Pattern Classification

EET3053

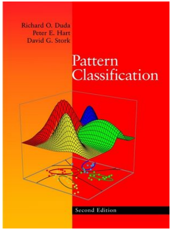
Lecture 01: Introduction

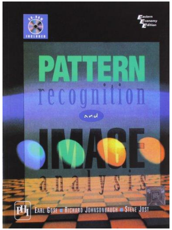
Dr. Kundan Kumar
Associate Professor
Department of ECE

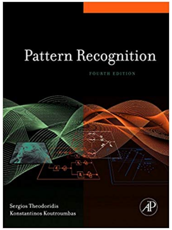
Faculty of Engineering (ITER)
S‘O’A Deemed to be University, Bhubaneswar, India-751030

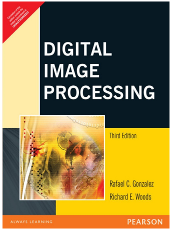
Outline

Text Books









Text Book:

- Pattern Classification, Duda-Hart, 2nd Edition
- Pattern Recognition and Image Analysis by Earl Gose
- Pattern Recognition by Theodoridis, 4th Edition
- Digital Image Processing by Gonzalez, 3rd Edition

Credits:

- 4 credits course, 4 Classes/week (1hr/Class)
- Prerequisite: MTH 2002 (Probability and Statistics)

Grading Pattern

- **Grading pattern**: 6

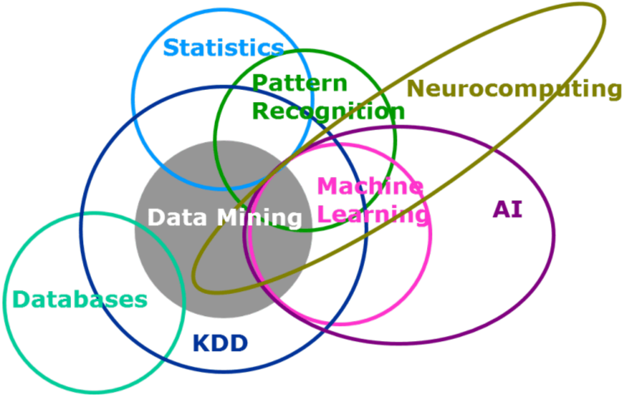
Attendance		5 Marks
Quiz/Assignment	:	10 Marks
Term Project	:	10 Marks
Mid-term examination	:	15 Marks
Total Internal	:	**40 Marks**

Theory examination	:	60 Marks
Total External	:	**60 Marks**

Syllabus

- **Syllabus:**
  - Introduction
  - Features Extraction
  - Bayesian Decision Theory
    - Continuous Features
    - Discrete Features
  - Parametric and Non-parametric Estimation Techniques
  - Component and Discriminant Analysis
    - Principal Component Analysis (PCA)
    - Fisher Linear Discriminant Analysis (FLD)
  - Linear Discriminant Functions
  - Support Vector Machine
  - Multilayer Neural Networks
  - Unsupervised Learning (Clustering)

Introduction to Pattern Classification



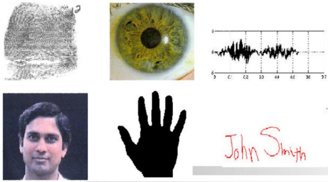
Relation with AI and ML

- **Artificial Intelligence**
  - The theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.
  - Any technique which enables computers to mimic human behavior.
- **Machine learning**
  A field of computer science that uses statistical techniques to give computer systems the ability to "learn" with data without being explicitly programmed and progressively improve performance on a specific task .
- **Pattern Classification**
  - Pattern classification is a sub-topic of machine learning.
  - Pattern classification can be defined as a technique to classify data (patterns) based either on a priori knowledge or statistical information extracted from the patterns.
  - Pattern recognition automatically discover the regularities in data through the use of learning algorithms.

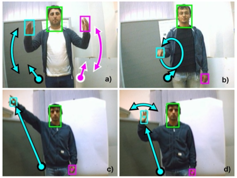
What is Pattern?

- A **pattern** is an entity, vaguely defined, that could be given a name, e.g.,
  - fingerprint image,
  - handwritten word,
  - human face,
  - speech signal,
  - DNA sequence, . . .
- A pattern could be an **object** or **event**.

Biometric pattern

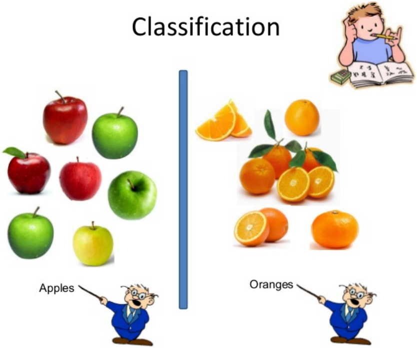


Hand gesture pattern



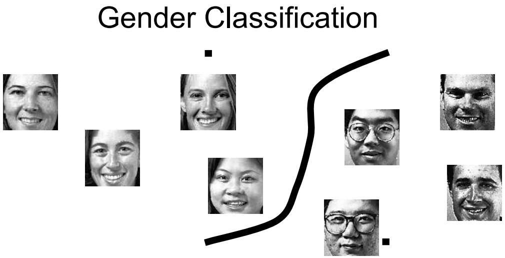
What is Pattern Classification?

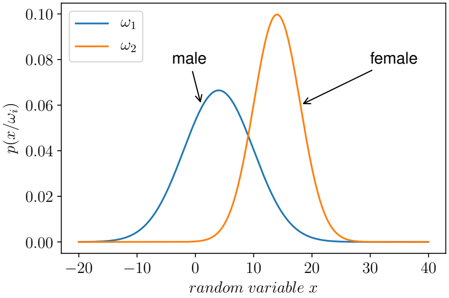
- **Pattern classification** is the study of how machines can
  - observe the environment,
  - learn to distinguish patterns of interest,
  - make sound and reasonable decisions about the categories of the patterns.



How do we model a Pattern Class?

- Typically, using a statistical model.
- Probability density function (e.g., Gaussian)





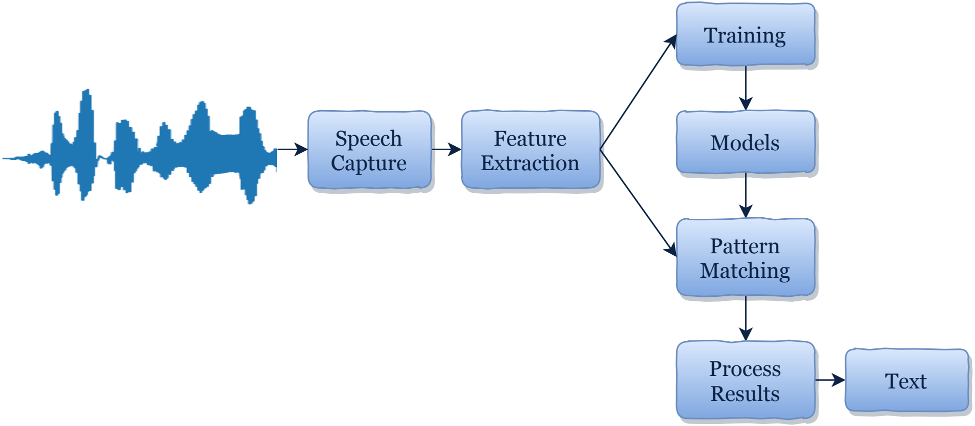
Applications

Machine Perception

Build a machine that can recognize patterns:

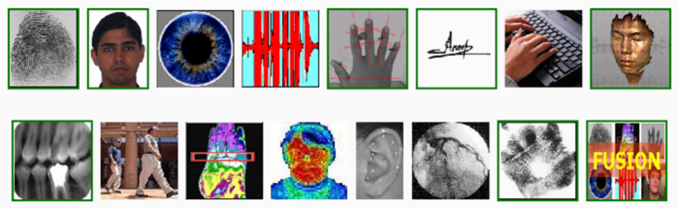
- Speech recognition
- Biometric recognition
- Fingerprint identification
- Face recognition
- OCR (Optical Character Recognition)
- DNA sequence identification
- Autonomous navigation

Applications: Speech Recognition

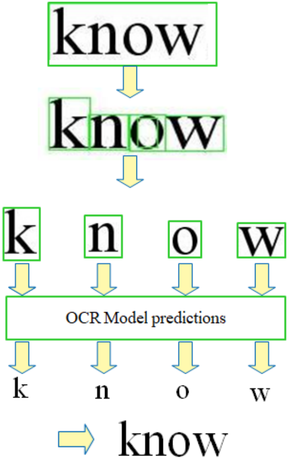


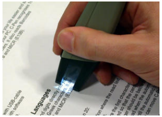
Applications: Biometric Recognition

- Fingerprint recognition
- Eyes - Iris recognition
- Face recognition, identification/verification
- Speech recognition
- Finger geometry recognition
- Hand geometry recognition
- Signature recognition
- Eyes - Retina recognition
- Typing recognition
- Gait recognition
- DNA identification



Applications: Optical Character Recognition (OCR)





1. Differentiate word contours associated with Image.
2. Differentiate letter contours associated with word contours associated with word contour image.
3. Preprocess letter images according to trained OCR input
4. Consolidate predictions associated OCR model to text.

Applications: Cancer Detection

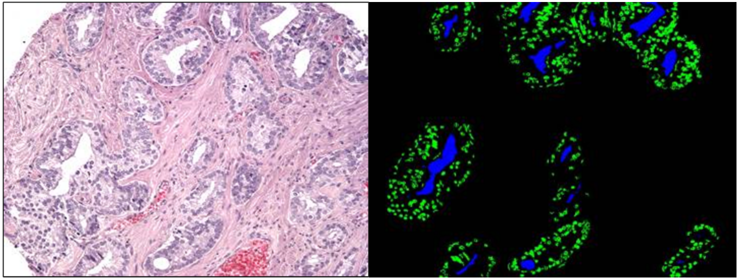


Figure: Cancer detection and grading using microscopic tissue data

Applications: Land Cover Detection

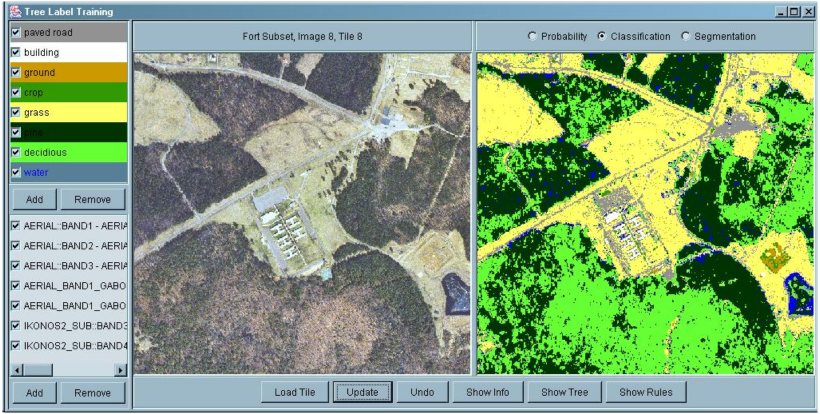


Figure: Land cover classification using satellite image

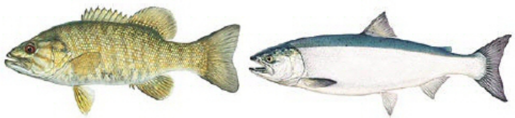
Applications: License Plate Recognition

Figure: License plate recognition: US license plates

A classic example to understand Pattern Classification

An Example

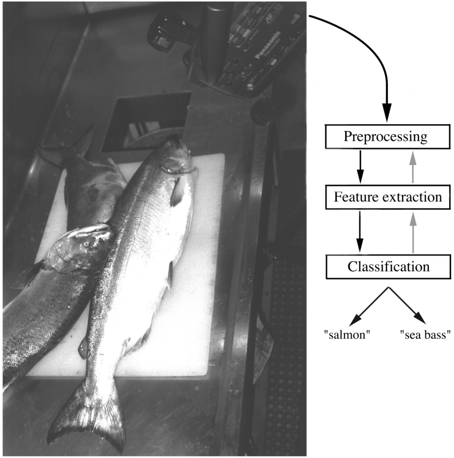
- "Sorting incoming Fish on a conveyor according to species using optical sensing"
- Species
  - Sea bass
  - Salmon



- Set up a camera and take some sample image to extract features
  - Length
  - Lightness
  - Width
  - Number and shape of fins
  - Position of the mouth, etc.
- This is the set of all suggested features to explore for use in our classifier.

An Example

- What can cause problems during sensing?
  - lighting conditions,
  - position of fish on the conveyor belt,
  - camera noise, etc.
- Use a segmentation operation to isolate fishes from one another and from the background.
- Information from a single fish is sent to a feature extractor whose purpose is to reduce the data by measuring certain features.
- The features are passed to a classifier.



An Example: Classification

- Select the length of the fish as a possible feature for discrimination

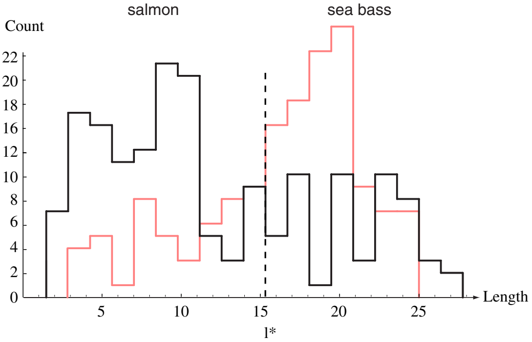


Figure: Histograms for the length feature for the two categories. No single threshold value l^* (decision boundary) will serve to unambiguously discriminate between the two categories; using length alone, we will have some errors. The value $l*$ marked will lead to the smallest number of errors, on average.

An Example: Classification

- The length is a poor feature alone
- Select the lightness as a possible feature

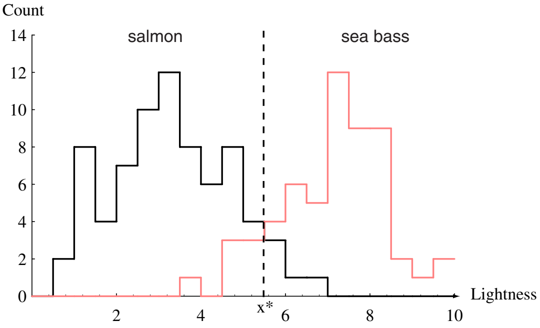


Figure: Histograms for the lightness feature for the two categories. No single threshold value x^* (decision boundary) will serve to unambiguously discriminate between the two categories; using lightness alone, we will have some errors. The value x^* marked will lead to the smallest number of errors, on average.

Threshold decision boundary and cost relationship

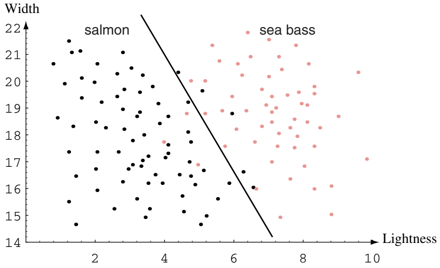
- Move our decision boundary toward smaller values of lightness in order to *minimize the cost* (reduce the number of sea bass that are classified salmon)

$$\Downarrow$$

Task of decision theory

An Example: Feature vector

- Adopt the lightness and add the width of the fish
- We can use two features in our decision:
  - lightness: x_1
  - width: x_2
- Each fish image is now represented as a point (feature vector) x in two-dimensional feature space.



$$\mathrm{x} = [x_1 \quad x_2]^T$$

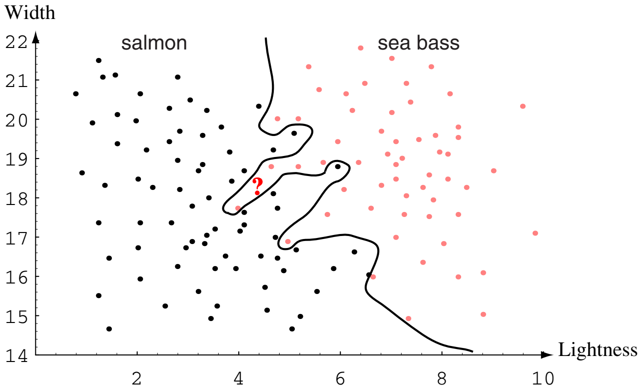
$x_1 \downarrow$ Lightness, $x_2 \downarrow$ Width

An Example: Feature vector

- We might add other features that are not correlated with the ones we already have. A precaution should be taken not to reduce the performance by adding such "noisy features".
- Does adding more features always improve the results?
  - unreliable features.
  - Be careful about correlations with existing features.
  - Be careful about measurement costs.
  - Be careful about noise in the measurements.
- Is there some *curse* for working in very high dimensions?

An Example: Feature vector

- Ideally, the best decision boundary should be the one which provides an optimal performance.



An Example: Issue of generalization

- How can we mange the *tradeoff* between complexity of decision rules and their performance to unknown samples?
- Our satisfaction is premature because the central aim of designing a classifier is to correctly classify novel input

$$\Downarrow$$

Issue of generalization

An Example: Decision Boundary

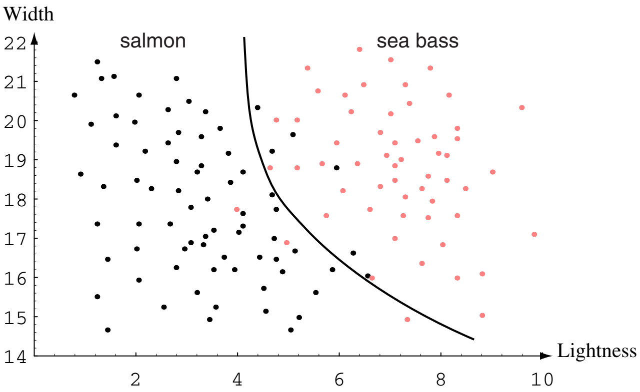


Figure: The decision boundary shown might represent the optimal trade of between performance on the training set and simplicity of classifier.

Pattern Recognition System and Design Cycle

Pattern Recognition Models

There are three main models of pattern recognition:

- Statistical:
  - To identify where specific piece belongs (for example, whether it is a cake or not).
  - Use of statistics to learn from examples.
- Syntactic/Structural:
  - To define a more complex relationship between elements taking into account more complex interrelationships between attributes.
  - Looks at clear structure in the patterns.
  - An example of this would be diagnosis of the heart with ECG measurements.
- Template Matching:
  - To match the object's features with the predefined template and identify the object by proxy.
  - One of the uses of such model is plagiarism checking.

Pattern Recognition Systems

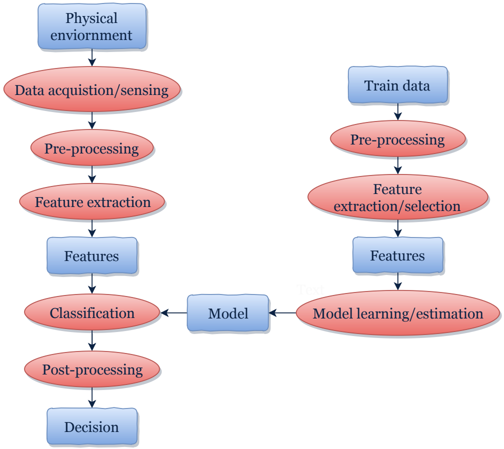


Figure: Object/Process diagram of a pattern recognition system

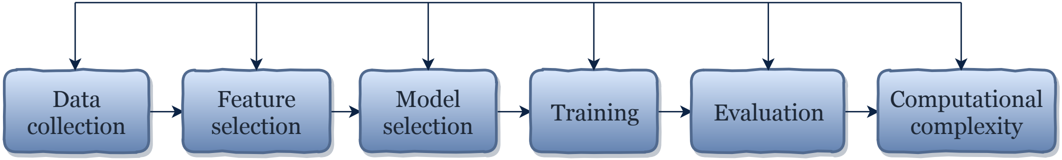
Pattern Recognition Systems

- Data acquisition and sensing:
  - Use of transducer (camera or microphone)
  - Measurements of physical variables.
  - Important issues: bandwidth, resolution, sensitivity, distortion, SNR, latency, etc.
- Pre-processing:
  - Removal of noise in data.
  - Isolation of patterns of interest from the background.
  - Segmentation and grouping
- Feature extraction:
  - Finding a new representation in terms of features.
  - Features should be well separated and should not overlap (Discriminative features)
  - Invariant features with respect to translation, rotation and scale.
  - Depends on the characteristics of the problem domain. Simple to extract, invariant to irrelevant transformation insensitive to noise.

Pattern Recognition Systems

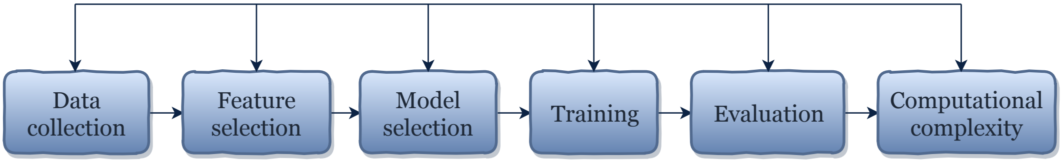
- Model learning and estimation:
  - Learning a mapping between features and pattern groups and categories.
  - Unsatisfied with the performance of classifier and want to jump to another class of model
- Classification:
  - Using features and learned models to assign a pattern to a category.
- Post-processing:
  - Evaluation of confidence in decisions.
  - Exploitation of context to improve performance.
  - Combination of experts.

The Design Cycle



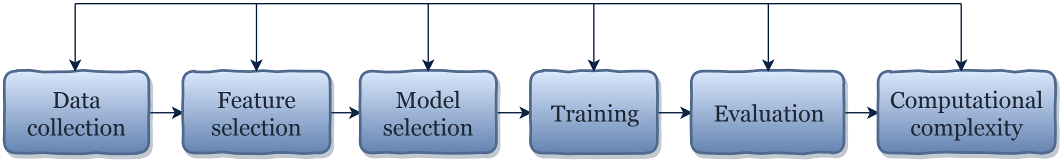
- Data Collection
  - Collecting training and testing data.
  - How do we know when we have collected an adequately large and representative set of examples for training and testing the system?

The Design Cycle



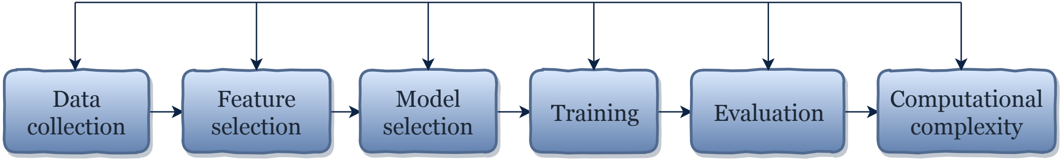
- Feature Selection
  - Domain dependence and prior information.
  - Computational cost and feasibility.
  - Discriminative features.
    - Similar values for similar patterns.
    - Different values for different patterns.
  - Invariant features with respect to translation, rotation and scale.
  - Robust features with respect to occlusion, distortion, deformation, and variations in environment.

The Design Cycle



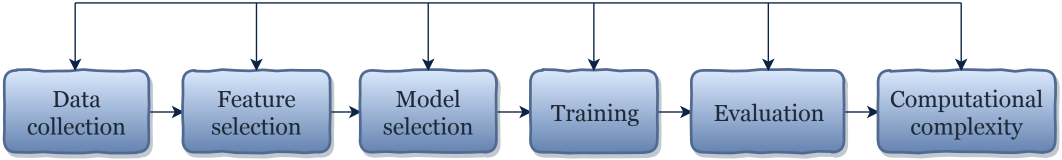
- Model selection
  - Domain dependence and prior information.
  - Definition of design criteria.
  - Parametric vs. non-parametric models.
  - Handling of missing features.
  - Computational complexity.
  - Types of models: templates, decision-theoretic or statistical, syntactic or structural, neural, and hybrid.
  - How can we know how close we are to the true model underlying the patterns?

The Design Cycle



- Training
  - How can we learn the rule from data?
  - Supervised learning: a teacher provides a category label or cost for each pattern in the training set.
  - Unsupervised learning: the system forms clusters or natural groupings of the input patterns.
  - Reinforcement learning: no desired category is given but the teacher provides feedback to the system such as the decision is right or wrong.

The Design Cycle



- Evaluation
  - How can we estimate the performance with training samples?
  - How can we predict the performance with future data?
  - Problems of overfitting and generalization.
- Computational Complexity
  - What is the trade-off between computational ease and performance?
  - How an algorithm scales as a function of the number of features, patterns or categories?

Different Learning Approaches

- Supervised learning/Classification
  - A teacher provides a category label or cost for each pattern in the training set
- Unsupervised learning/Clustering
  - The system forms clusters or "natural groupings" of the input pattern (no explicit teacher)
- Semi-supervised learning
  - Semi-supervised learning is the problem of learning from examples for which you have labels for only a (small) subset.
- Reinforcement learning
  - Learning with critic, no desired category is known; instead, the only teaching feedback is that the tentative category is right or wrong. It utilizes reward function to learn. Ex: Autonomous driving

References

[1] Hart, P. E., Stork, D. G., & Duda, R. O. (2000). Pattern classification. Hoboken: Wiley.

Thank you!