Pattern Recognition
CSE4213

Text Books

- Pattern Recognition
  - S. Theodoridis & K. Koutrumbas
- Pattern Classification
  - R. Duda *et al*.
- Pattern Recognition *Statistical, Structural and Neural Approaches*
  - R. Shalkoff

- Humans have developed highly sophisticated skills for sensing their environment and taking actions according to what they observe, e.g.,
  - recognizing a face,
  - understanding spoken words,
  - reading handwriting,
  - distinguishing fresh food from its smell.
- We would like to give similar capabilities to machines.

What is Pattern Recognition?

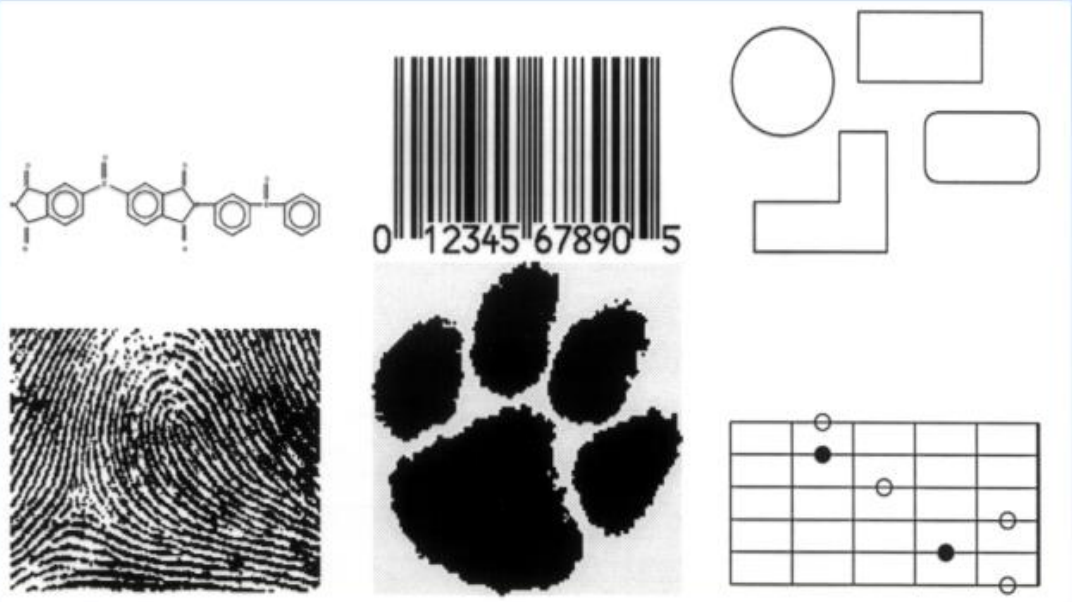
- A *pattern* is an entity, vaguely defined, that could be given a name, e.g.,
  - fingerprint image,
  - handwritten word,
  - human face,
  - speech signal,
  - DNA sequence,
  - ...
- *Pattern recognition* is the study of how machines can
  - observe the environment,
  - learn to distinguish patterns of interest,
  - make sound and reasonable decisions about the categories of the patterns.

Pattern Recognition

The act of taking as input sensed data (measurements) and taking an action based on the "category" or "class" of the pattern.

What is a Pattern?

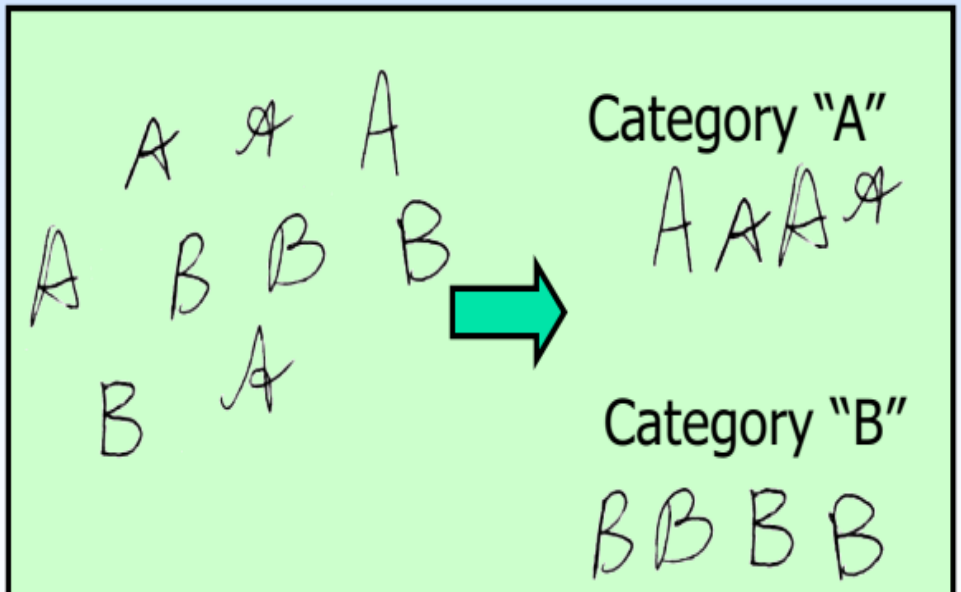
"A pattern is the opposite of a chaos; it is an entity vaguely defined, that could be given a name." (Watanabe)

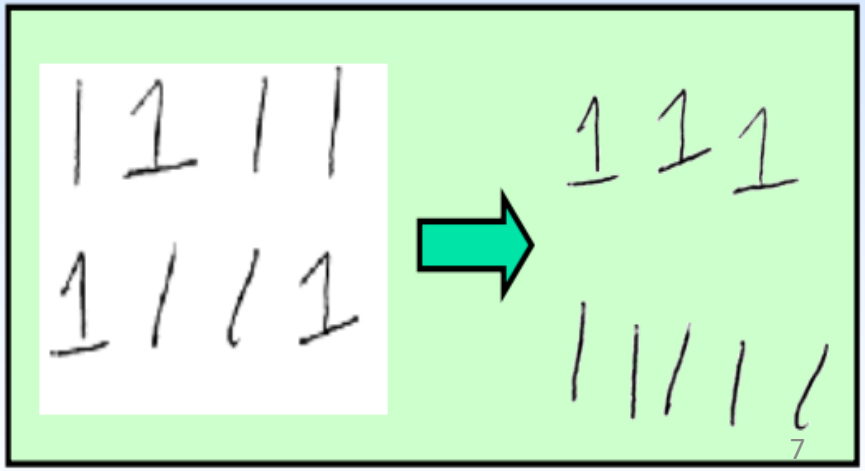


Recognition

Identification of a pattern as a member of a category (class) we already know, or we are familiar with

- Classification (known categories)
- Clustering (learning categories)





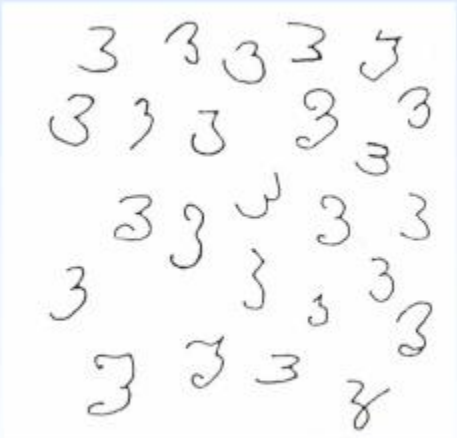
Clustering

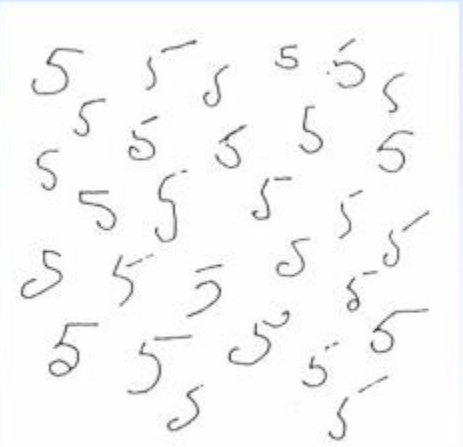
Pattern Class

- A collection of similar (not necessarily identical) objects
- A class is defined by class samples (exemplars, prototypes)
- Intra-class variability
- Inter-class similarity
- How to define similarity?

Intra-Class Variability

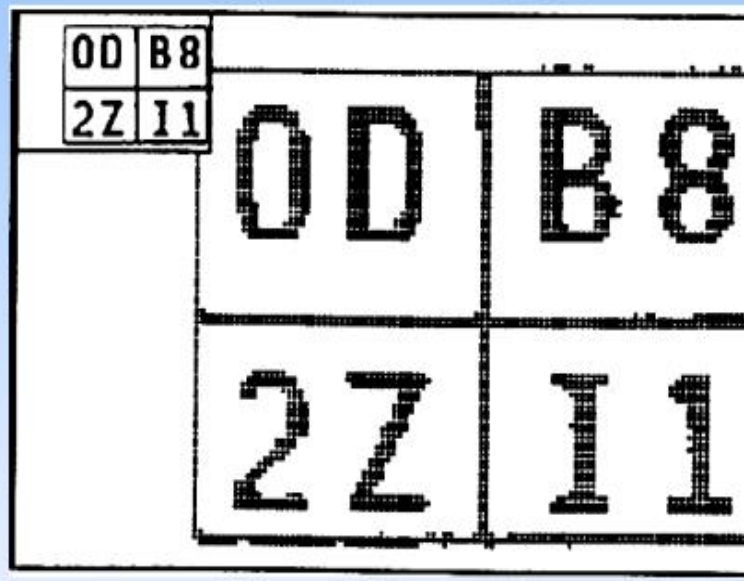
Handwritten numerals





Inter-class Similarity

Identical twins



Characters that look similar

Cat vs. Dog:
2-class Classification

(Supervised) Classification

Labeled training samples for classifier design

Clustering: Unsupervised Classification

Training samples are unlabeled

What It Does

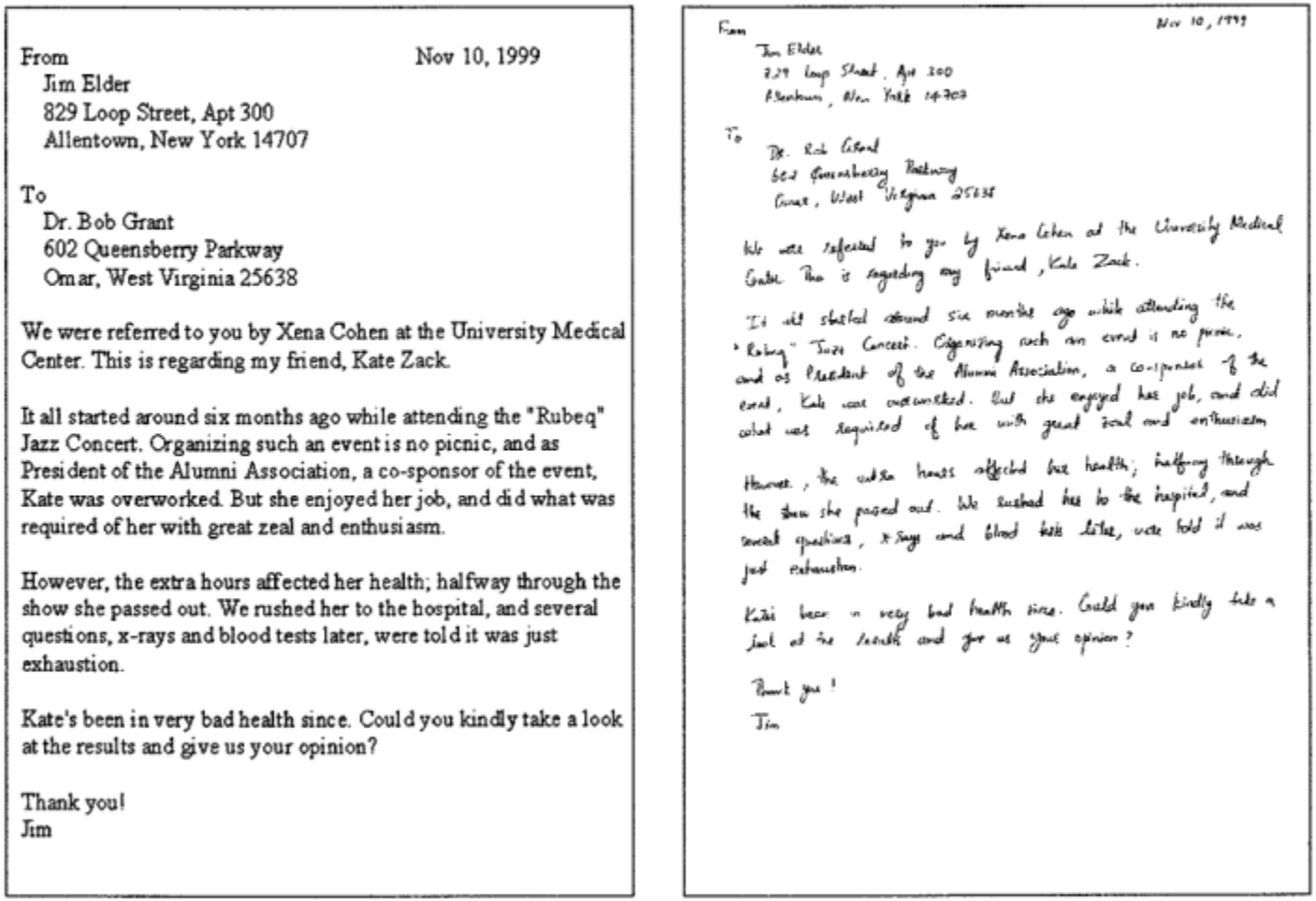
- Build a machine that can recognize patterns:

- The task: Assign unknown objects – patterns – into the correct class. This is known as classification.

What It Does

- Areas:
  - Machine vision
  - Character recognition (OCR)
  - Computer aided diagnosis
  - Speech recognition
  - Face recognition
  - Bioinformatics
    - Image Data Base retrieval
    - Data mining
    - Biometrics
      - Fingerprint identification
      - Iris Recognition
    - DNA sequence identification

Pattern Recognition Applications



From
Jim Elder
829 Loop Street, Apt 300
Allentown, New York 14707

Nov 10, 1999

To
Dr. Bob Grant
602 Queensberry Parkway
Omar, West Virginia 25638

We were referred to you by Xena Cohen at the University Medical Center. This is regarding my friend, Kate Zack.

It all started around six months ago while attending the "Rubeq" Jazz Concert. Organizing such an event is no picnic, and as President of the Alumni Association, a co-sponsor of the event, Kate was overworked. But she enjoyed her job, and did what was required of her with great zeal and enthusiasm.

However, the extra hours affected her health; halfway through the show she passed out. We rushed her to the hospital, and several questions, x-rays and blood tests later, were told it was just exhaustion.

Kate's been in very bad health since. Could you kindly take a look at the results and give us your opinion?

Thank you!
Jim

Figure 1: English handwriting recognition.

Pattern Recognition Applications

Figure 3: Fingerprint recognition.

Pattern Recognition Applications

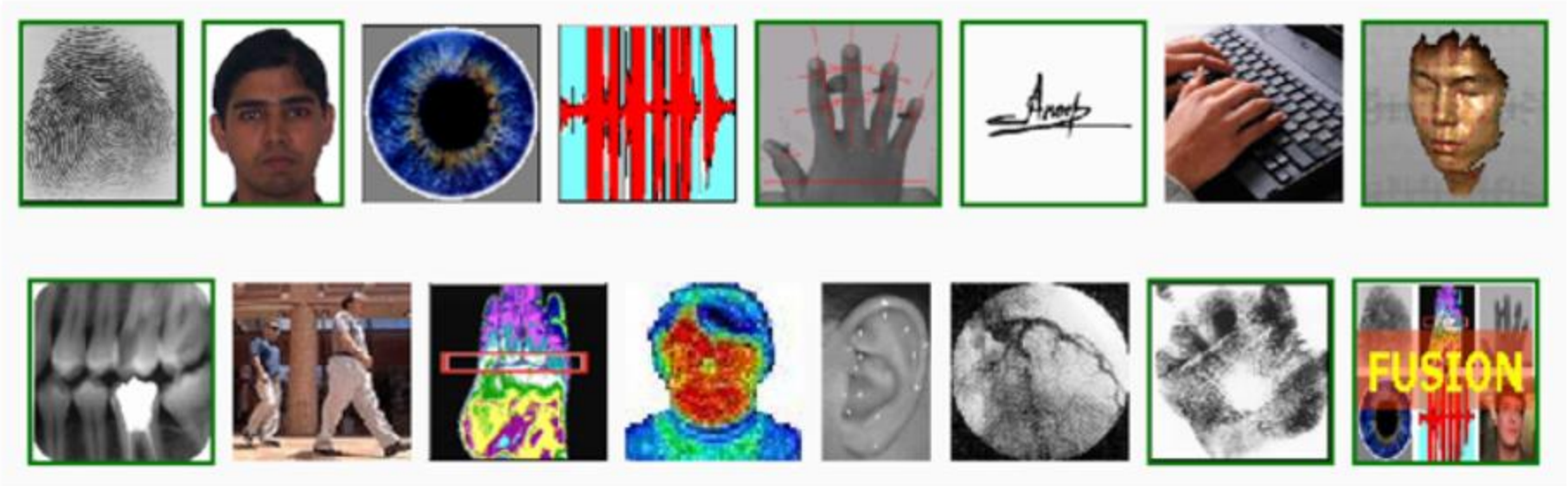


Figure 4: Biometric recognition.

Pattern Recognition Applications

Figure 5: Autonomous navigation.

Pattern Recognition Applications

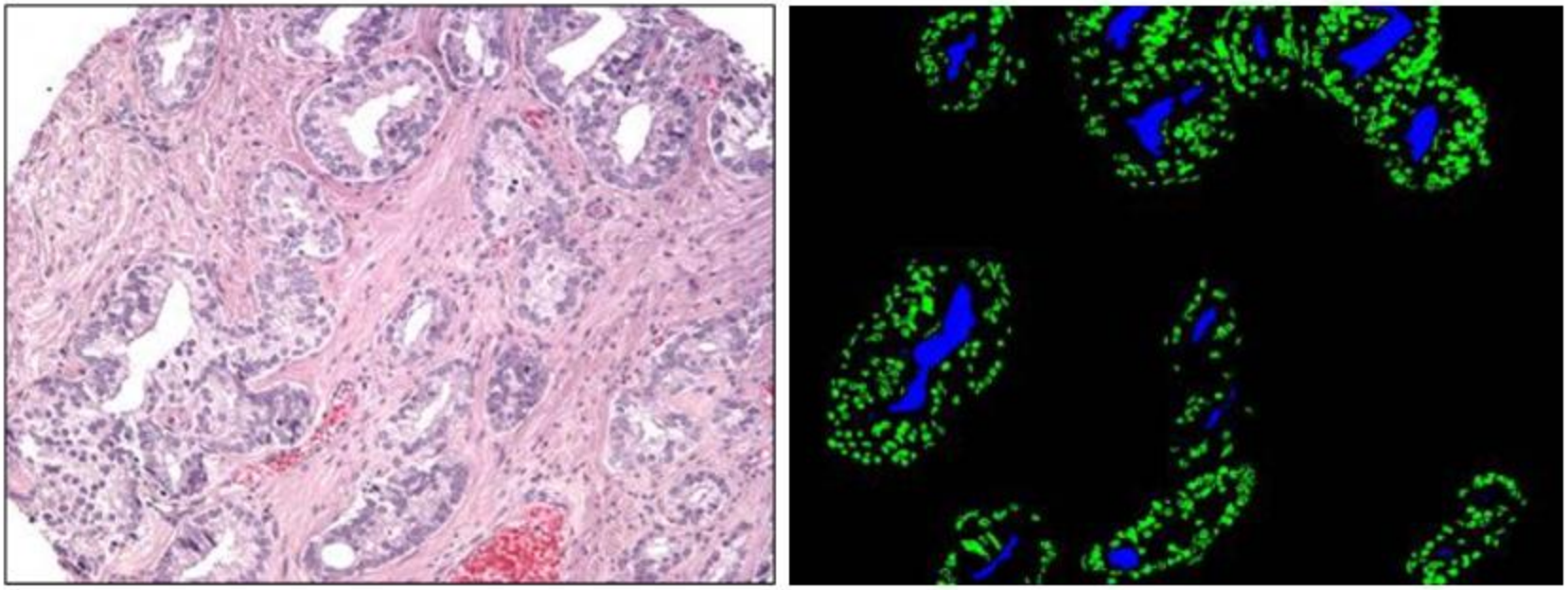


Figure 6: Cancer detection and grading using microscopic tissue data.

Pattern Recognition Applications

Figure 8: Land cover classification using satellite data.

Pattern Recognition Applications

Figure 9: Building and building group recognition using satellite data.

Pattern Recognition Applications

Figure 10: License plate recognition: US license plates.

Pattern Recognition Applications

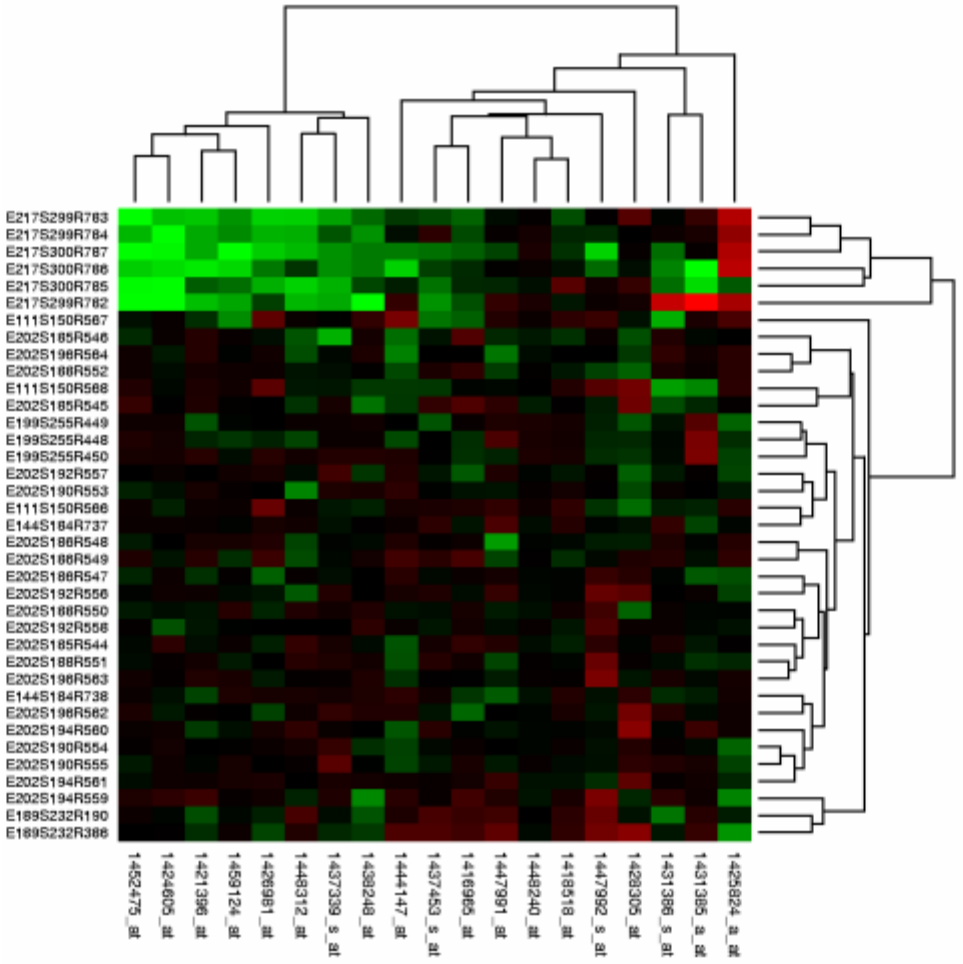


Figure 11: Clustering of microarray data.

Representation of patterns

- Features:
  - measurable quantities from the patterns
  - determines the classification task
- Feature vectors: A number of features

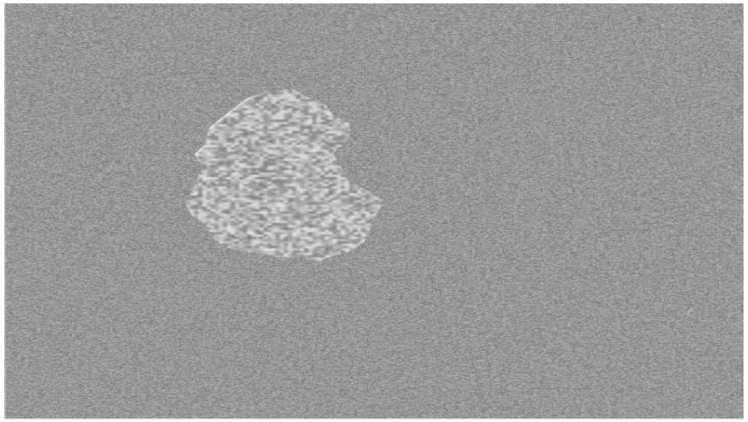
$$x_1, \ldots, x_l,$$

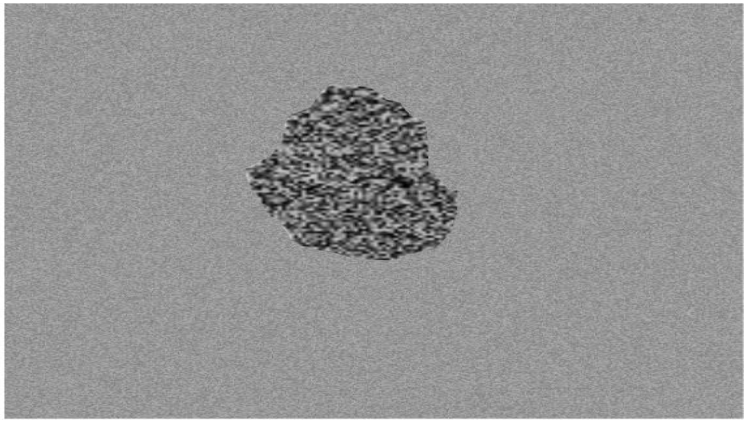
constitute the feature vector

$$\underline{x} = [x_1, \ldots, x_l]^T \in R^l$$

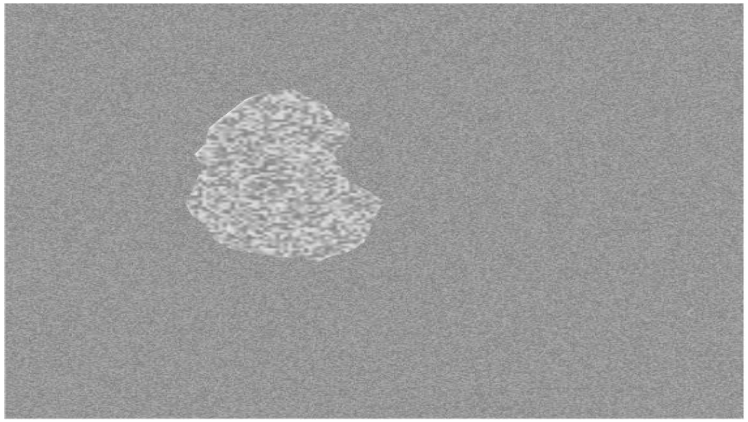
Feature vectors are treated as random vectors.

Example 1:

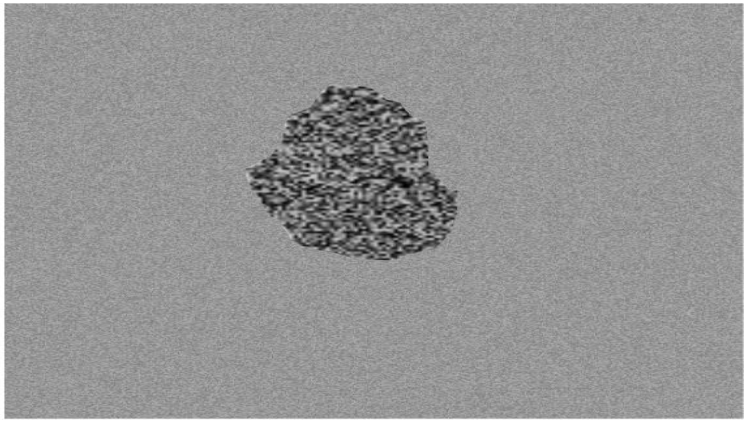




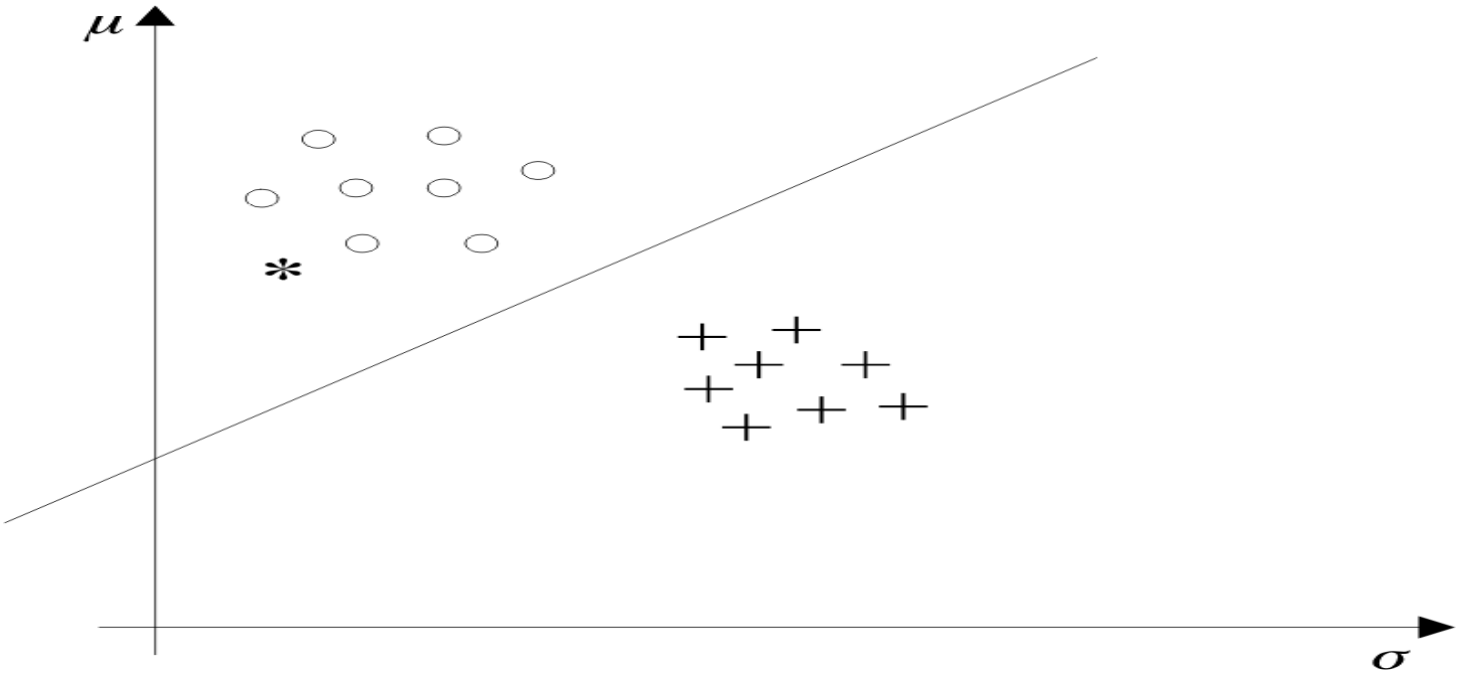
Example 1:



(a)



(b)



Issues in Pattern Recognition

- How are features generated?
- What is the best number of features?
- How are they used to design a classifier?
- How good is the classifier?

Example 2

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

Species

Sea bass

Salmon

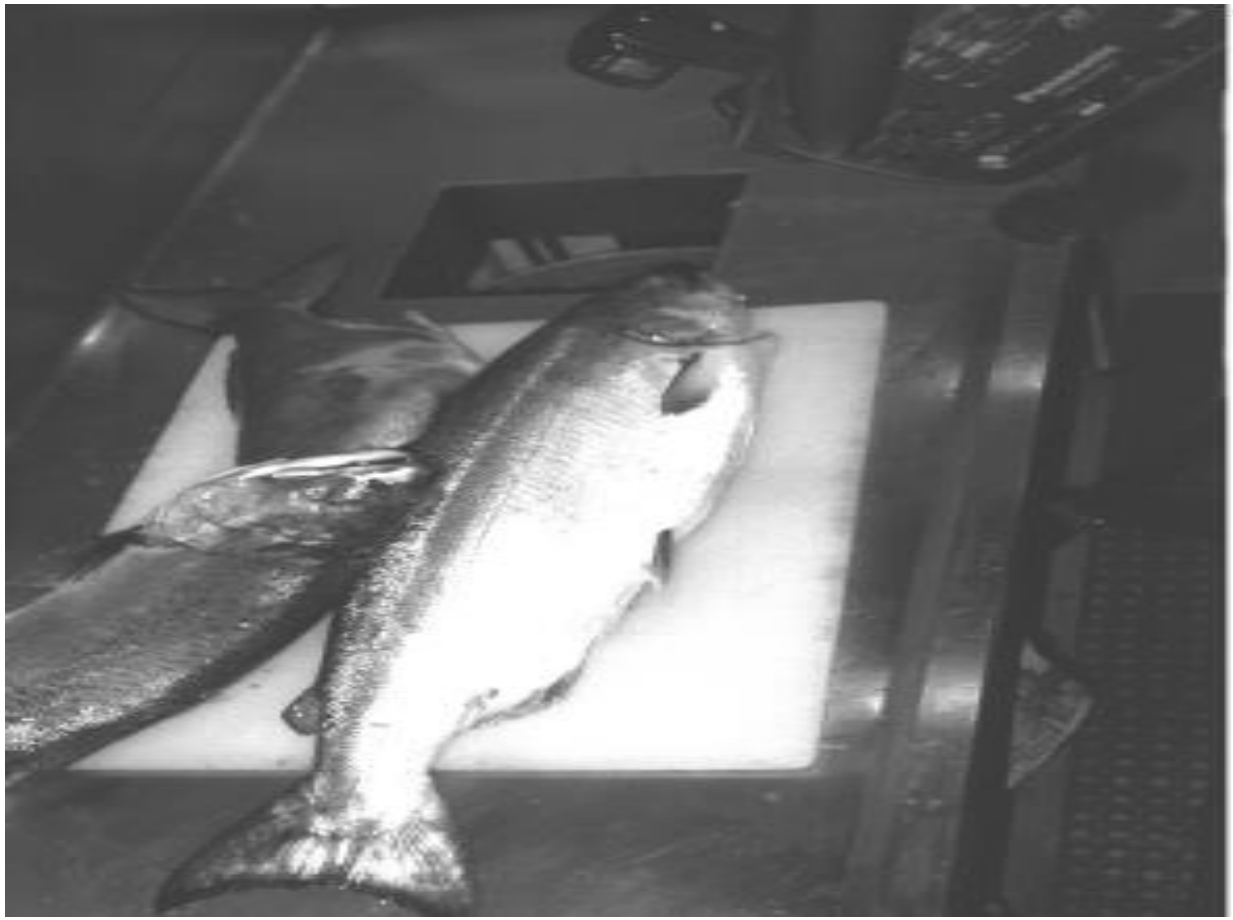
- Problem Analysis

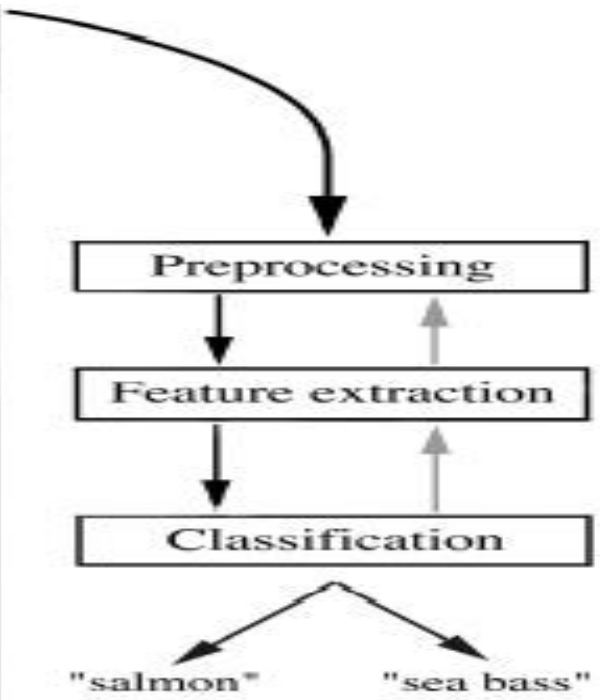
  – Set up a camera and take some sample images to extract features

    - Length
    - Lightness
    - Width
    - Number and shape of fins
    - Position of the mouth, etc…

- Preprocessing
  - isolate fishes from one another and from the background
- Feature Extraction
  - send isolated fish image to feature extractor
  - it reduces the data too
- Classification
  - pass the features to a classifier

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




Preprocessing
Feature extraction
Classification
"salmon"
"sea bass"

An Example: Decision Process

- What kind of information can distinguish one species from the other?
  - length, width, weight, number and shape of fins, tail shape, etc.
- What can cause problems during sensing?
  - lighting conditions, position of fish on the conveyor belt, camera noise, etc.
- What are the steps in the process?
  - capture image $\rightarrow$ isolate fish $\rightarrow$ take measurements $\rightarrow$ make decision

An Example: Selecting Features

- Assume a fisherman told us that a sea bass is generally longer than a salmon.
- We can use length as a *feature* and decide between sea bass and salmon according to a threshold on length.
- How can we choose this threshold?

An Example: Selecting Features

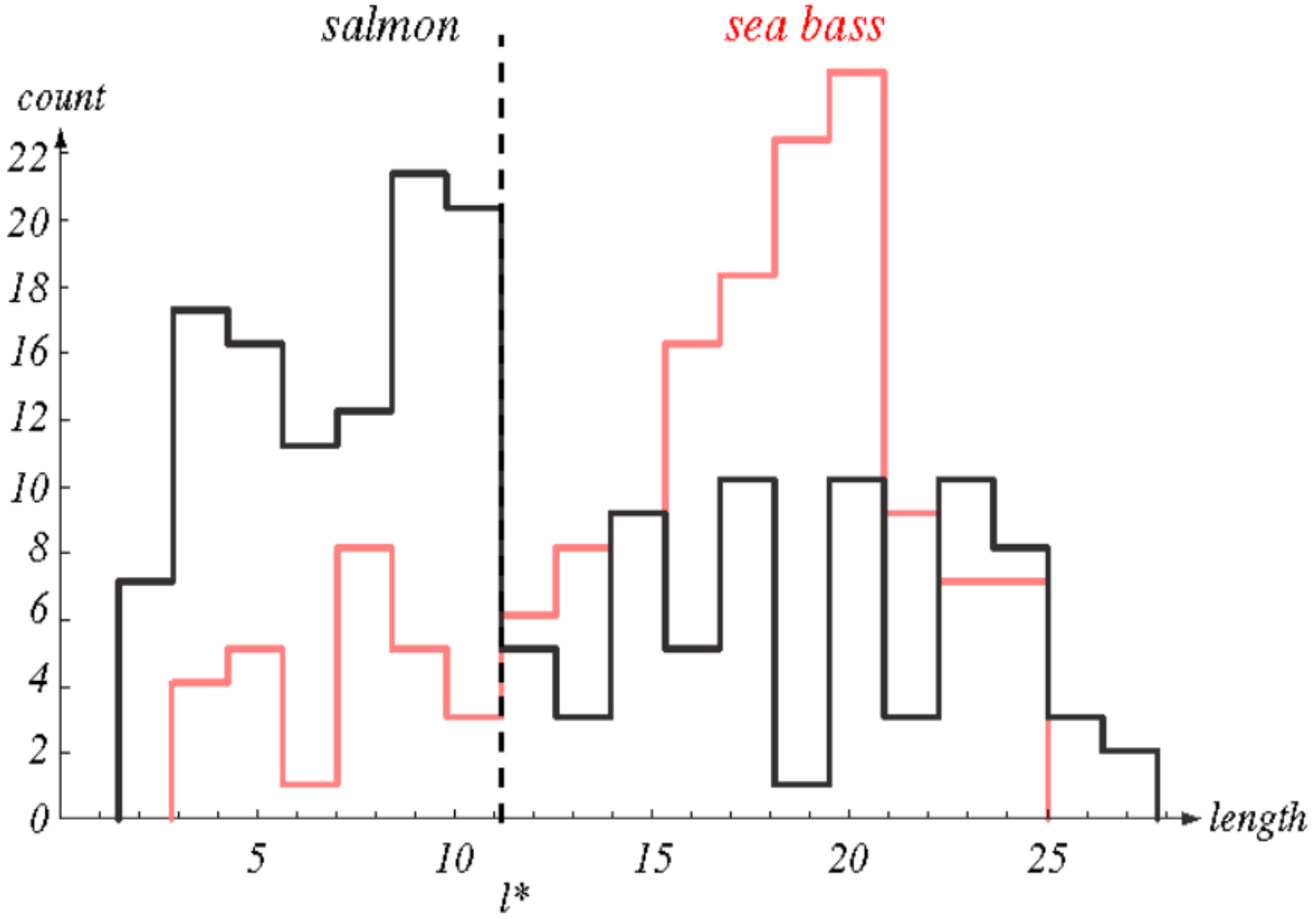


Figure 13: *Histograms* of the length feature for two types of fish in *training samples*. How can we choose the threshold l^* to make a reliable decision?

An Example: Selecting Features

- Even though sea bass is longer than salmon on the average, there are many examples of fish where this observation does not hold.
- Try another feature: average lightness of the fish scales.

An Example: Selecting Features

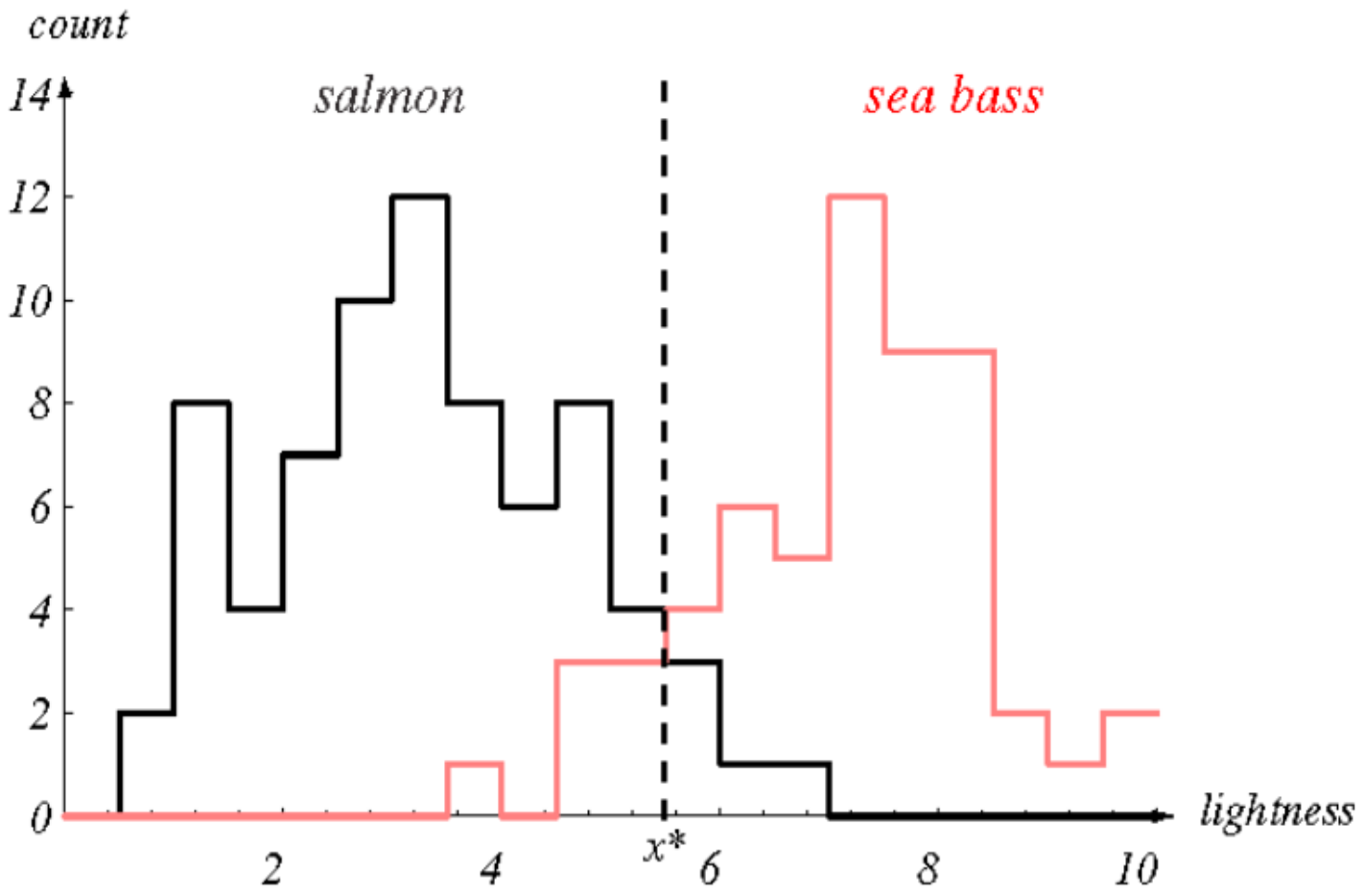


Figure 14: Histograms of the lightness feature for two types of fish in training samples. It looks easier to choose the threshold x^* but we still cannot make a perfect decision.

An Example: Cost of Error

- We should also consider *costs of different errors* we make in our decisions.
- For example, if the fish packing company knows that:
  - Customers who buy salmon will object vigorously if they see sea bass in their cans.
  - Customers who buy sea bass will not be unhappy if they occasionally see some expensive salmon in their cans.
- How does this knowledge affect our decision?

An Example: Multiple Features

- Assume we also observed that sea bass are typically wider than salmon.
- 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*)

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

in a two-dimensional *feature space*.

An Example: Multiple Features

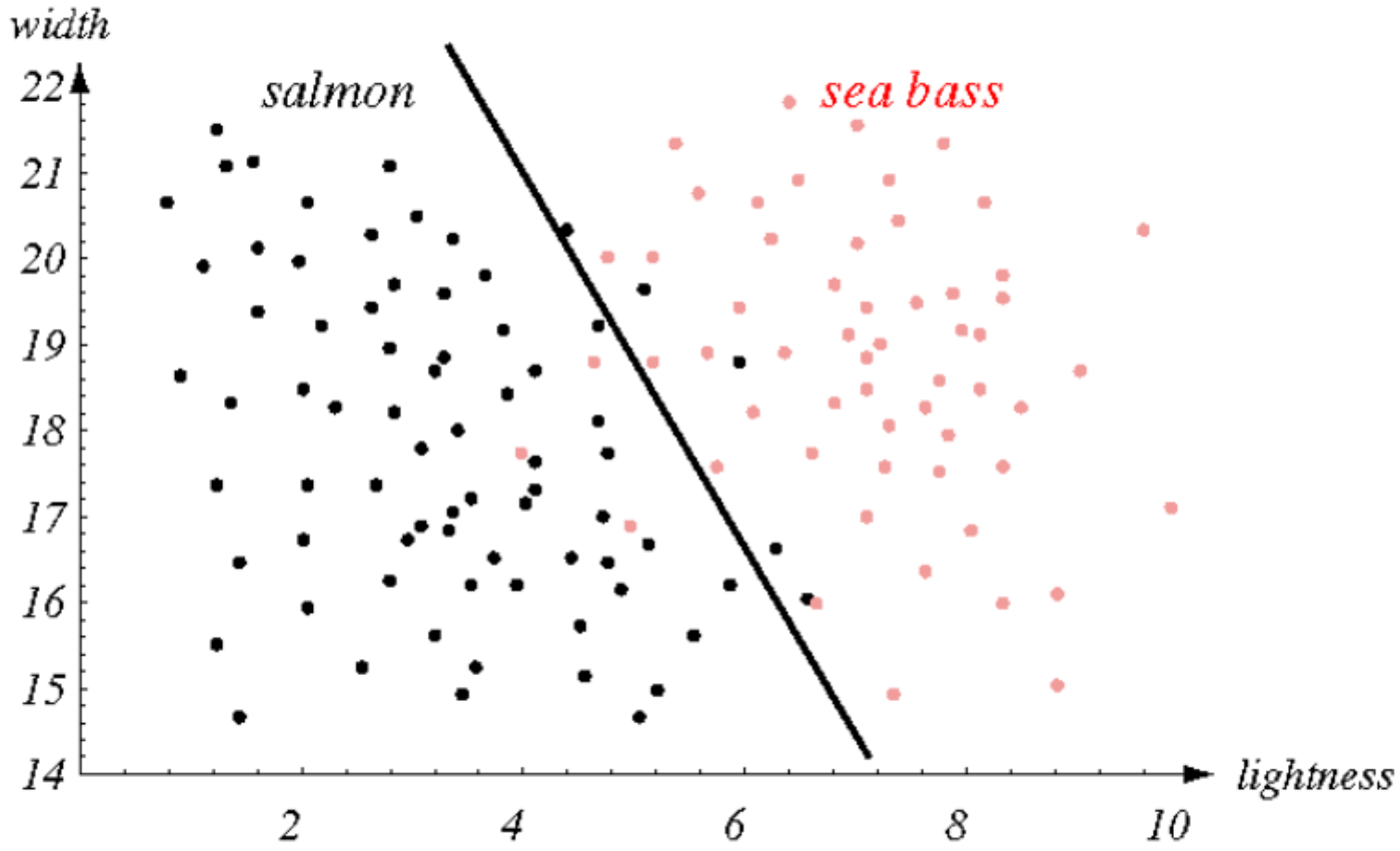


Figure 15: *Scatter plot* of lightness and width features for training samples. We can draw a *decision boundary* to divide the feature space into two regions. Does it look better than using only lightness?

An Example: Multiple Features

- Does adding more features always improve the results?
  - Avoid 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: Decision Boundaries

- Can we do better with another decision rule?
- More complex models result in more complex boundaries.

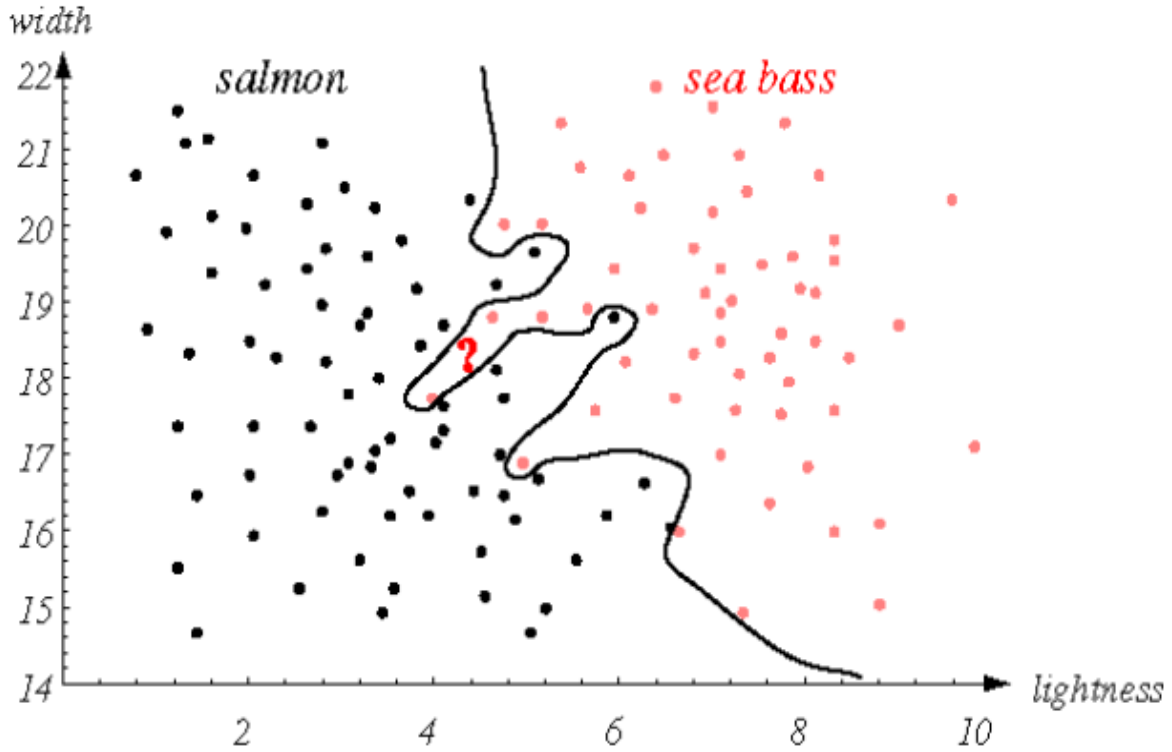


Figure 16: We may distinguish training samples perfectly but how can we predict how well we can *generalize* to unknown samples?

An Example: Decision Boundaries

- How can we manage the *tradeoff* between complexity of decision rules and their performance to unknown samples?

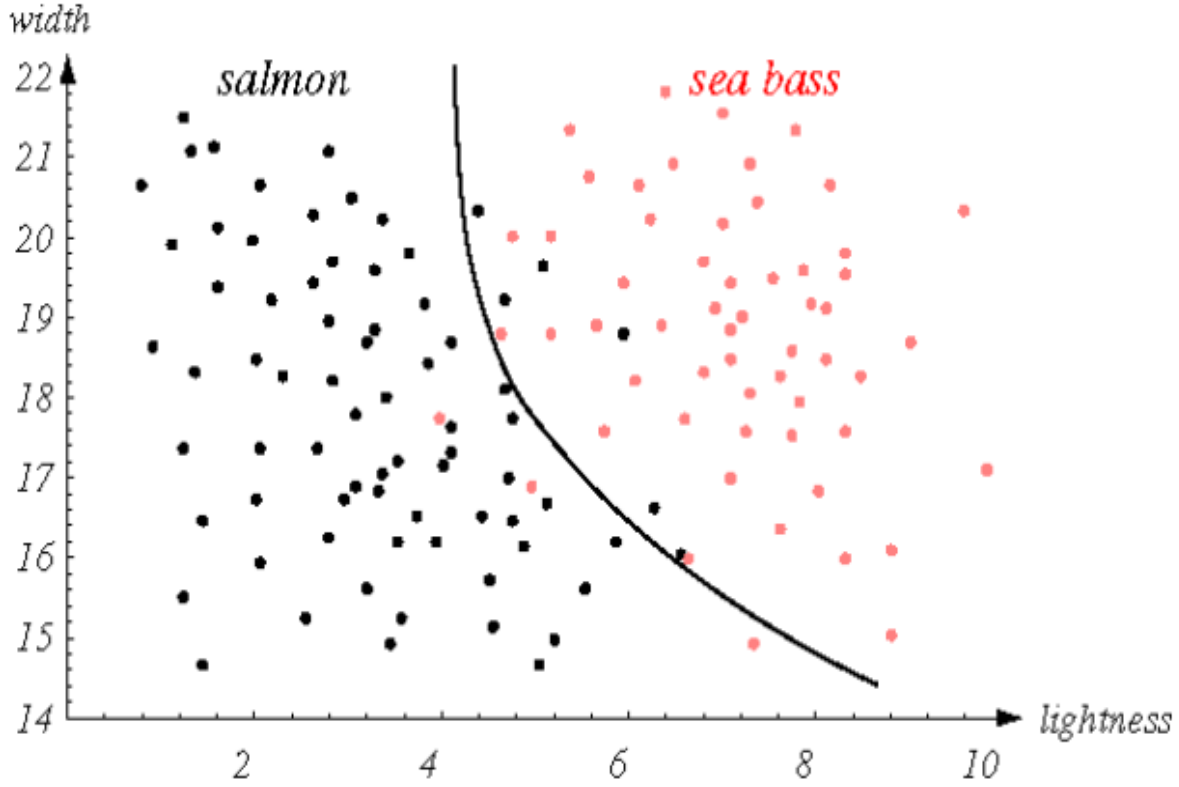


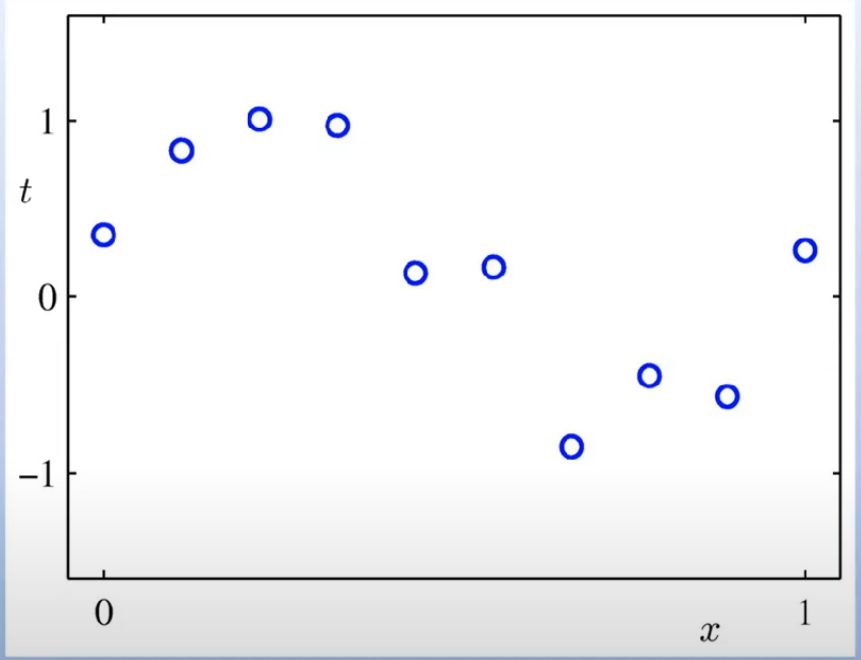
Figure 17: Different criteria lead to different decision boundaries.

- adding correlated feature does not improve anything and is thus redundant

- too many features may lead to *curse of dimensionality*

Overfitting

Overfitting vs Generalization

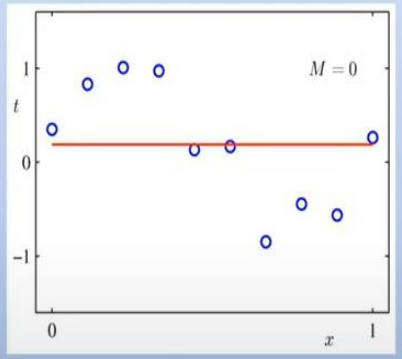


Of Which order polynomial will be best for the data?

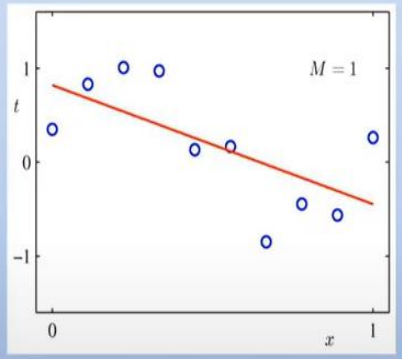
Overfitting

Overfitting vs Generalization

- Of Which order polynomial will be best for the data?
  - The model which has the least error as much as possible



0th order polynomial regression

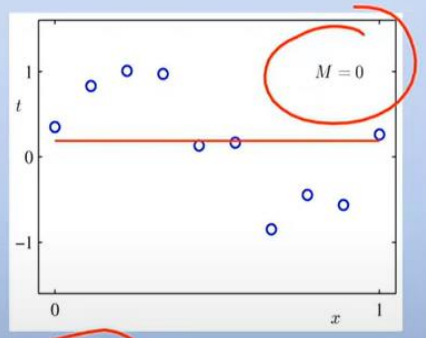


1st order polynomial regression

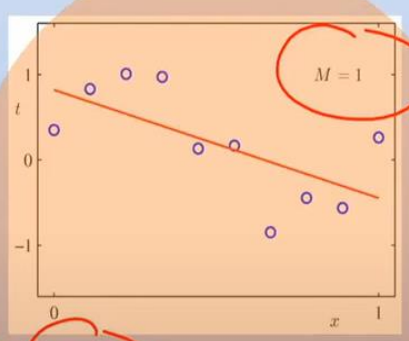
Overfitting

Overfitting vs Generalization

- Of Which order polynomial will be best for the data?
  - The model which has the least error as much as possible



0th order polynomial regression



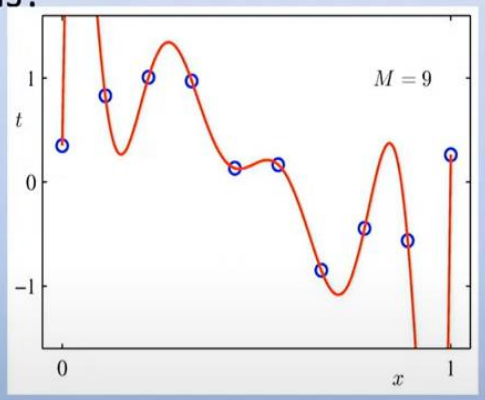
1st order polynomial regression

This is better because it has less error

Overfitting

Overfitting vs Generalization

- Of Which order polynomial will be best for the data?
  - What about this?



9th order polynomial regression

- This may be the BEST because the error is ZERO!!

Overfitting

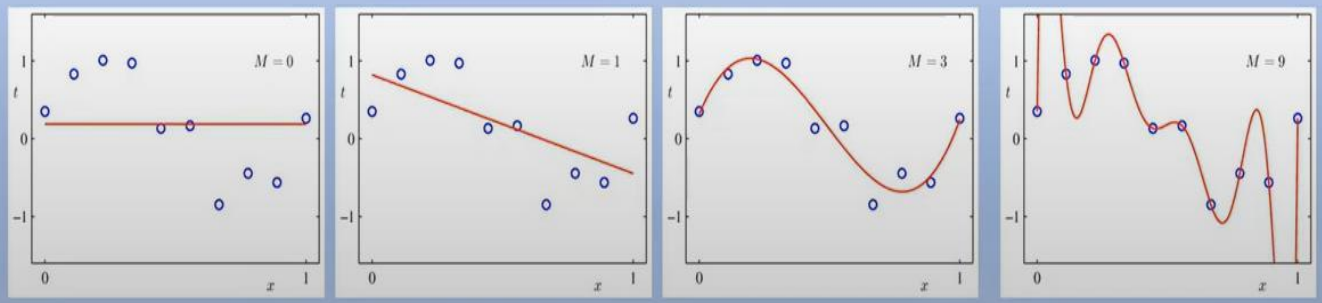
Overfitting vs Generalization

- What is the purpose of Machine Learning?

Learning the given data as exactly as possible

vs

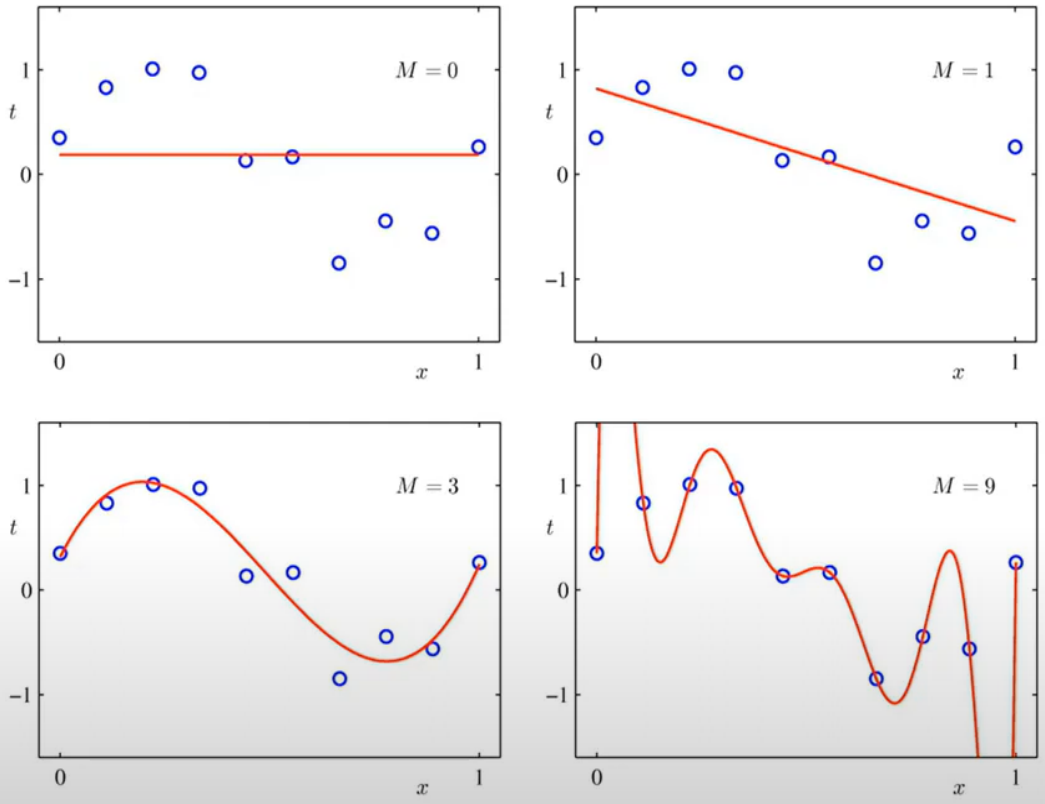
Predict the unknown data as exactly as possible based on the given data



Overfitting

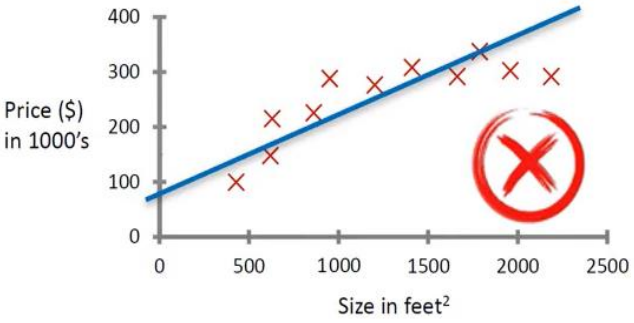
Overfitting vs Generalization

- Then, which one looks best?
  - As the order (M) increases,
    - the complexity of model increases
  - As the complexity of model increases,
    - the model can more exactly learn the given data
    - However, the prediction accuracy does not necessarily increase



The problem of overfitting - regression

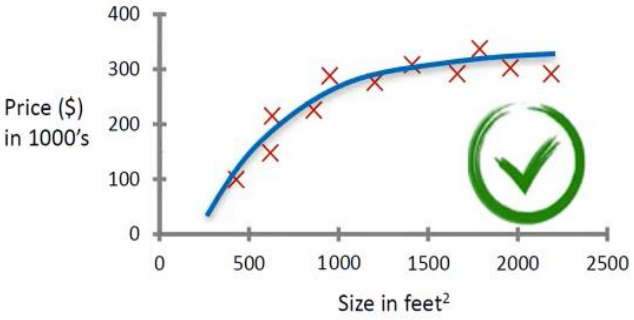
- **Underfitting**
- The model has a high **bias**
- The model makes a strong assumption that the housing prices will vary linearly with their size, but ends up not fitting the training data well (poor fit).



Simple model

$$h_\theta(x) = \theta_0 + \theta_1 x$$

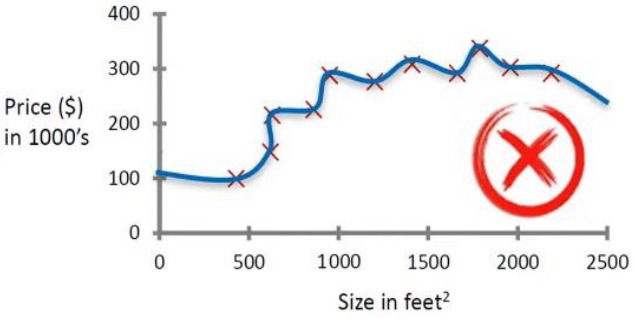
- Just right
- It fits the data pretty well



More complex model

$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

- **Overfitting**
- The model has a high **variance**
  - The space of possible hypothesis functions of this order is too large (too variable), and we do not have enough data to construct a good hypothesis of this type.
- Seems to fit the training data perfectly, but will have very poor performance on new data. It has 0 error on the training data, but it does not generalize well.



Much more complex model

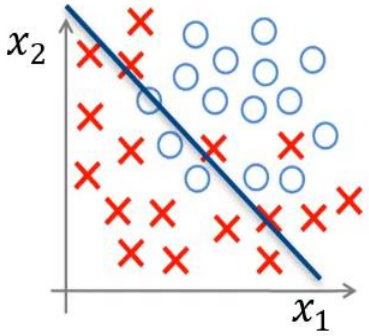
$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5 + \theta_6 x^6 + \theta_7 x^7 + \theta_8 x^8 + \theta_9 x^9 + \theta_{10} x^{10}$$

The problem of overfitting - classification

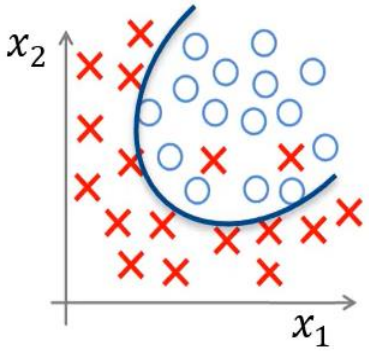
Example: Classification
(with Logistic Regression)

NOTE: g here is the sigmoid function.

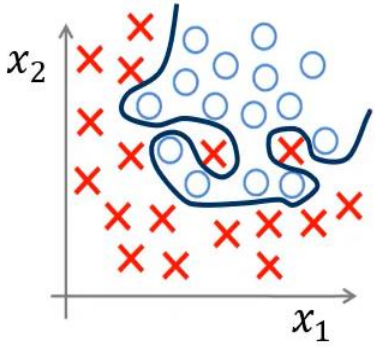
Which of these models do you think is a good model **for this data**?



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

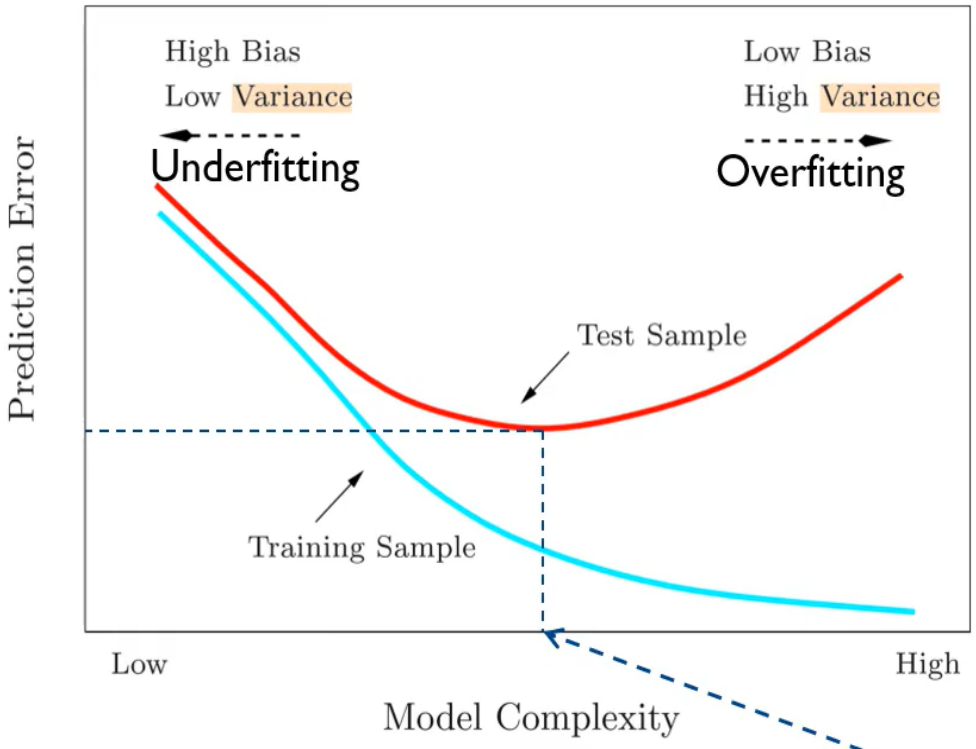


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5\, x_1 x_2)$$



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4\, x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \cdots)$$

The problem of overfitting



- What makes it more likely to overfit?
  - ❖ Not enough training examples (small training dataset)
  - ❖ Too many features
  - ❖ Using a non-convenient type of models / hypothesis functions (e.g. too much complex for our problem / data).

A good model complexity

Addressing Overfitting

1. **Hyperparameters tuning**
   - You can try various models of different complexity (e.g. with various hyperparameters values), compute the generalization error for each of them (as explained previously), and keep the best hyperparameters.

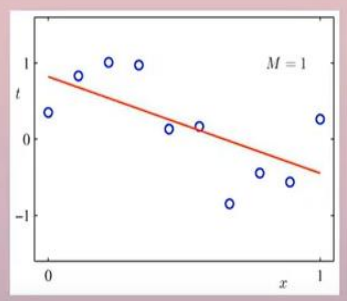
2. **Reducing the number of features**
   - We are more likely to overfit when the number of features is high (relatively to the size of the dataset).

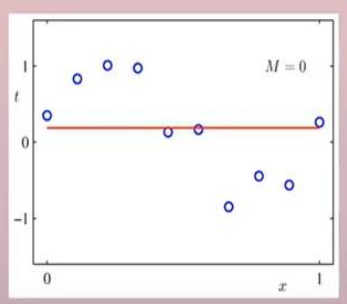
3. **Using an ensemble methods**
   - Such as *Random Forest* which addresses overfitting in *Decision Tree*.

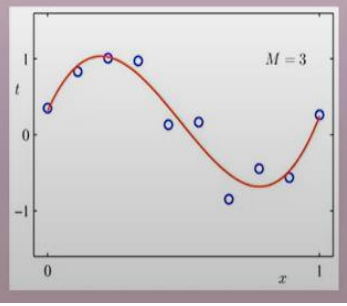
4. **Using regularization**
   - Keep all features, but reduce the magnitude (value) of parameters θ_j
   - Works well when we have a lot of features, and each feature contributes a bit to predicting the output.

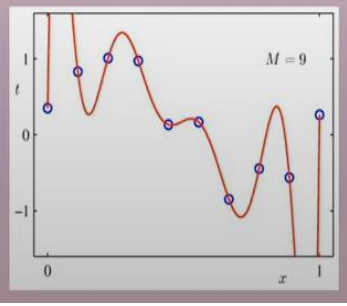
Model Evaluation

Model Evaluation





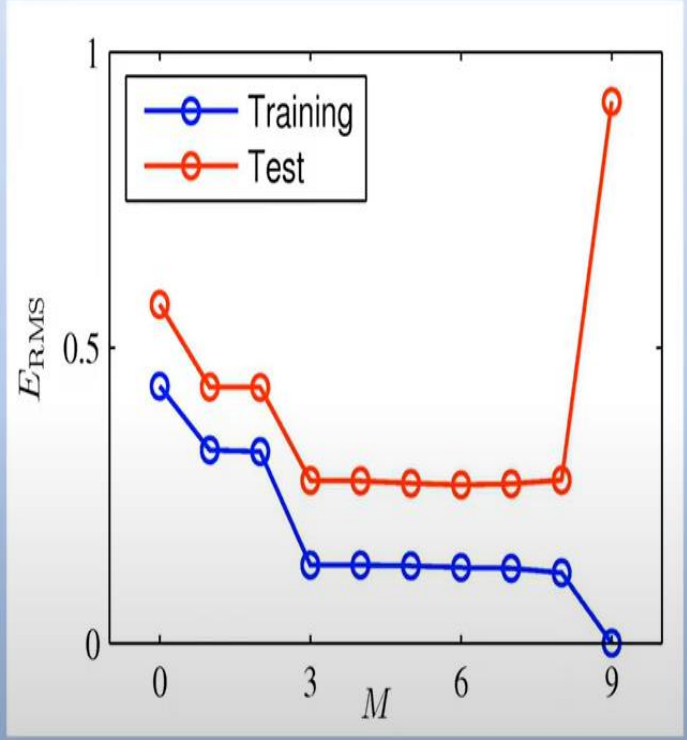




- Which model is best?
  - You may try several approaches and need to choose one
  - You may try several different parameters of a model and need to choose one
- Model Evaluations
  - Based on Training & Testing data set
  - Cross-Validation

Training Set and Test Set

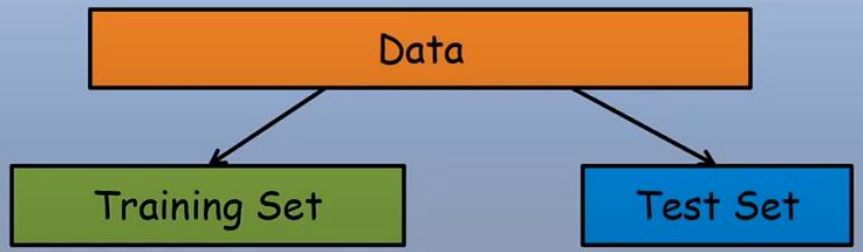
- Performance Graph



Model Evaluation

Training Set and Test Set

- How to choose a good model
  - Divide the given data into TRAINING set and TEST set
    - Training set and Test set should NOT overlap each other!!
    - Both need to be independent as much as possible
  - With Training set, build various models
  - With Test set, evaluate each model
  - Choose the model which shows the best performance with Test set

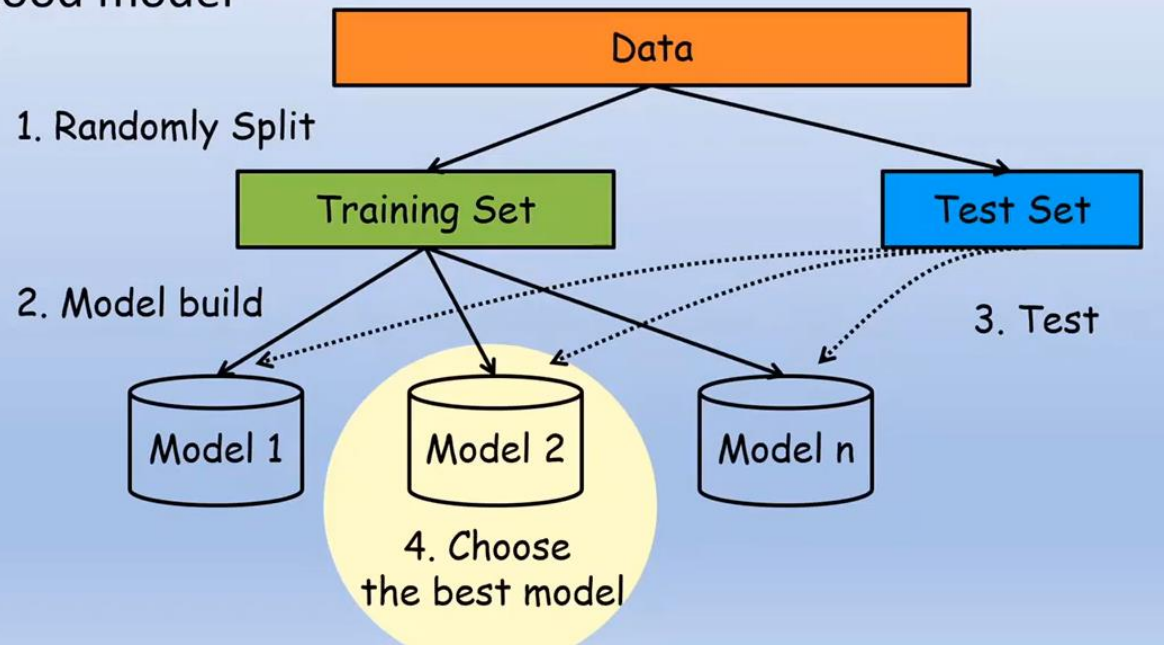


8

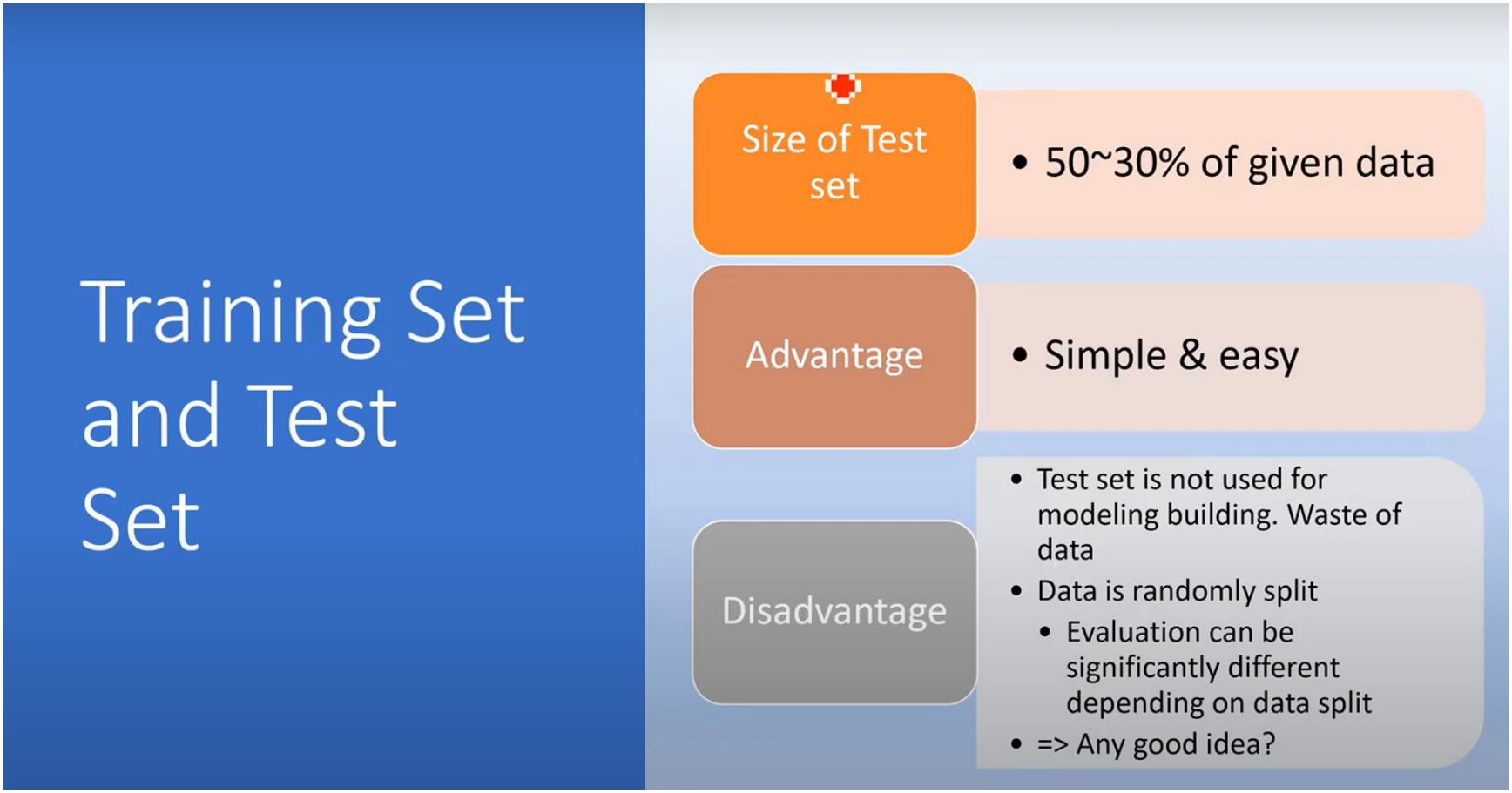
Model Evaluation

Training Set and Test Set

- How to choose a good model



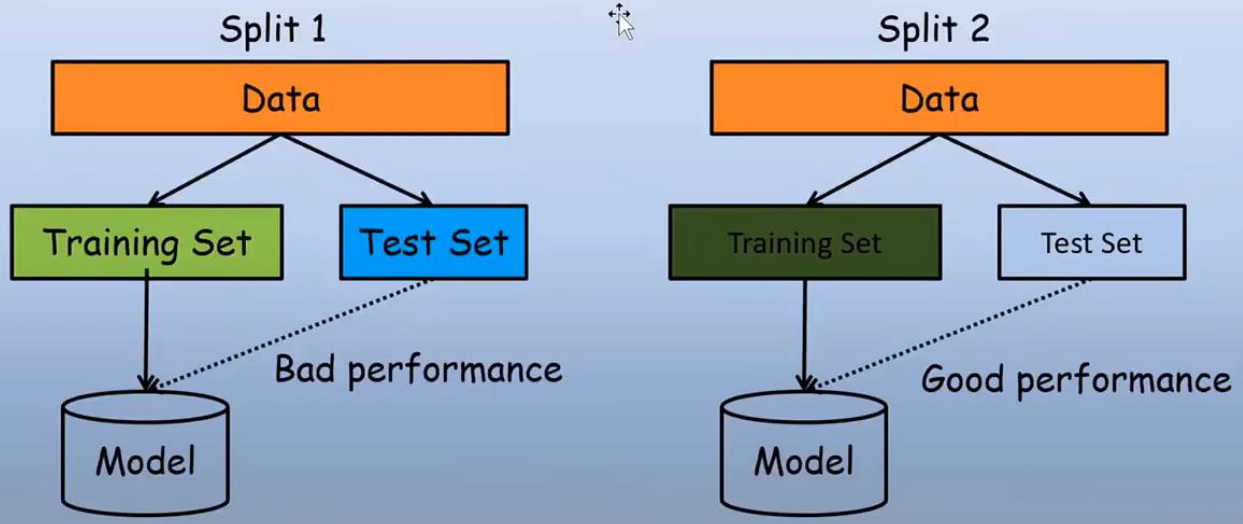
Model Evaluation



Model Evaluation

Training Set and Test Set

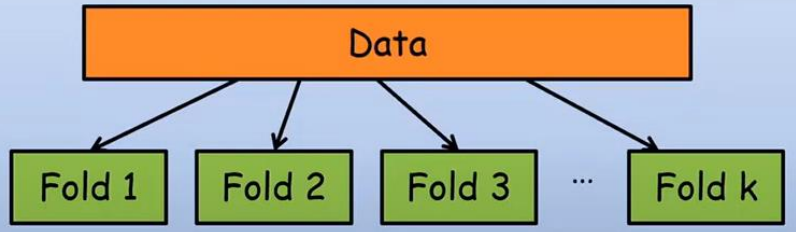
- Data Waste & Random Split



Cross Validation

Cross Validation

- In order to reduce statistical variance
  - Usually, k-fold cross validation is widely used
- K-fold Cross Validation
  - Split given data into *K* folds
  - Folds should not overlap with each other

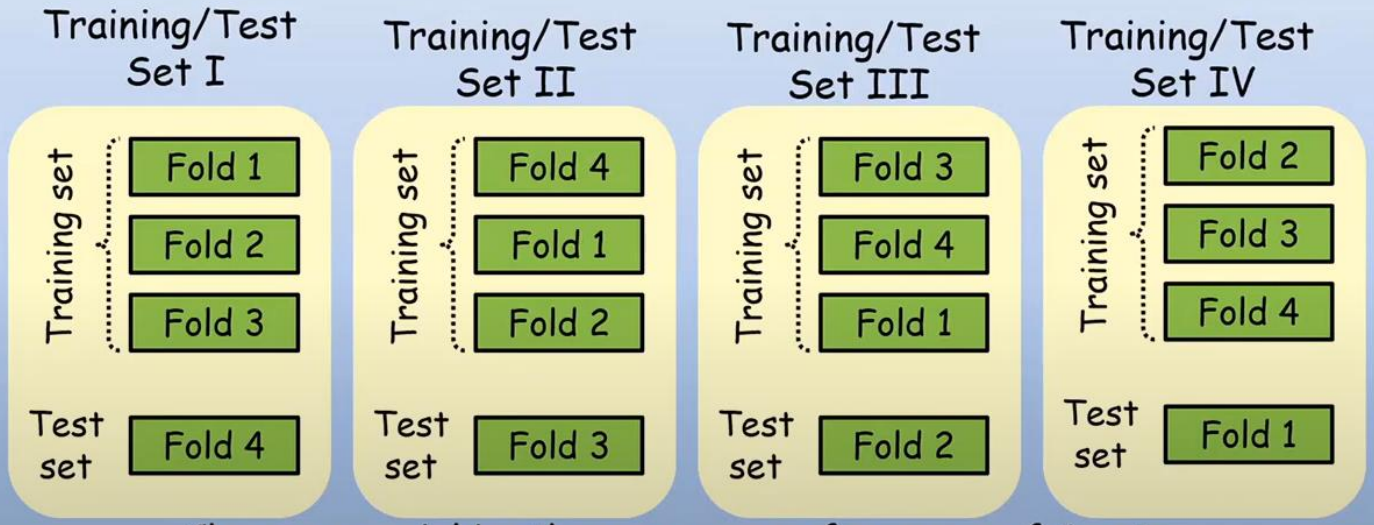


- Compose *k-1* training set and 1 test set with k folds

Model Evaluation

Cross Validation

- Example: 4-fold cross validation

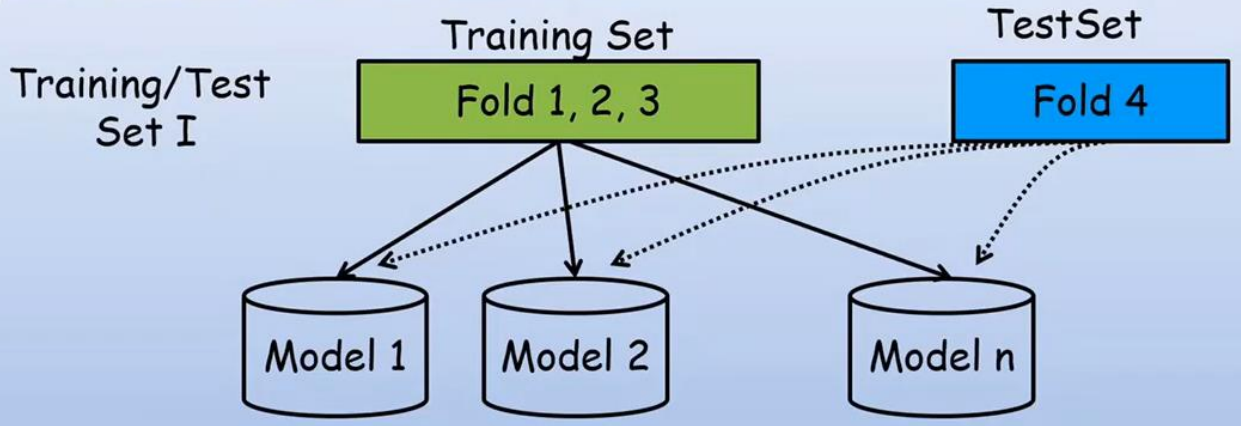


Choose a model by the average performance of 4 sets

Cross Validation

Cross Validation

- Example: 4-fold cross validation



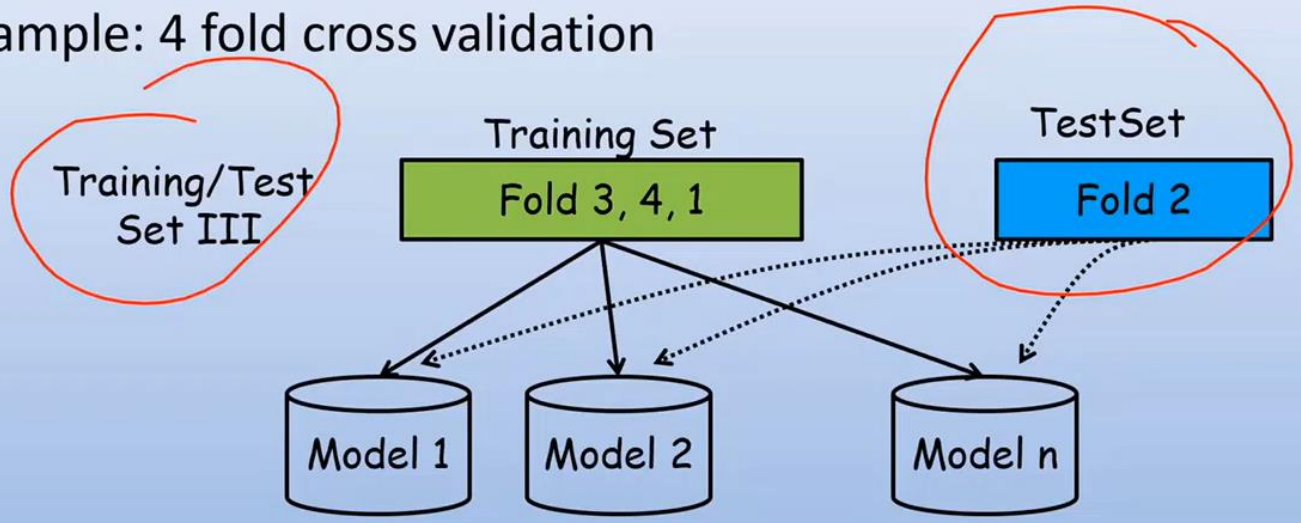
Set I	3	4	...	5
Set II			...	
Set III			...	
Set IV			...	

Performance

Cross Validation

Cross Validation

- Example: 4 fold cross validation



Set I	3	4	...	5
Set II	4	1	...	2
Set III	5	2	...	3
Set IV			...	

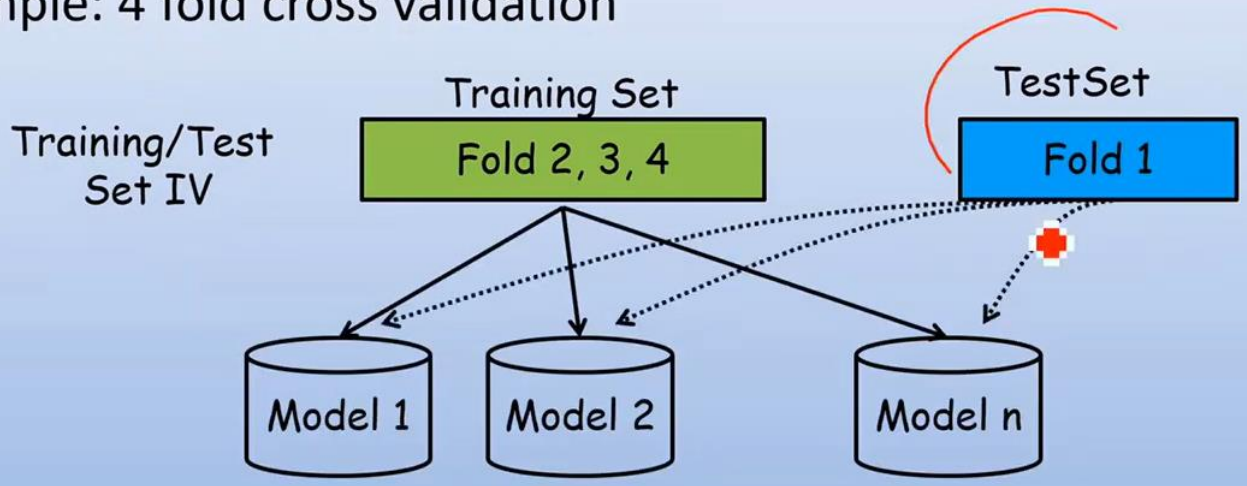
Performance

17

Cross Validation

Cross Validation

- Example: 4 fold cross validation

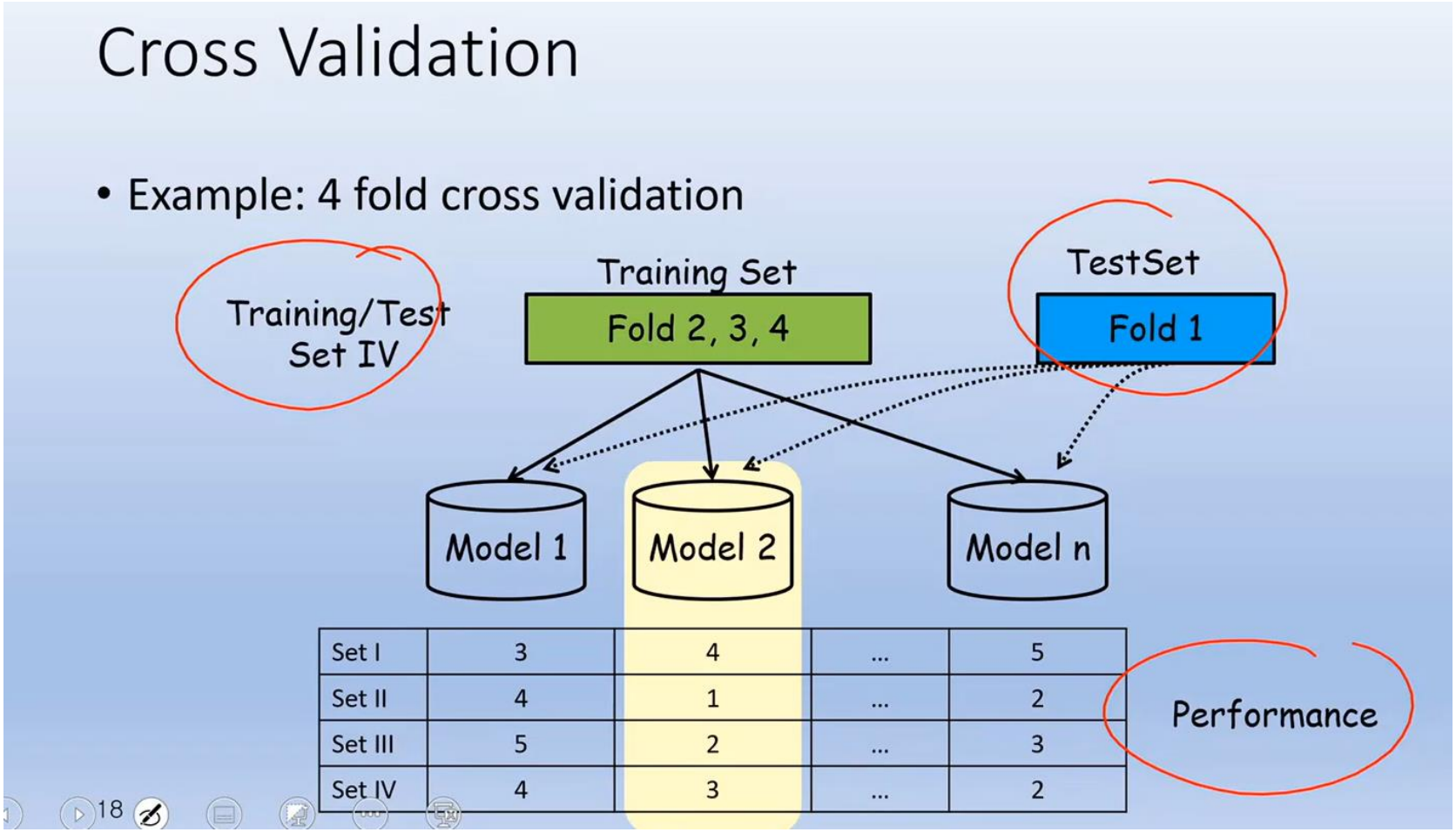


Set I	3	4	...	5
Set II	4	1	...	2
Set III	5	2	...	3
Set IV	4	3	...	2

Performance

18

Cross Validation



Cross Validation

Cross Validation

Summary

The data set is divided into *k* subsets, and the holdout method is repeated *k* times.

Each time, one of the *k* subsets is used as the test set and the other *k-1* subsets are put together to form a training set.

Then the average error across all *k* trials is computed.

The variance is reduced as *k* is increased.

Advantage

Less dependent on how the data gets divided.

Every data point gets to be in a test set exactly once, and gets to be in a training set k-1 times.

Disadvantage

Time !!

19

Pattern Recognition Systems

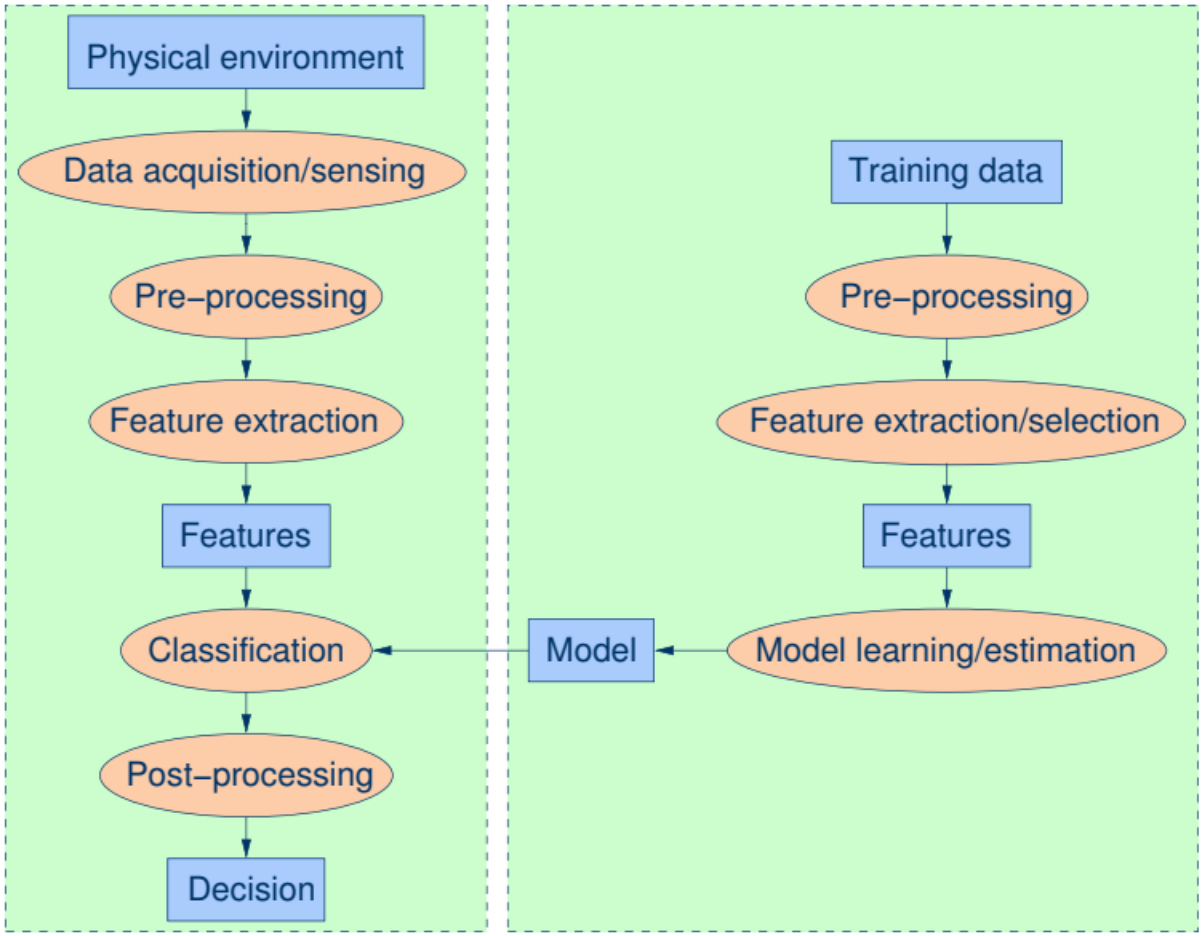


Figure 21: Object/process diagram of a pattern recognition system.

Pattern Recognition Systems

- Data acquisition and sensing:
  - 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.
- Feature extraction:
  - Finding a new representation in terms of features.

Pattern Recognition Systems

- Model learning and estimation:
  - Learning a mapping between features and pattern groups and categories.
- 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

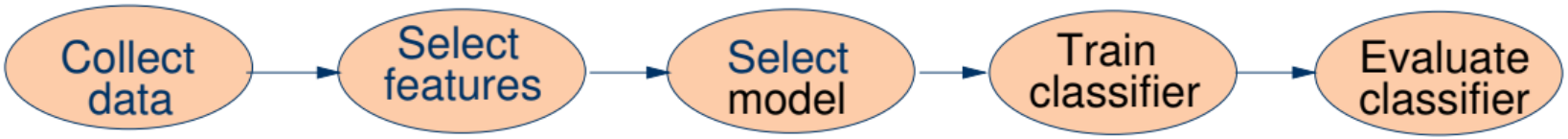


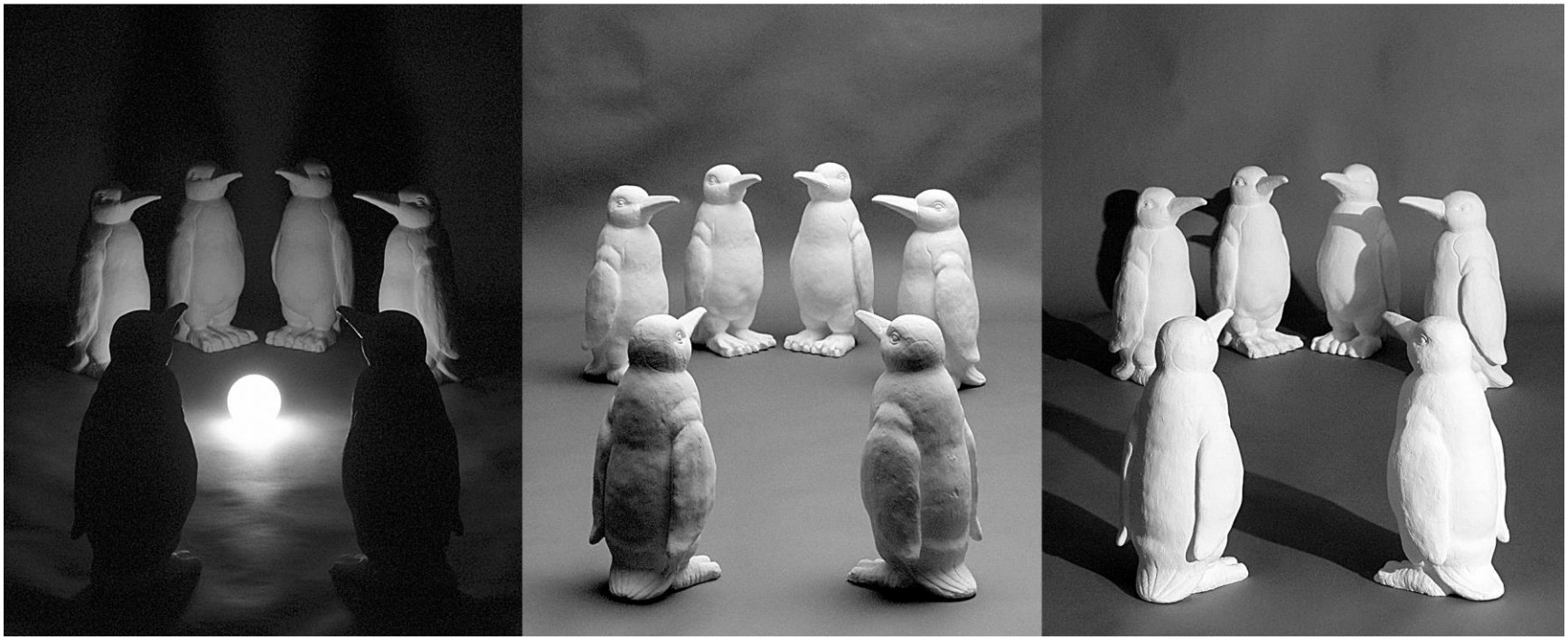
Figure 22: The design cycle.

- Data collection:
  - Collecting training and testing data.
  - How can we know when we have adequately large and representative set of samples?

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.

Challenges: illumination



Challenges: viewpoint variation

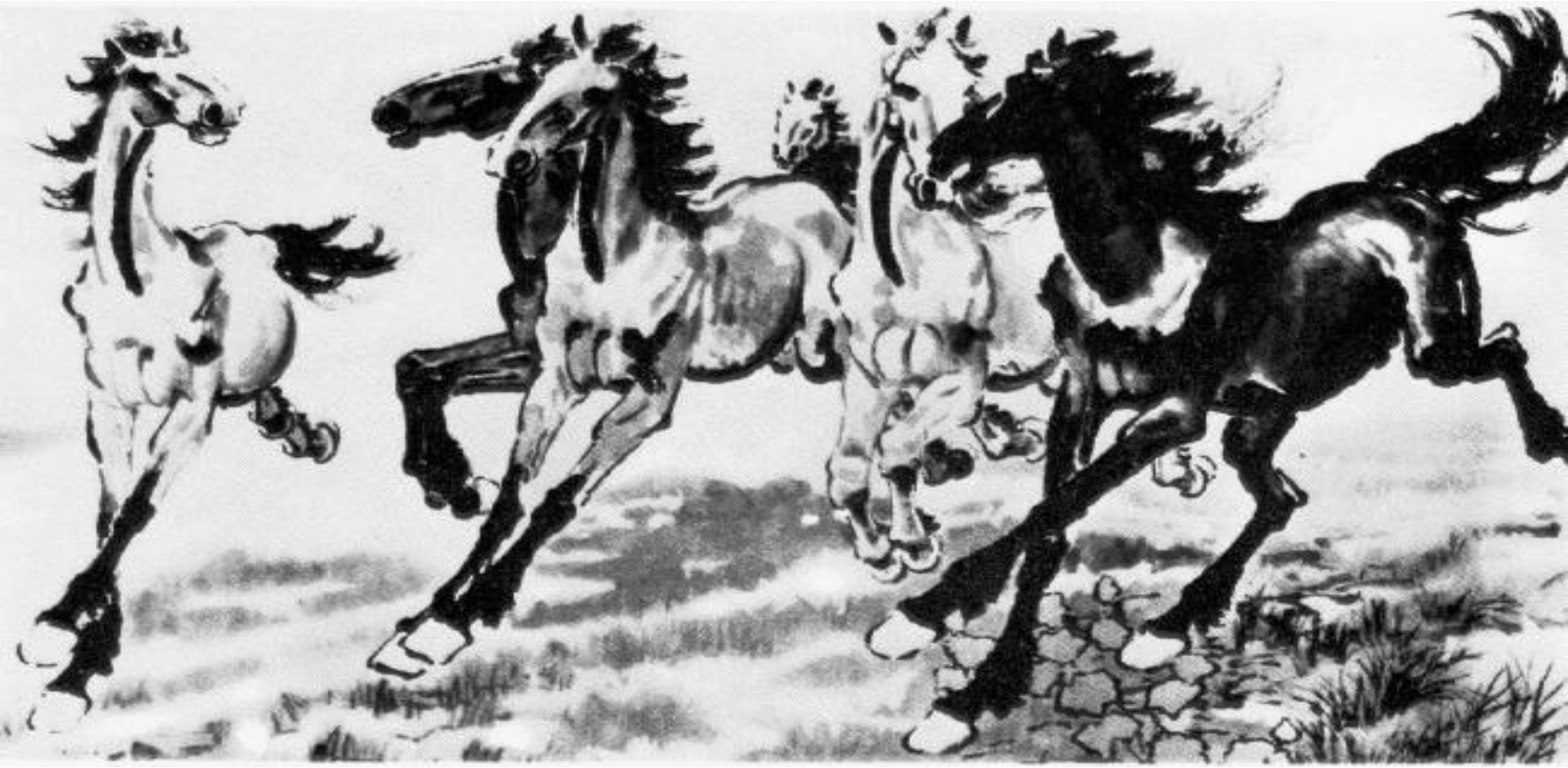
Michelangelo 1475-1564

slide credit: Fei-Fei, Fergus

Challenges: scale

slide credit: Fei-Fei, Fergus

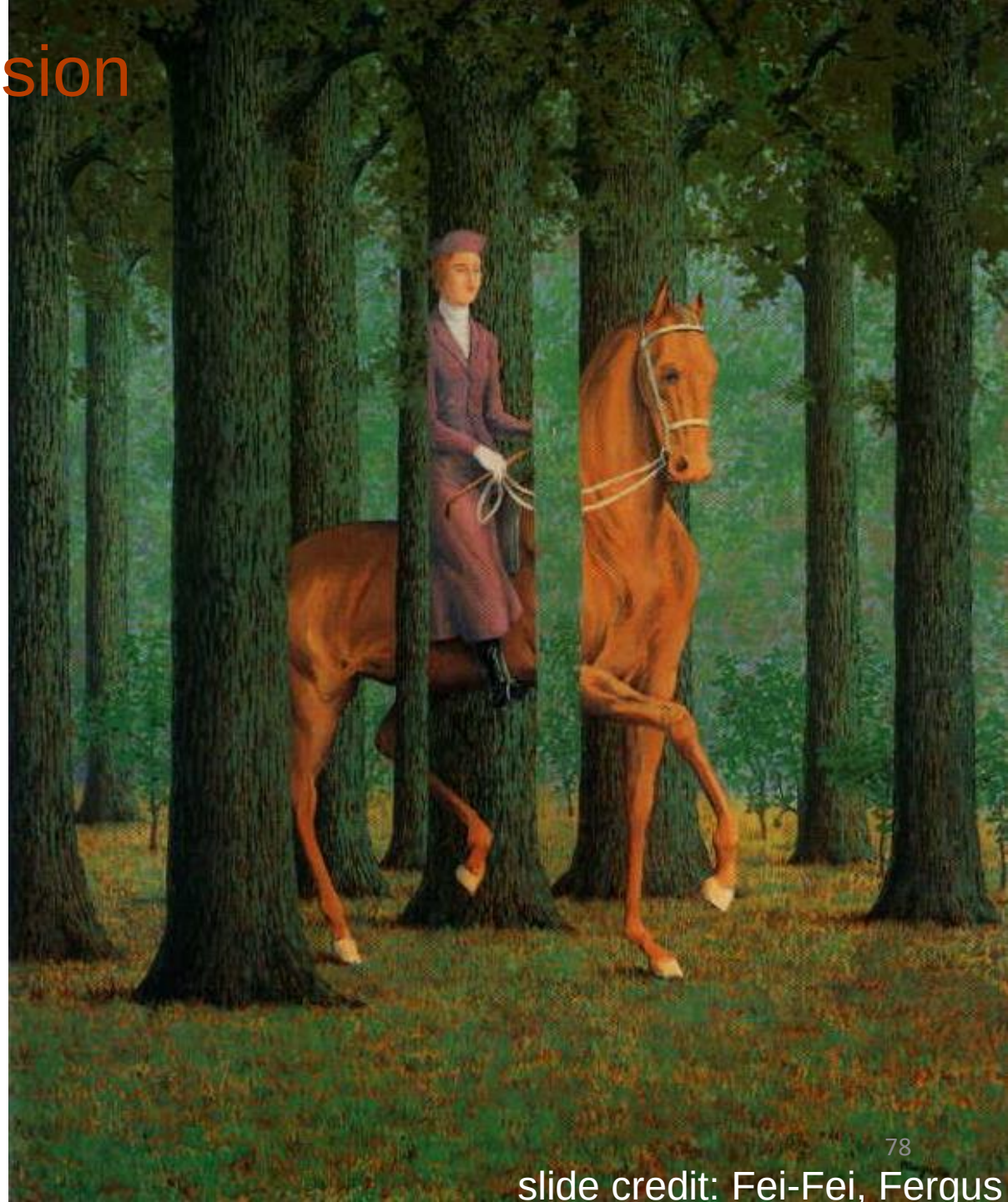
Challenges: deformation



Xu, Beihong 1943

slide credit: Fei-Fei, Fergus

Challenges: occlusion



Magritte, 1957

slide credit: Fei-Fei, Fergus

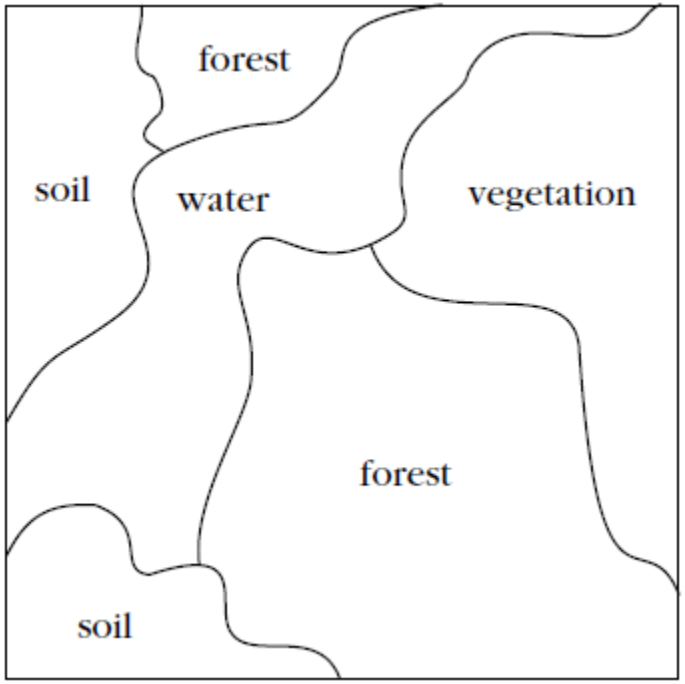
The Design Cycle

- 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?

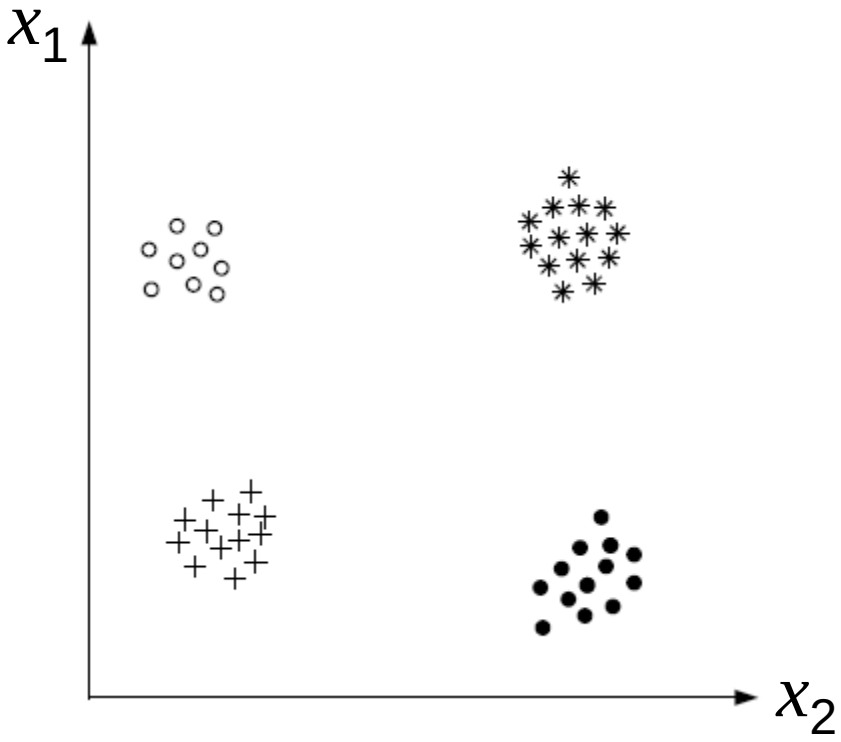
The Design Cycle

- 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.

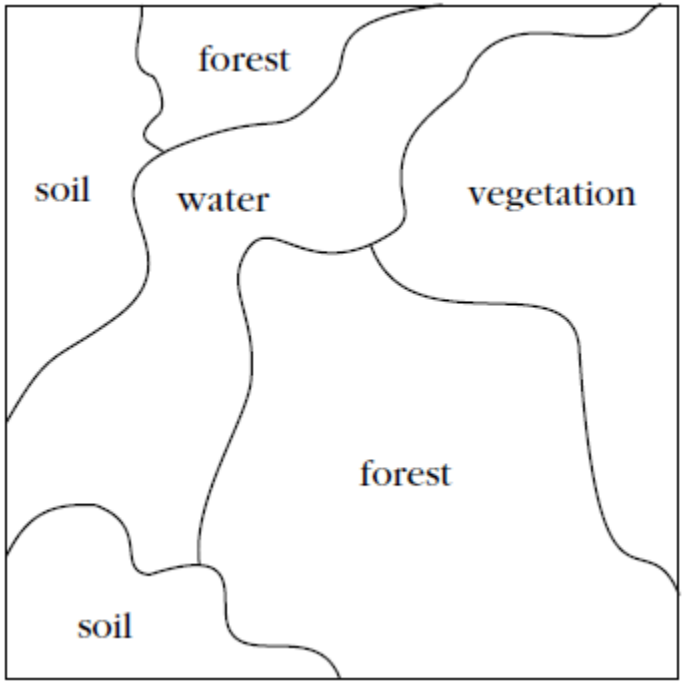
Unsupervised Learning

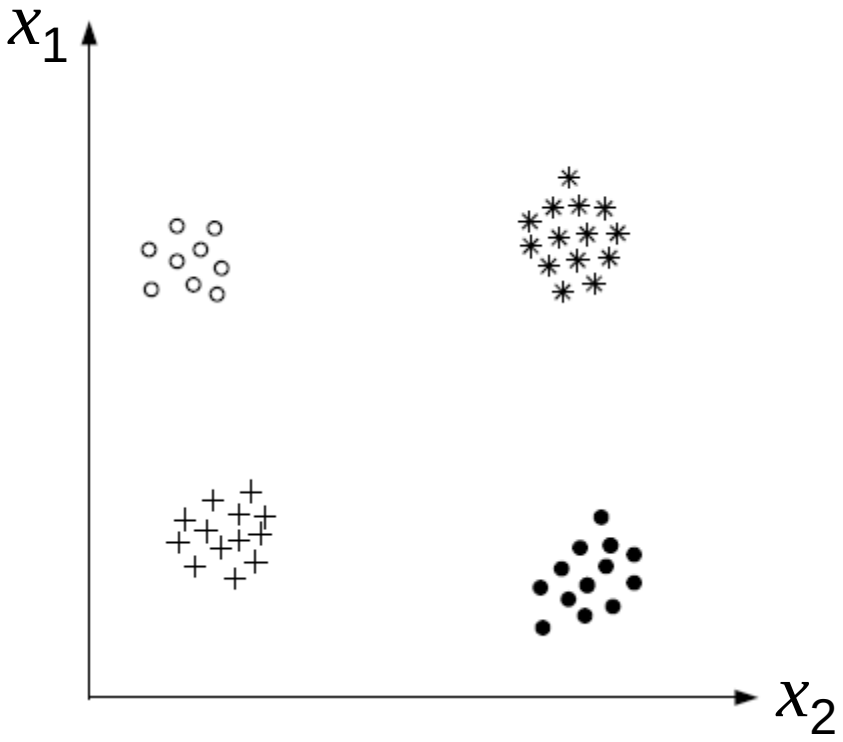


Unsupervised Learning

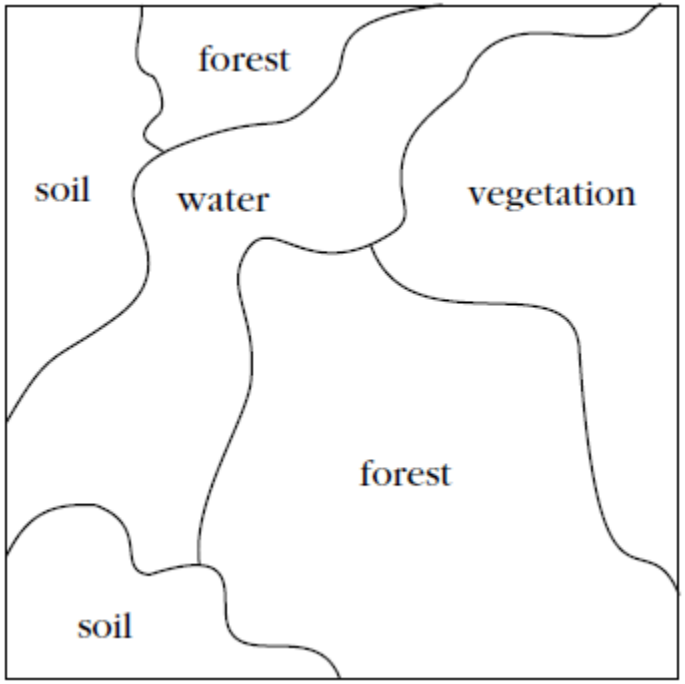


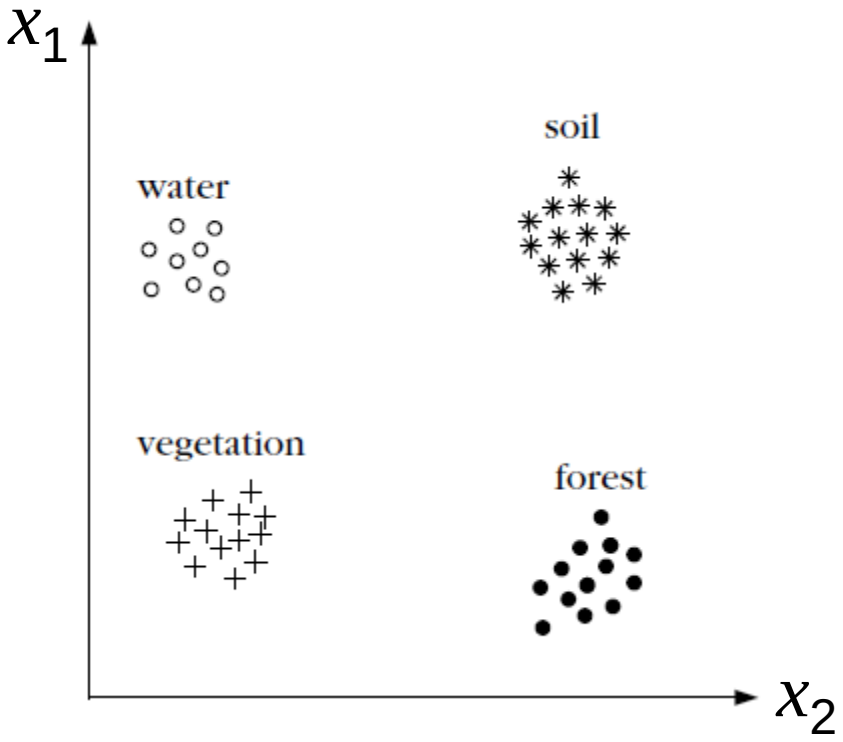
Unsupervised Learning





Unsupervised Learning





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.

Summary

- Pattern recognition techniques find applications in many areas: machine learning, statistics, mathematics, computer science, biology, etc.
- There are many sub-problems in the design process.
- Many of these problems can indeed be solved.
- More complex learning, searching and optimization algorithms are developed with advances in computer technology.
- There remain many fascinating unsolved problems.