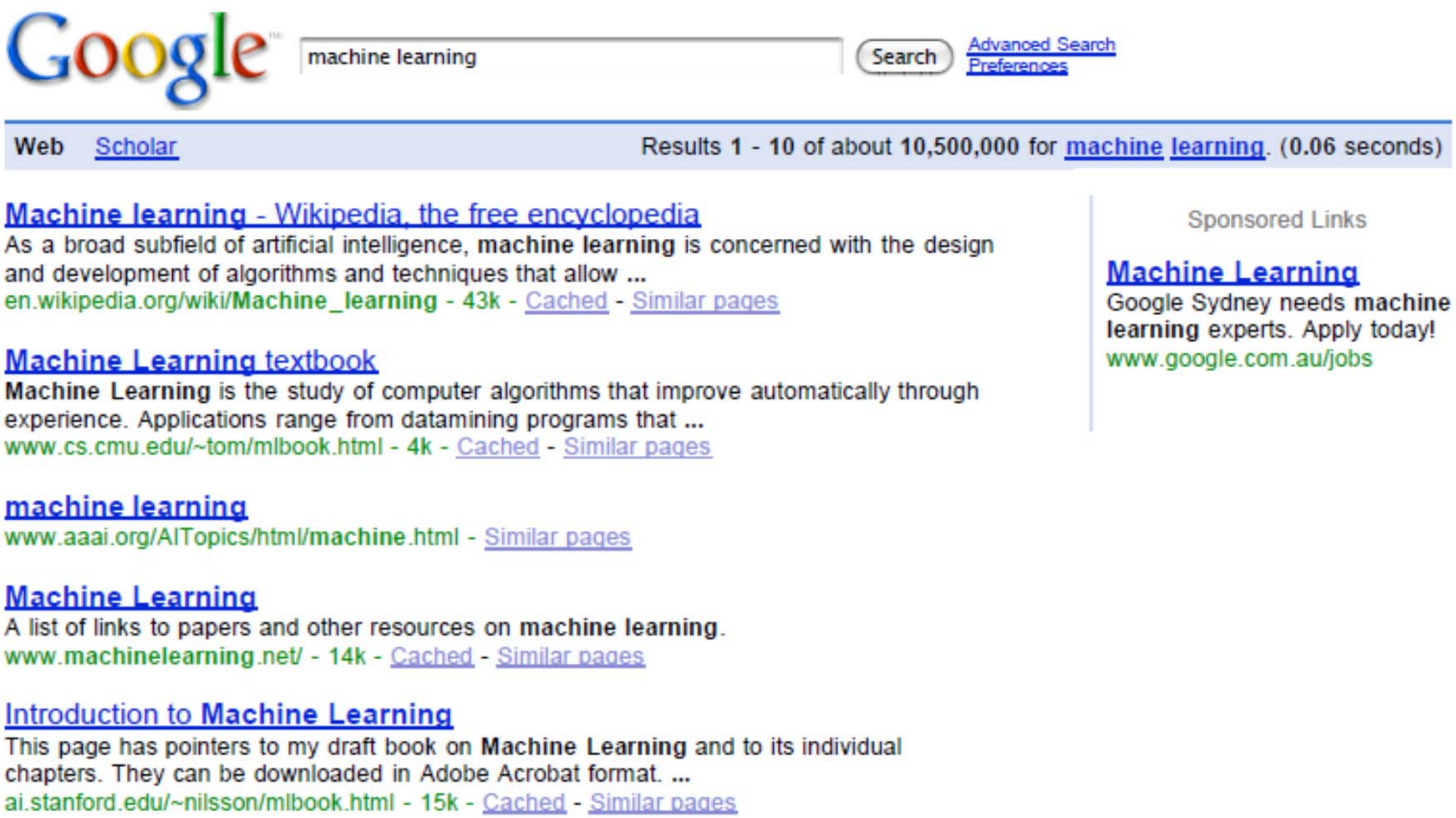
Machine Learning:



Predicting weather today

Machine Learning - Applications

Web Page Ranking



Recommendation Systems

Face Detection & Recognition

Named Entity Recognition

HAVANA (Reuters) - The European Union's top development aid official left Cuba on Sunday convinced that EU diplomatic sanctions against the communist island should be dropped after Fidel Castro's retirement, his main aide said.

<TYPE="ORGANIZATION">HAVANA</> (<TYPE="ORGANIZATION">Reuters</>) - The <TYPE="ORGANIZATION">European Union</>'s top development aid official left <TYPE="ORGANIZATION">Cuba</> on Sunday convinced that EU diplomatic sanctions against the communist <TYPE="LOCATION">island</> should be dropped after <TYPE="PERSON">Fidel Castro</>'s retirement, his main aide said.

Machine Learning
&
Pattern Recognition

What is Pattern Recognition?

- Pattern Recognition (PR) is the scientific discipline that concerns the description and classification (recognition) of patterns (objects).
- PR is an important component of intelligent systems.
- PR is the study of how machines can
  - observe the environment
  - distinguish patterns of interest from their background
  - make sound and reasonable decisions about the categories (classes) of the patterns

Definitions from the literature

- “The assignment of a physical object or event to one of several prespecified categories” *–Duda and Hart*
- “A problem of estimating **density functions** in a high-dimensional space and dividing the space into the regions of categories or classes” – *Fukunaga*
- “Given some examples of complex signals and the correct decisions for them, make decisions automatically for a stream of future examples” – *Ripley*
- “The science that concerns the description or classification (recognition) of measurements” *–Schalkoff*
- “The process of giving names ω to observations **x**”, *–Schürmann*
- Pattern Recognition is concerned with answering the question “***What is this?***” – *Morse*

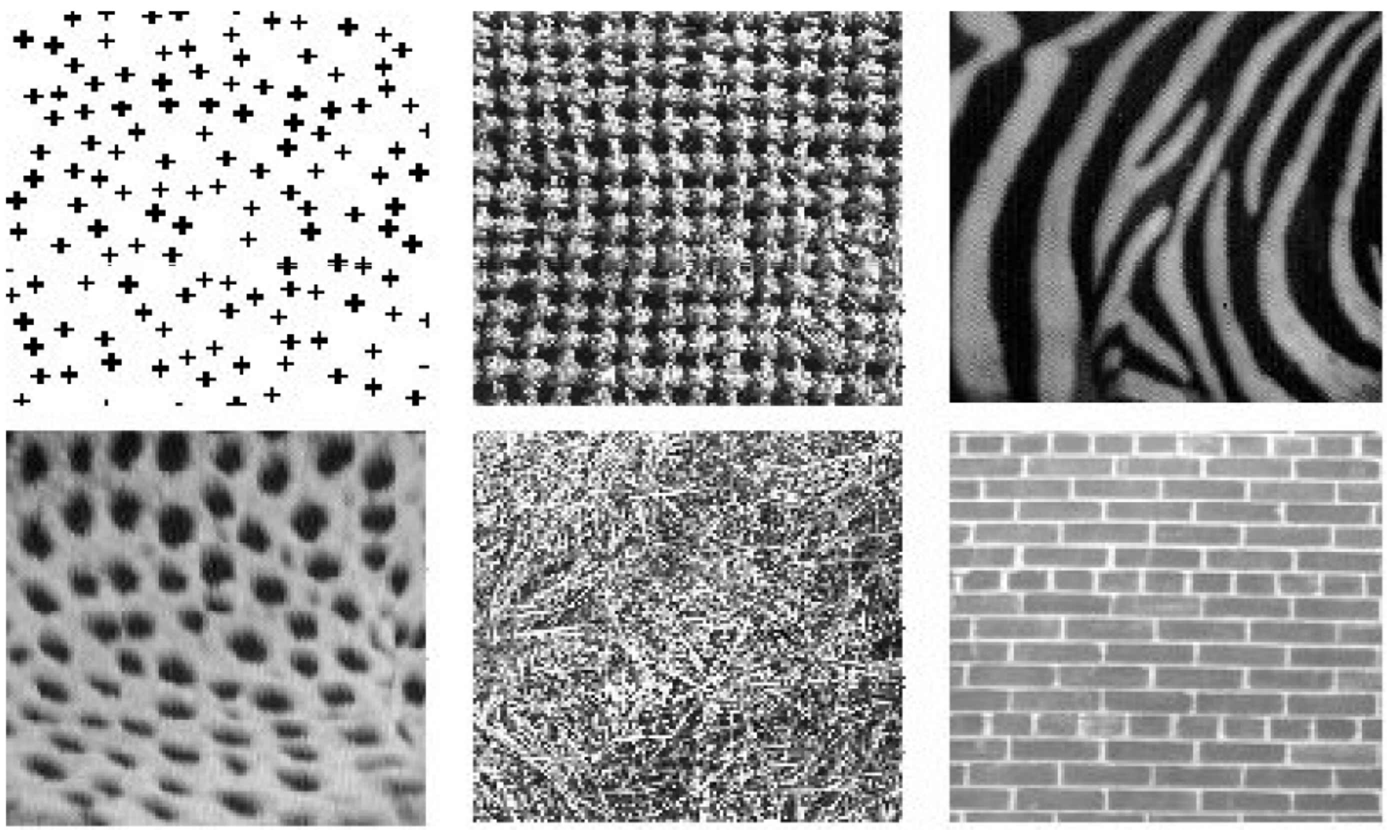
What is a pattern?

- Watanabe defines a pattern as ***"the opposite of chaos; it is an entity, vaguely defined, that could be given a name."*** (S. Watanabe, Pattern recognition: human and mechanical. Wiley, 1985)

- A set of instances that
  - share some regularities and similarities
  - is repeatable
  - is observable, sometimes partially, using sensors
  - may have noise and distortions

Example of Patterns

- Texture patterns: A wide variety of texture patterns are generated by various **stochastic processes**.

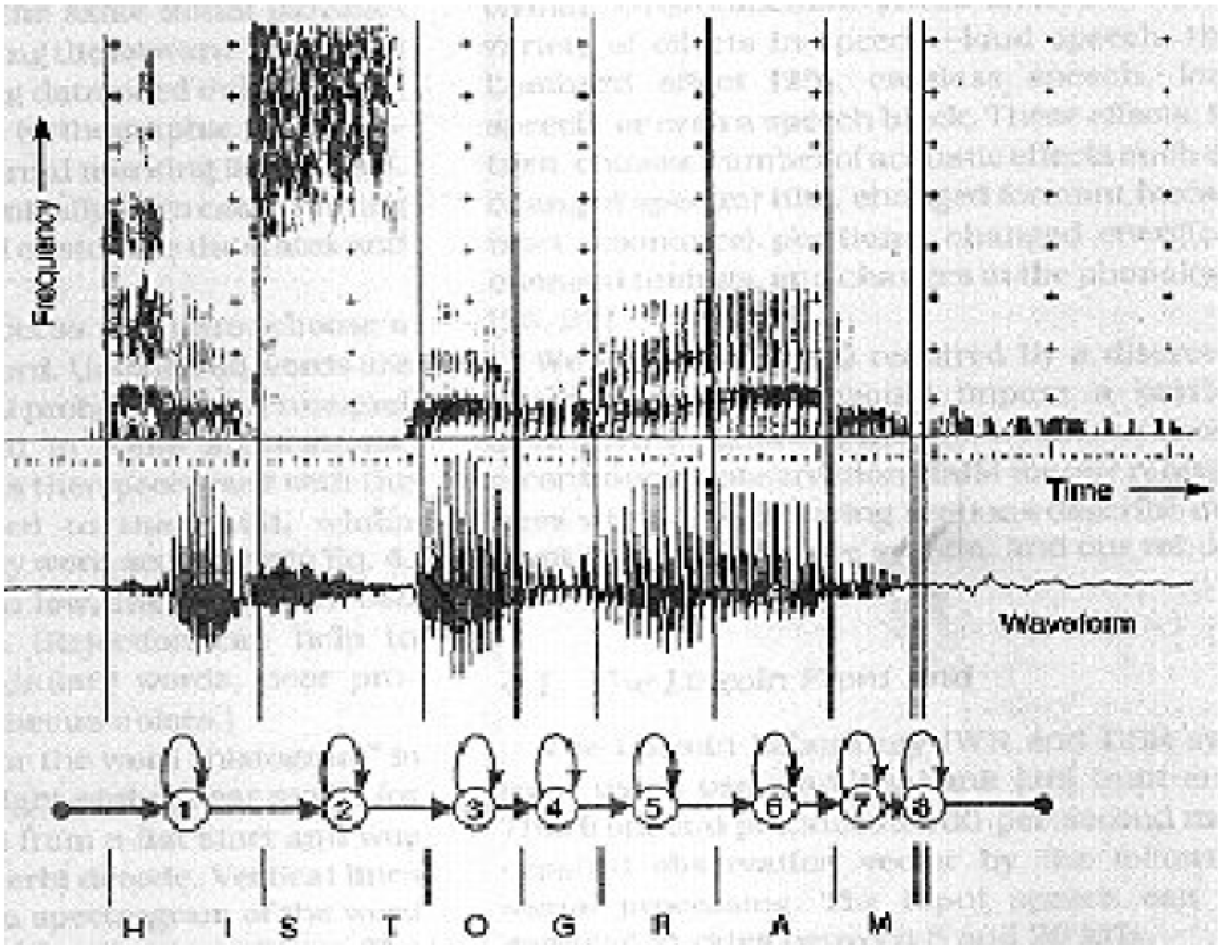


Example of Patterns

- Human Faces: Sample images from the facial expression database of CMU Advanced Multimedia Processing Lab.

Example of Patterns

- Speech Patterns: Speech signal and hidden Markov model



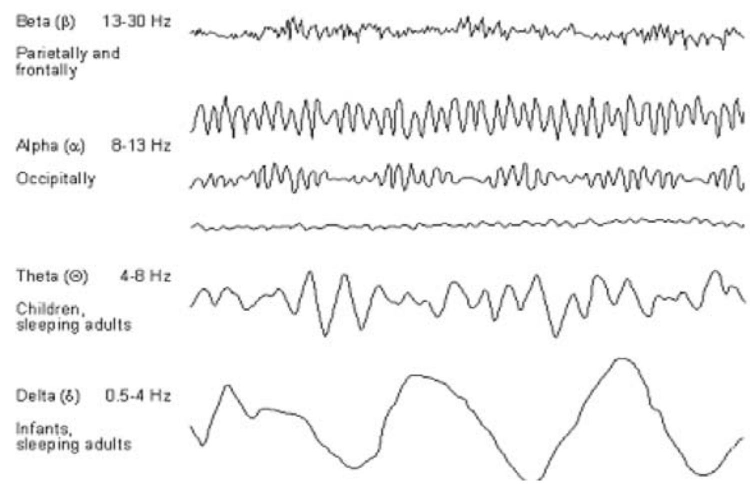
Example of Patterns

- Hand Written Digits: Samples from MNIST handwritten digit database – built from NIST (National Institute of Standards and Technology) Special Databases 1 and 3.

Example of Patterns



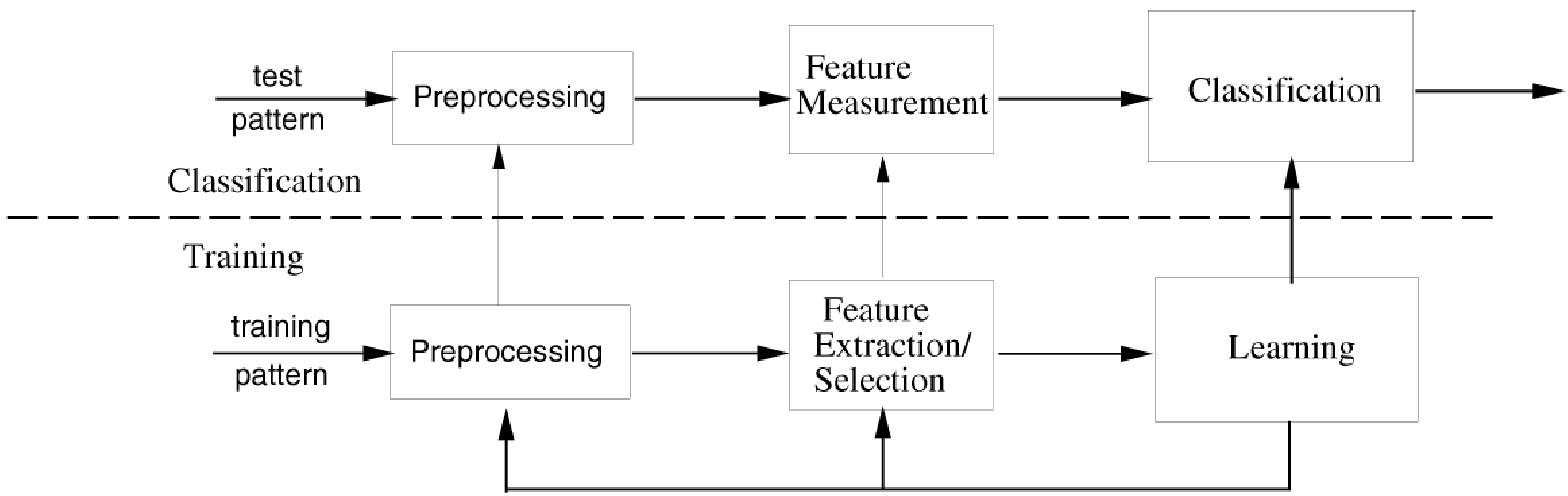
- Fingerprints, Iris images
- Handwritten characters
- Text patterns
- Patterns of brain activities (EEG, fMRI)
- Video category patterns
- Human behavioral patterns
- Biological signals
- Medical images (MRI, CT…)
- Many others …



Examples of Pattern Recognition Applications

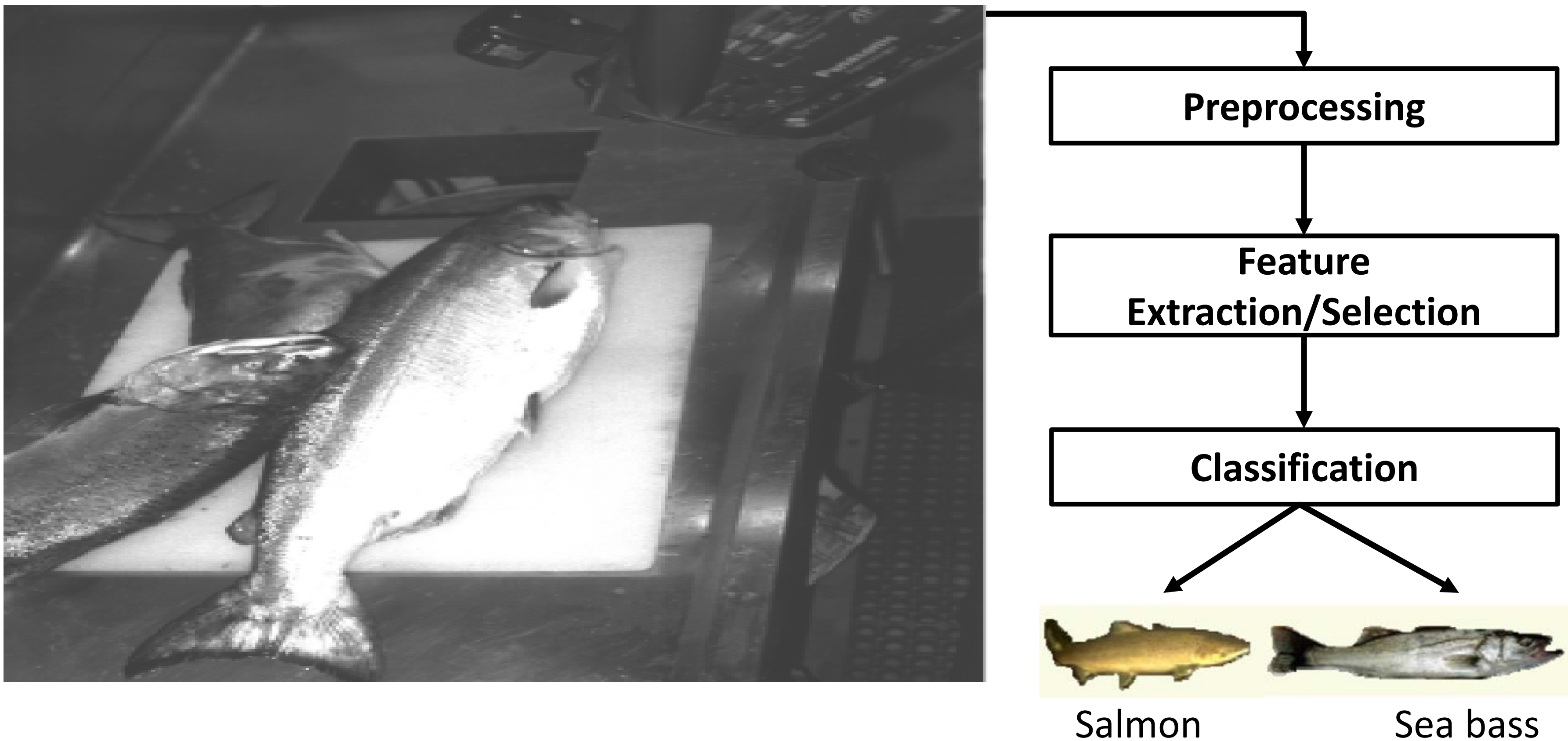
Problem Domain	Application	Input Pattern	Pattern Classes
Bioinformatics	Sequence analysis	DNA/Protein sequence	Known types of genes/patterns
Data mining	Searching for meaningful patterns	Points in multi-dimensional space	Compact and well-separated clusters
Document classification	Internet search	Text document	Semantic categories (business, sports, science, etc.)
Document image analysis	Reading machine for the blind	Document image	Alphanumeric characters, words
Multimedia database retrieval	Internet search	Video clip	Video genres (e.g. action, dialogue, romantic, etc.)
Speech recognition	Telephone directory enquiry without operator assistance	Speech waveform	Spoken words

Pattern Recognition Model



An Example

- Sort incoming fish on a conveyor into two classes: Salmon and Sea Bass



Pattern Recognition Process

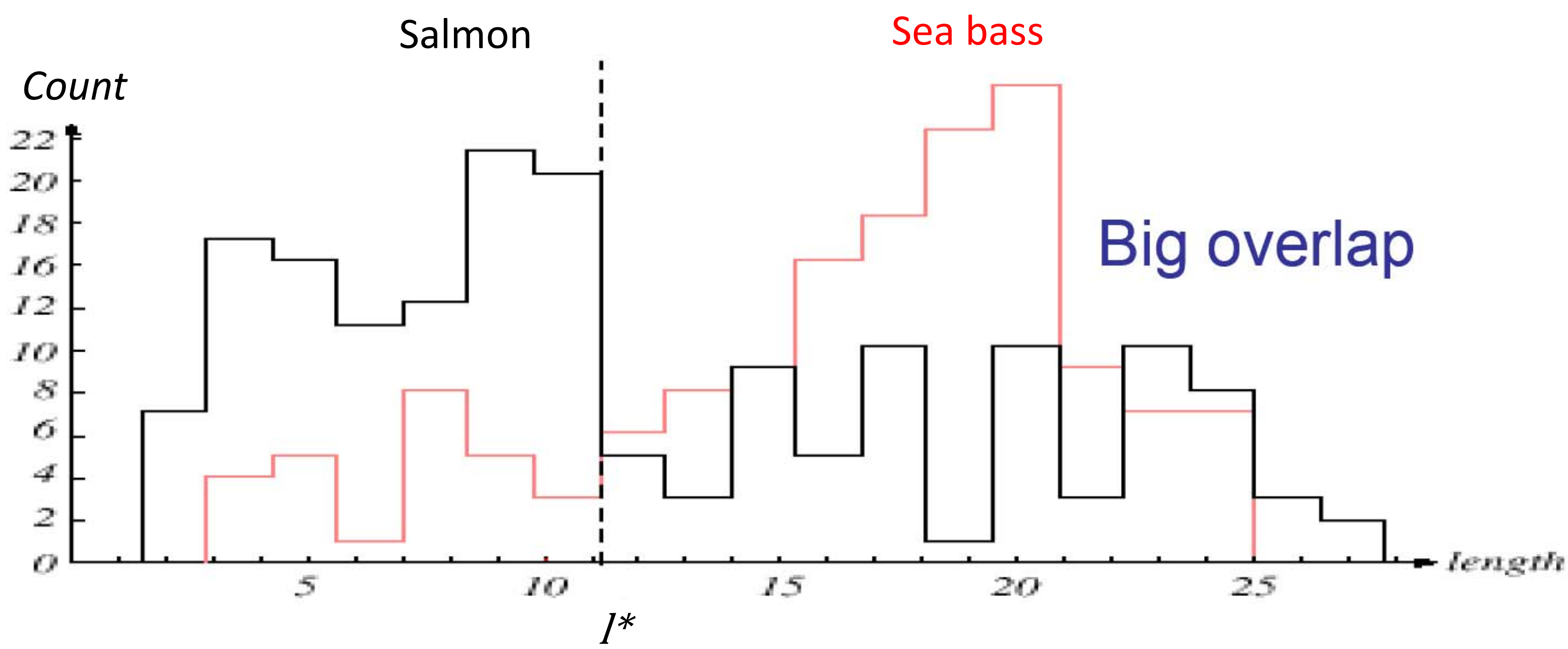
- **Sensing**
  - Capture an image of the fish using a camera
- **Preprocessing**
  - Image enhancement
  - Separate touching or occluding fishes
  - Extract a single fish, i.e., find the boundary
- **Feature extraction/selection**
  - Measure certain features of the fish to be classified
- **Classification**
  - Make a final decision: either salmon or sea bass

Possible Features

- Length
- Lightness (Average intensity of the scales)
- Width
- Number and shape of fins
- Position of the mouth, etc.

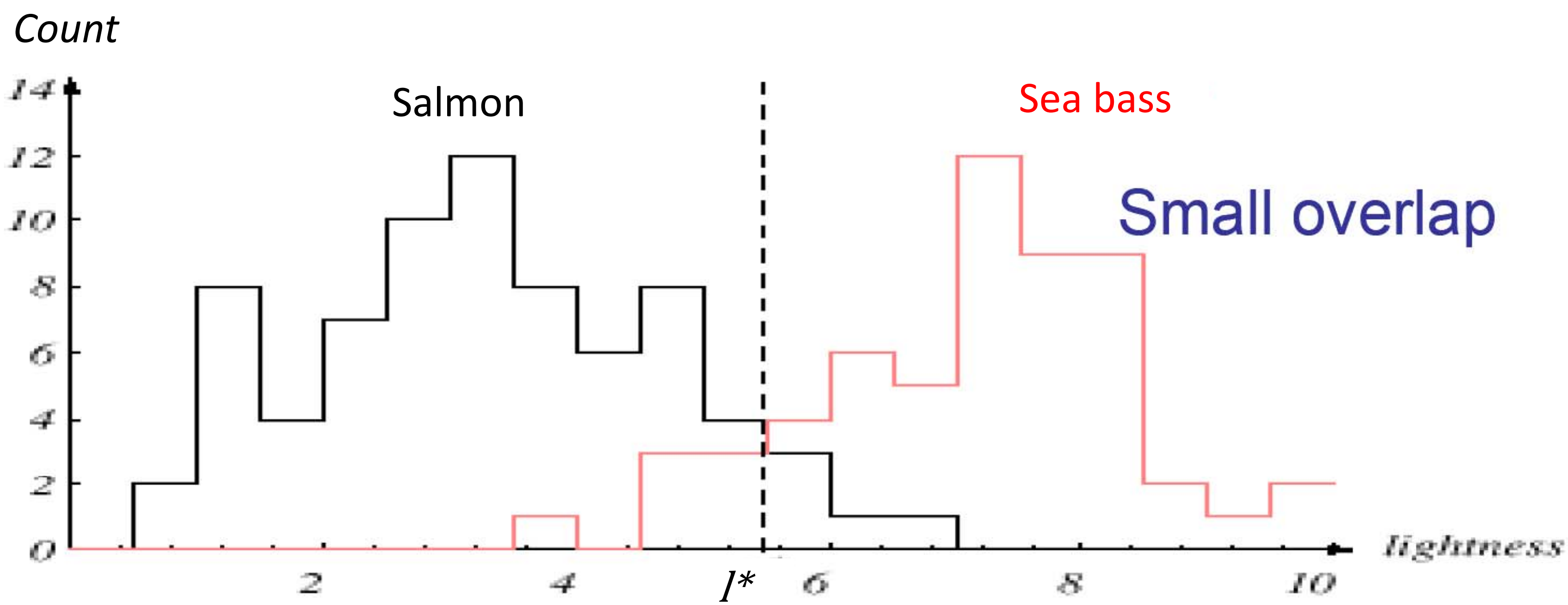
- Prior knowledge:
  Suppose we know that a sea bass is generally longer than a salmon.

Length Histograms



Two classes cannot be reliably separated!

Lightness Histograms



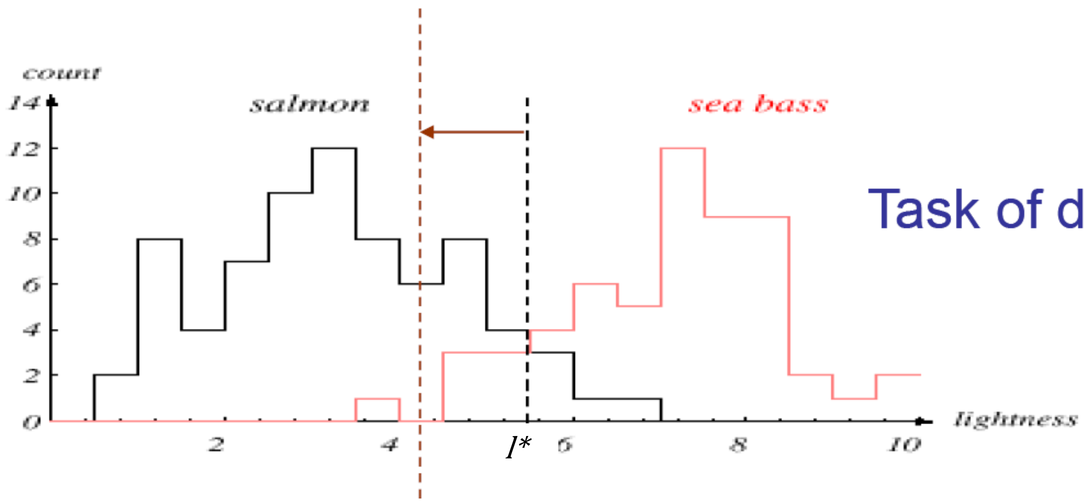
Two classes are much better separated!

Cost of Misclassification

- Two types of classification errors:
  - Deciding the fish was a sea bass when it was a salmon
  - Deciding the fish was a salmon when it was a sea bass
- Cost of different errors?
  - Which error is more costly?
- Prior knowledge:
  - Salmon is more expensive (tastier) than sea bass.
  - Customers easily accept pieces of salmon in cans labeled sea bass.
  - Customers are very annoyed by pieces of sea bass in cans labeled salmon.

Threshold Decision Boundary and Cost Relationship

- To increase customer satisfaction:
  - Reduce the number of sea bass that are classified as salmon
  - Move the decision boundary toward *smaller* values of lightness



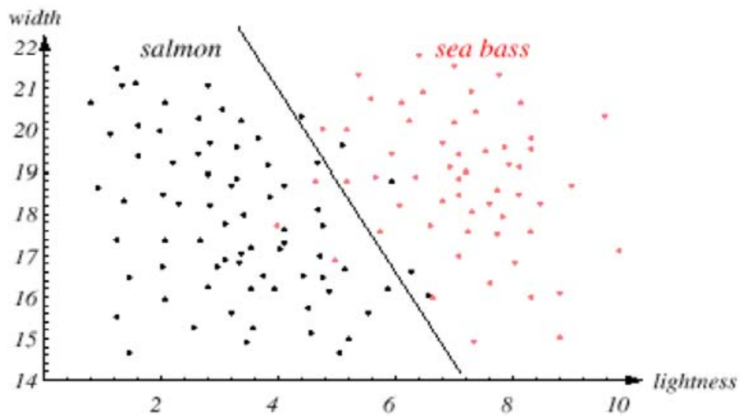
Task of decision theory

Decision Making Using Multiple Features

Decision Making Using Multiple Features

- Use more than one feature at a time
  - Single features might not yield the best performance.
  - Combinations of features might yield better performance.

- Two features:
  - lightness and width of the fish: $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ $\quad x_1: lightness$ $\quad x_2: width$



Decision boundary - partition the feature space into two regions.

Two features together are better than individual features.

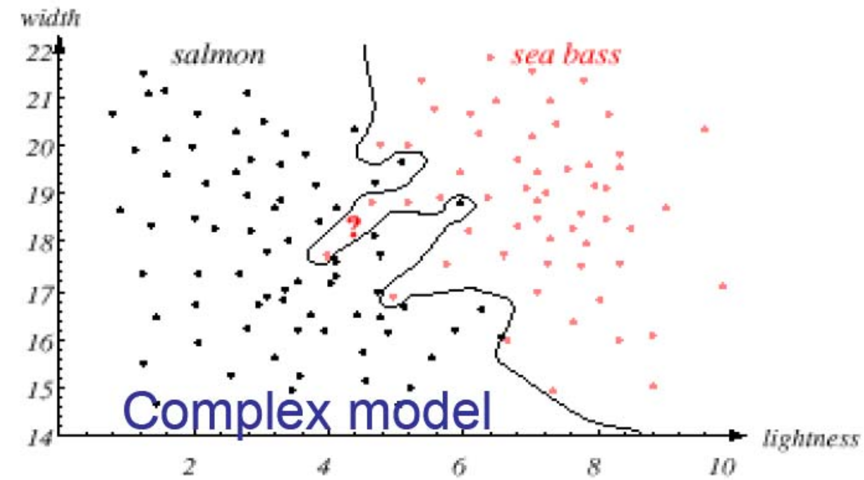
How Many Features and Which Ones?

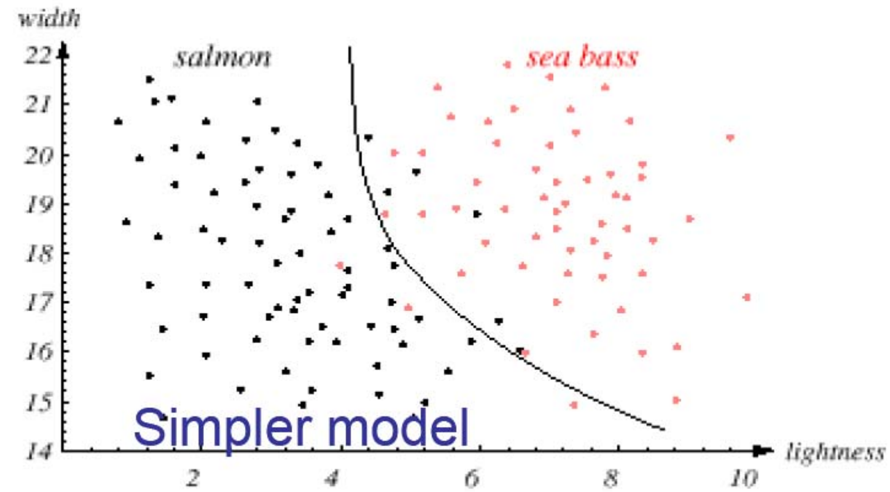
- Issues with feature extraction/selection:
  - It may be difficult to extract certain features.
  - It might be computationally expensive to extract many features.
  - Correlated features do not improve performance.
  - Some features may be redundant.

- “Curse” of dimensionality
  - Problems with too many features especially when we have a small number of training samples.
  - Adding too many features can, paradoxically, lead to a worsening of performance.

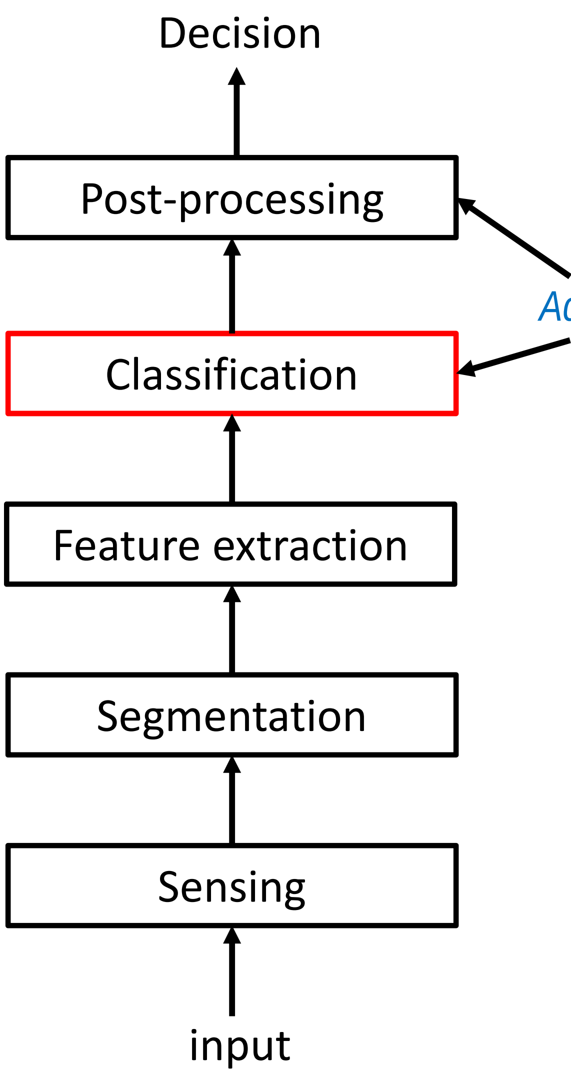
Generalization

- Model complexity
  - Complex models can achieve perfect classification on training data.
  - Too *tuned* to particular training samples, rather than some true model (**overfitting**)
- Issue of generalization:
  - Generalization: the ability of the classifier to produce correct results on *novel* patterns
  - How to get good generalization with a limited number of examples?
- Tradeoff between performance on the training set and Simpler model simplicity of classifier





Pattern Recognition Systems



Exploit context dependent information to improve system performance

Adjustments for context

Use feature values to assign the object to a category

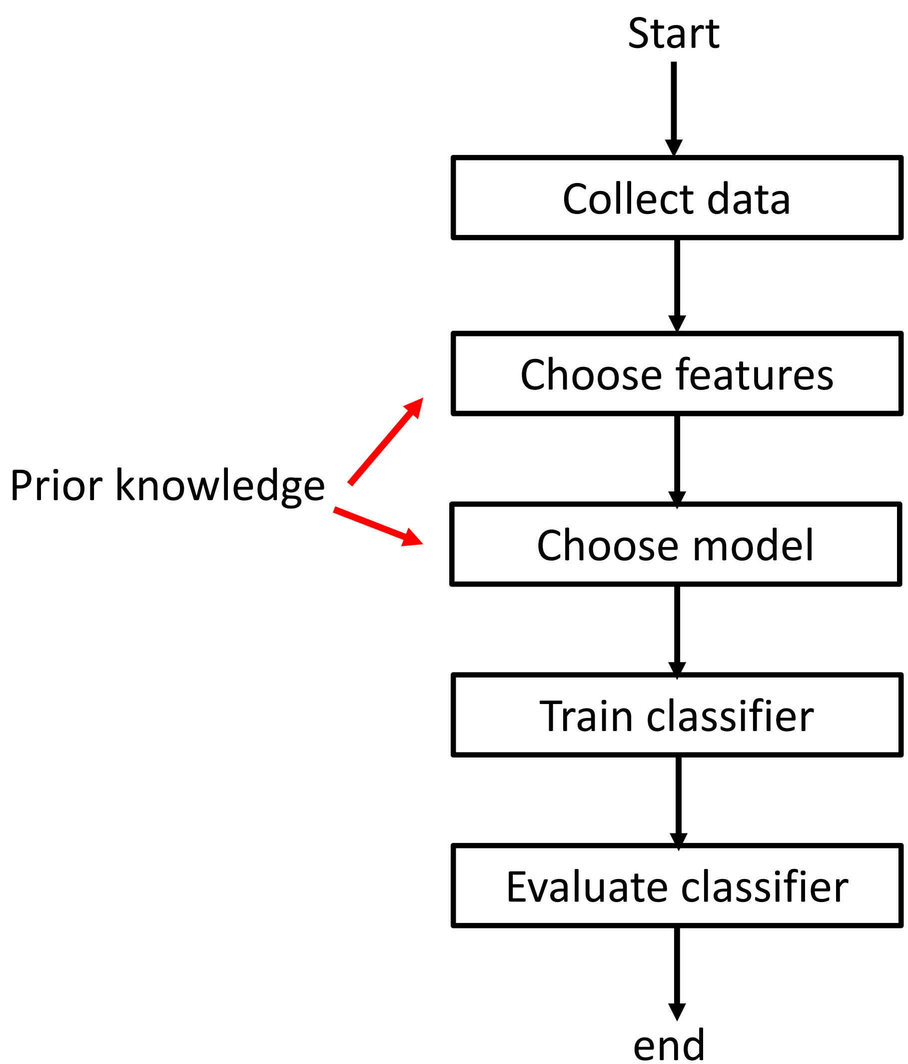
Extract discriminative & invariant features

Patterns should be well separated (no overlapping)

Using a transducer: camera, microphone, medical imaging devices, etc.

How much information are you missing?

The Design Cycle



The Design Cycle

- **Data Collection**
  - Can be quite costly.
  - How do we know when we have collected an adequately large and representative set of examples for training and testing the system?

- **Feature Choice**
  - Domain specific: use of prior knowledge.
  - How many and which ones to use?
  - Should be discriminative.
  - Simple to extract, invariant to irrelevant transformation, and insensitive to noise.

The Design Cycle

- **Model Choice**
  - What type of classifier to use?
  - When to reject a class of models and try another one?
  - What is the best classifier for the problem?
  - A trial and error process?

- **Training**
  - Use data to determine the parameters of the classifier.
  - Many different procedures for training classifiers.
  - Time consuming process

The Design Cycle

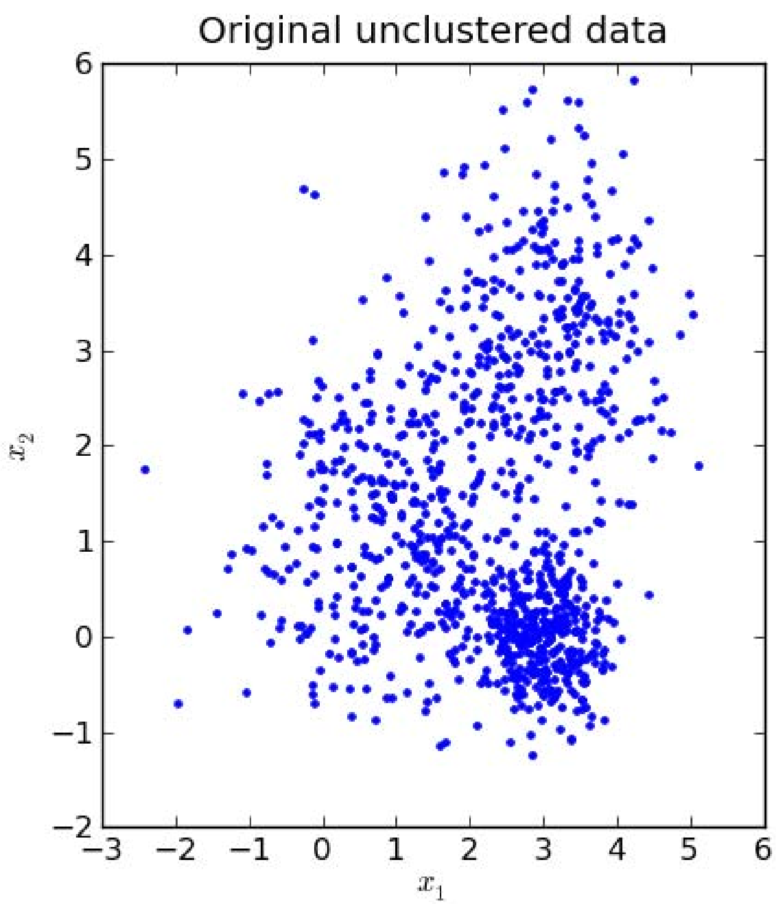
- **Evaluation**
  - Measure system performance, i.e., the error rate for:
    - different feature sets
    - different training methods
    - different training and test data sets
  - Identify the need for improvements.
  - How does an algorithm scale with the number of features, patterns or categories?

- **Trade-off** between computational complexity and performance.
  - Perfect classification results $\leftrightarrow$ impractical time and memory requirements

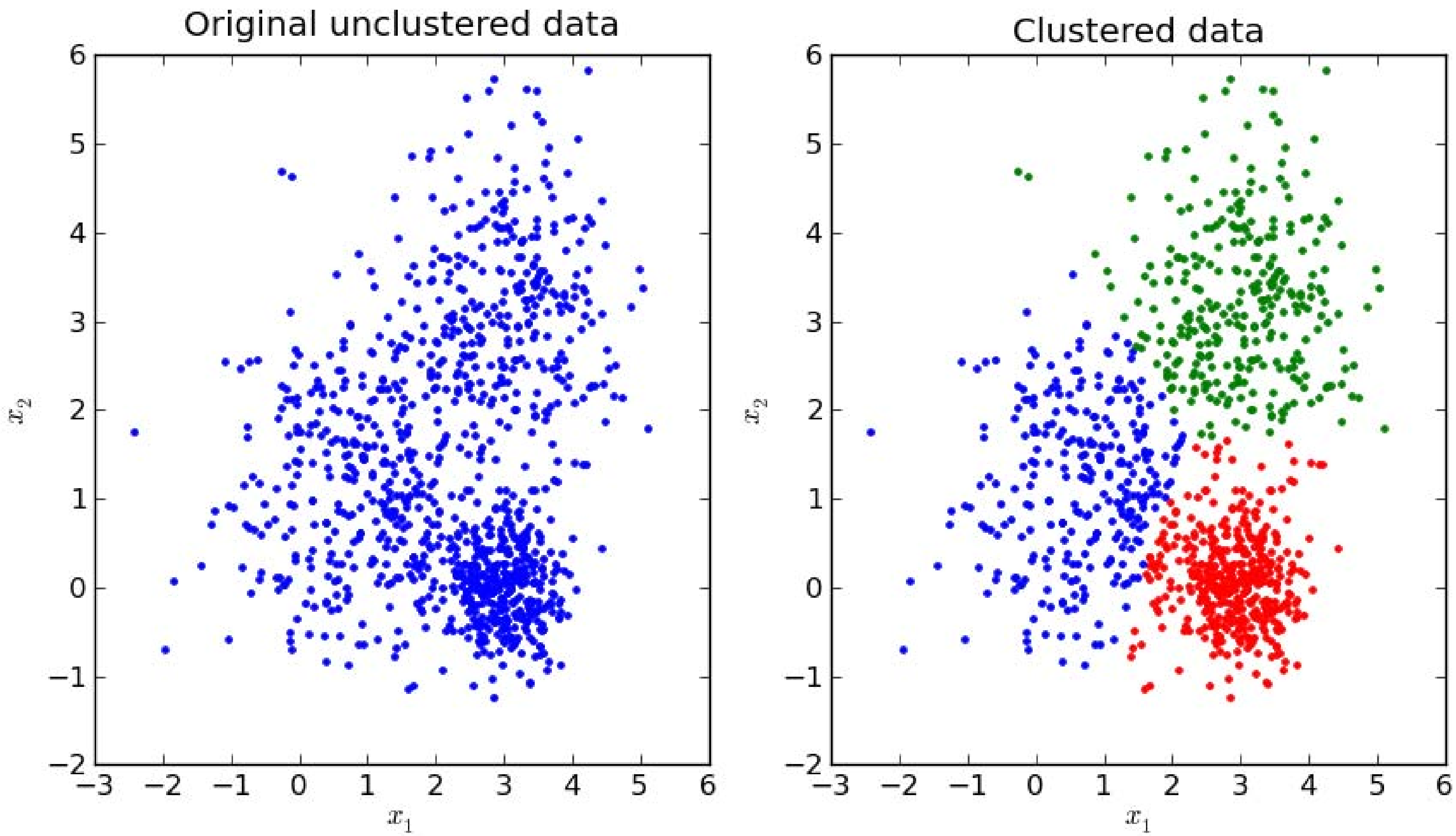
Supervised & Unsupervised Learning

- **Supervised learning**
  - Training samples are labeled.
  - Labeling by experts can be very expensive.
- **Unsupervised learning**
  - Training samples are unlabeled.
  - The system forms clusters or “natural groupings” of the input patterns.
- **Semi-supervised learning**
  - Use both labeled and un-labeled patterns to reduce the labeling cost

Unsupervised Learning



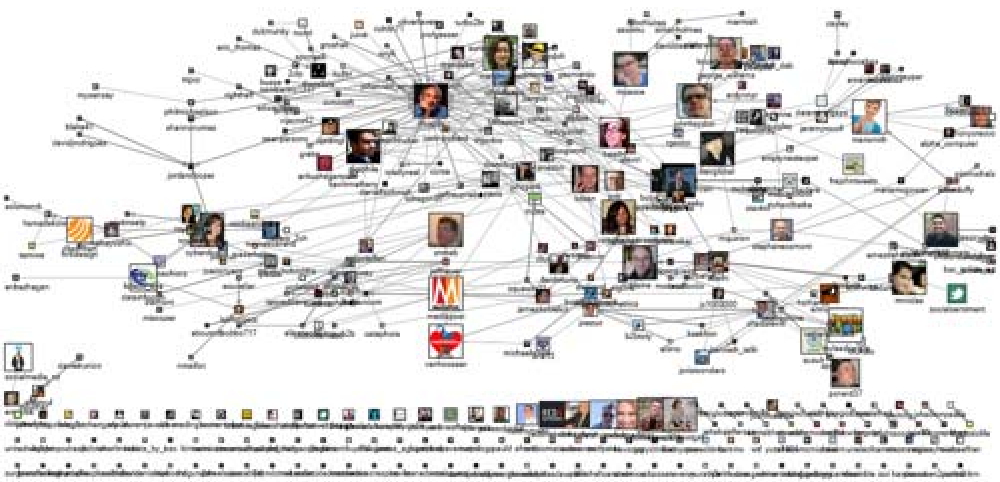
Unsupervised Learning



Unsupervised Learning

Organize computing clusters

Market segmentation



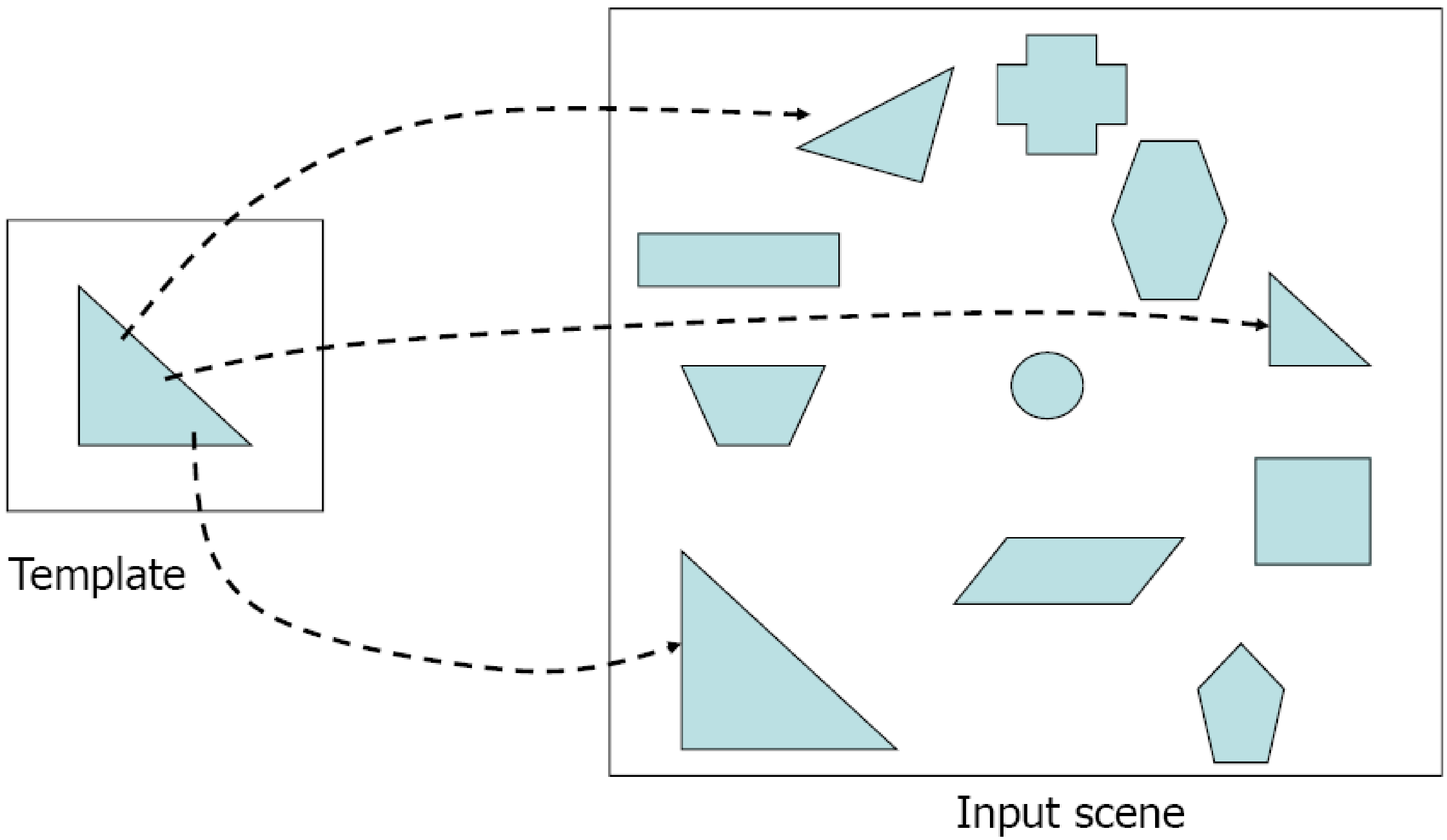
Social network analysis

Image Segmentation

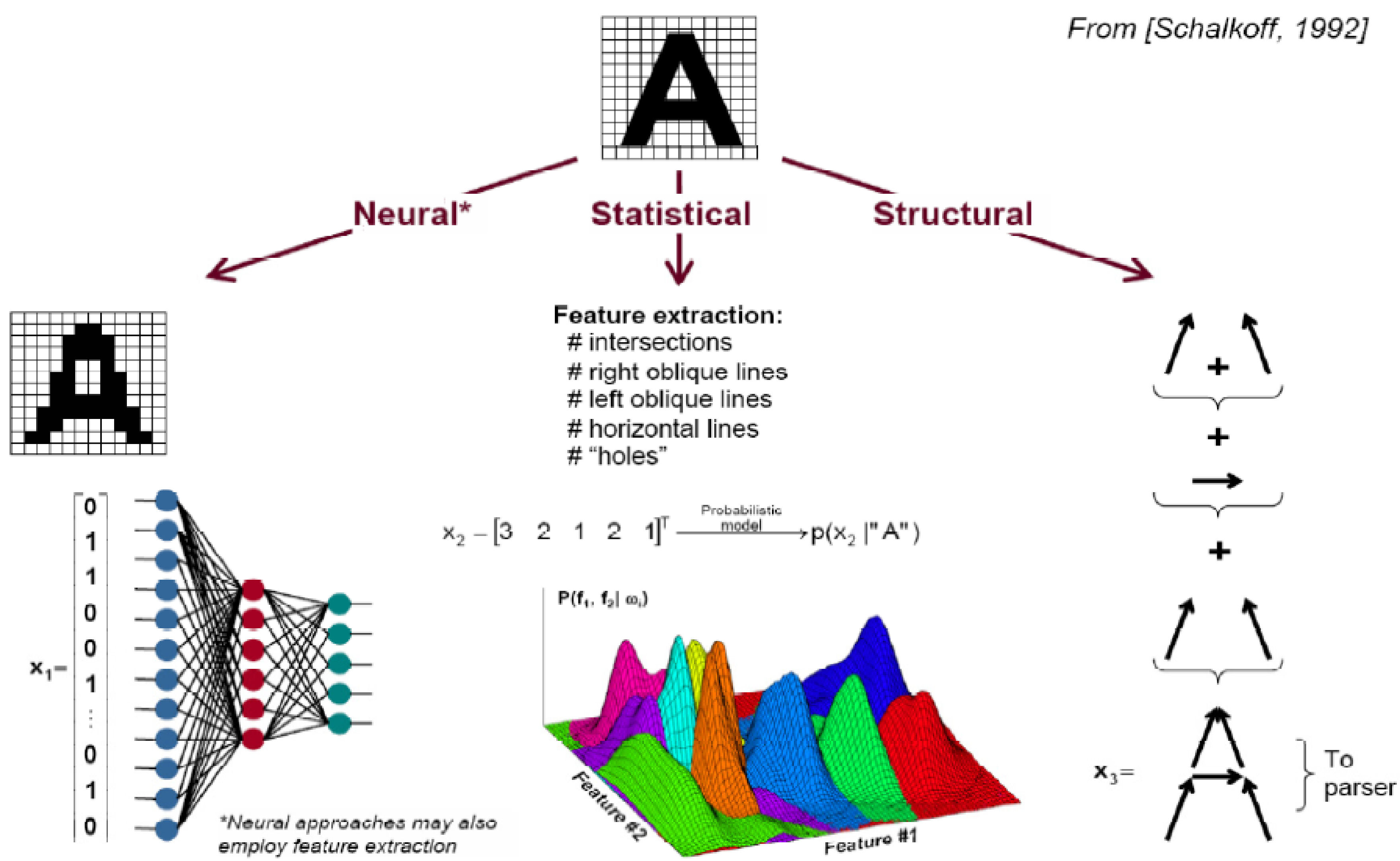
Machine Learning Approaches

- **Template matching**
  - Matches pattern against a stored template while taking into account all allowable <u>pose</u> and <u>scale</u> changes.
- **Artificial Neural Networks**
  - Inspired by biological neural network models
- **Statistical**
  - Focuses on the statistical properties of the patterns
- **Structural (Syntactic)**
  - Describe complicated objects in terms of simple primitives and structural relationships.
  - Decisions grammars consist of logical rules or grammars.

Template Matching



Neural, Statistical and Structural



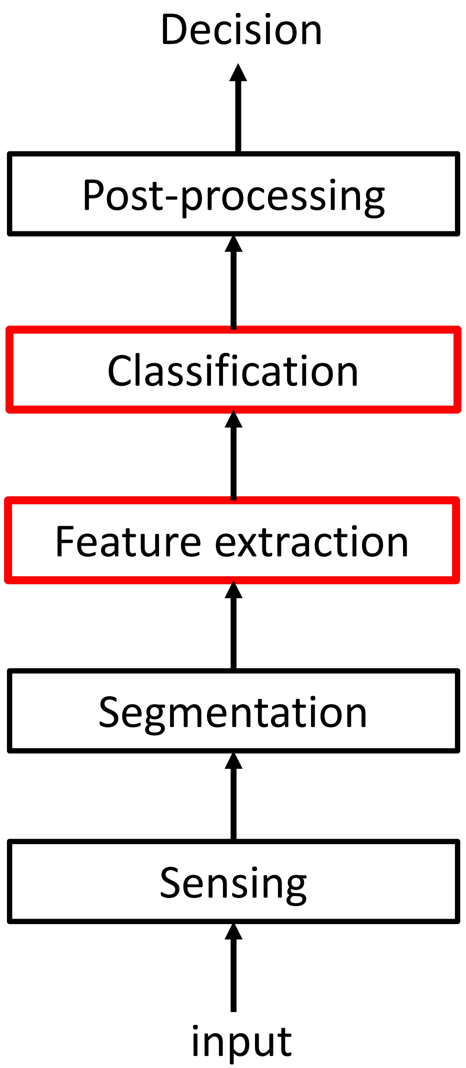
Comparison of Approaches

- **Template matching**
  - Assumes very small intra-class variability
  - Learning is difficult for deformable templates.
- **Statistical**
  - Assumption of density model for each class
- **Structural (Syntactic)**
  - Primitive extraction is sensitive to noise.
  - Describing a pattern in terms of primitives is difficult.
- **Artificial Neural Networks**
  - Parameter tuning and local minima in learning

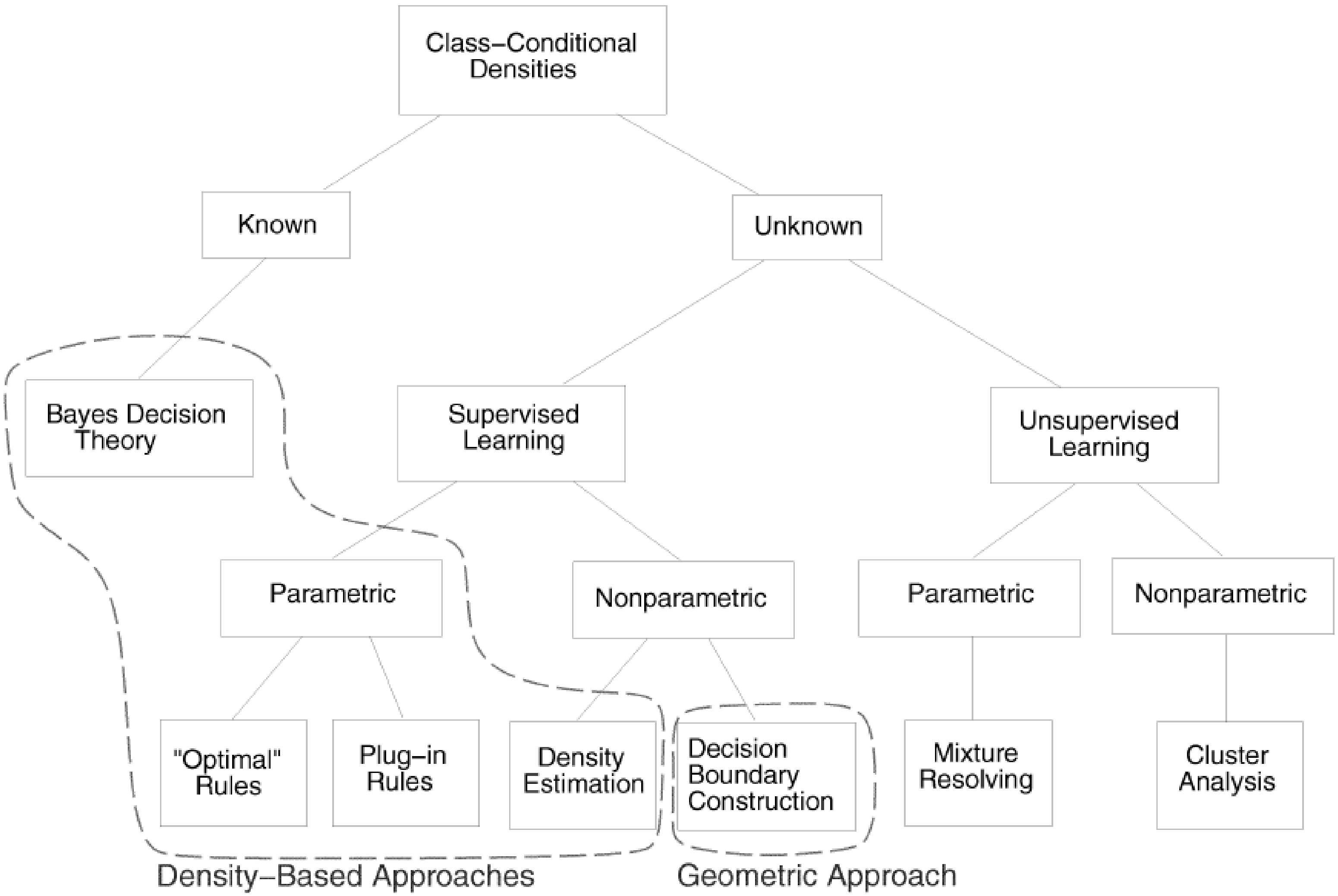
Our Focus

- Feature extraction and Classification
  – Assume basic pattern elements prepared already from raw data
  – Assume features extracted already initially
  – Feature extraction using methods taught

- Focus on
  – Represent and describe *statistical* characteristics of the pattern elements through statistical analysis
  – Emphasize the main concepts behind the mathematical formula whenever possible

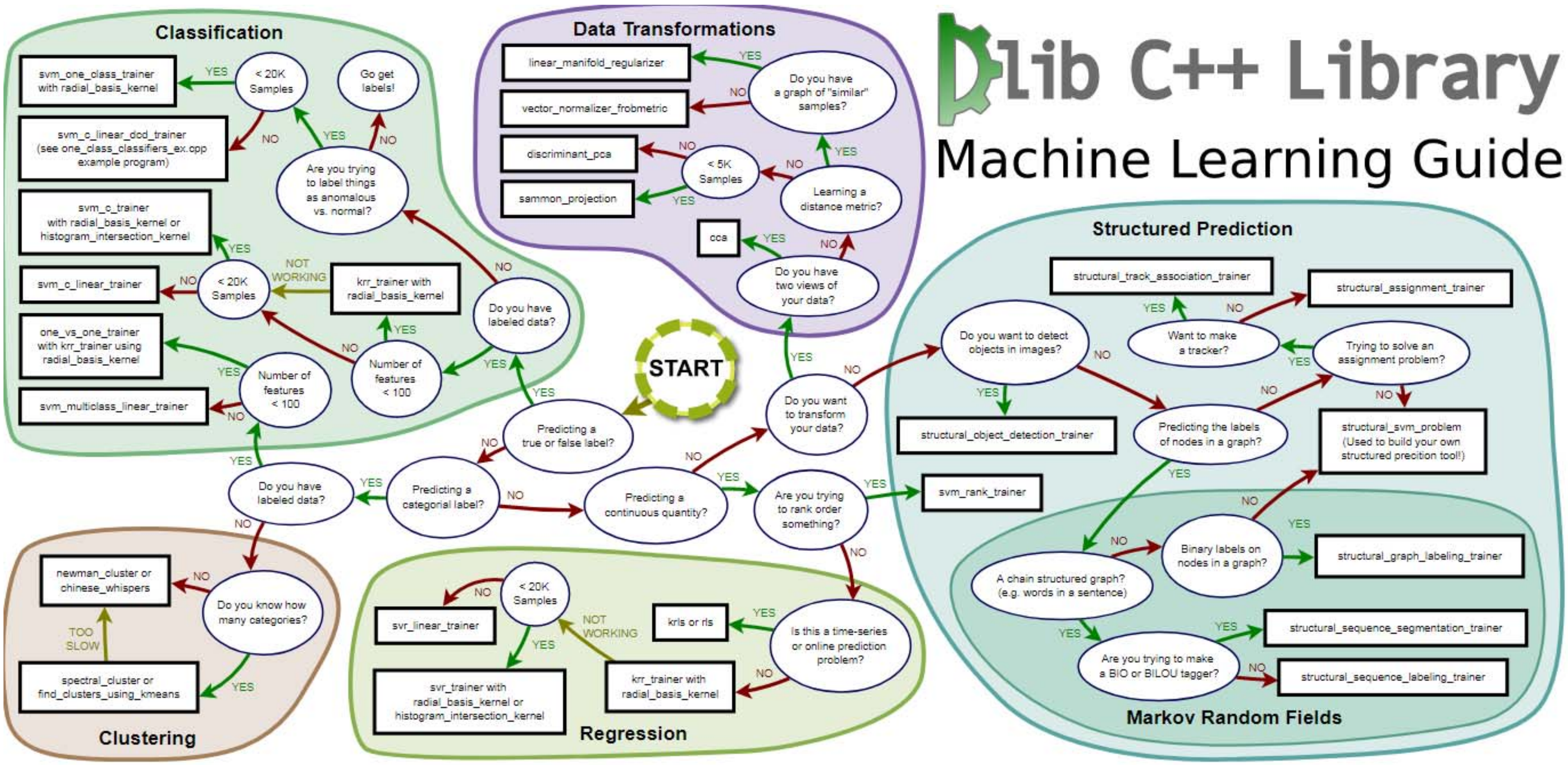


Approaches in Pattern Recognition



From: A.K. Jain, R.P.W. Duin and J. Mao, "Statistical pattern recognition: a review," IEEE Trans. PAMI, vol. 22, no. 1, 2000, 4-37

Approaches in Machine Learning



Dlib Machine Learning Toolkit

Conclusions

- Machine learning systems aim to recognize patterns based on their features.
- Machine learning is extremely useful - widely used in a lot of exciting and important applications.
- Machine learning is a very difficult task – many issues must be solved in order to build a successful machine learning application.
- Challenges remain to achieve human like performance.

Readings

- A.K. Jain, R.P.W. Duin and J. Mao, "Statistical pattern recognition: a review," IEEE Trans. PAMI, vol. 22, no. 1, 2000, 4-37: Section 1 and 2