Neural Networks and Learning Systems
TBMI26 / 732A55
2021

Lecture 1
Introduction

Magnus Borga
Department of Biomedical Engineering (IMT)
magnus.borga@liu.se

Course information

- All information will be available on Lisam
- Lectures will be published during the course

- You must register for classes and labs on Lisam
  - Chose group A or B for classes – and follow your group!
  - Chose group “Wednesday” or ”Thursday” for the labs (not connected to class)

Staff

- **Examiner:** Magnus Borga, IMT (magnus.borga@liu.se)
- **Course admin:** David Abramian, IMT (david.abramian@liu.se)

- Lectures:
  - Magnus Borga
  - Anders Eklund
- Lessons:
  - David Abramiam
  - Martin Hultman
- Laboratory exercises:
  - Martin Hultman
  - David Abramiam
  - Iulian Tampu
  - Marco Cirillo

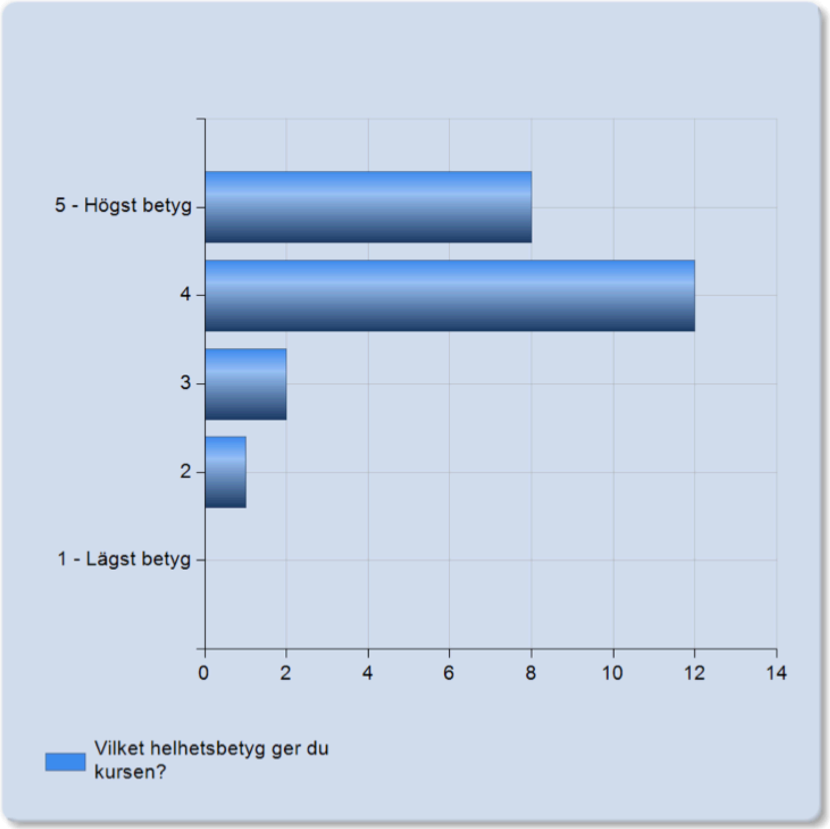
Course evaluation and development

9. Vilket helhetsbetyg ger du kursen?

Vilket helhetsbetyg ger du kursen?	Antal svar
5 - Högst betyg	8 (35%)
4	12 (52%)
3	2 (9%)
2	1 (4%)
1 - Lägst betyg	0 (0%)
Summa	23 (100%)

Average score 2020: 4.17

Course evaluation 2020

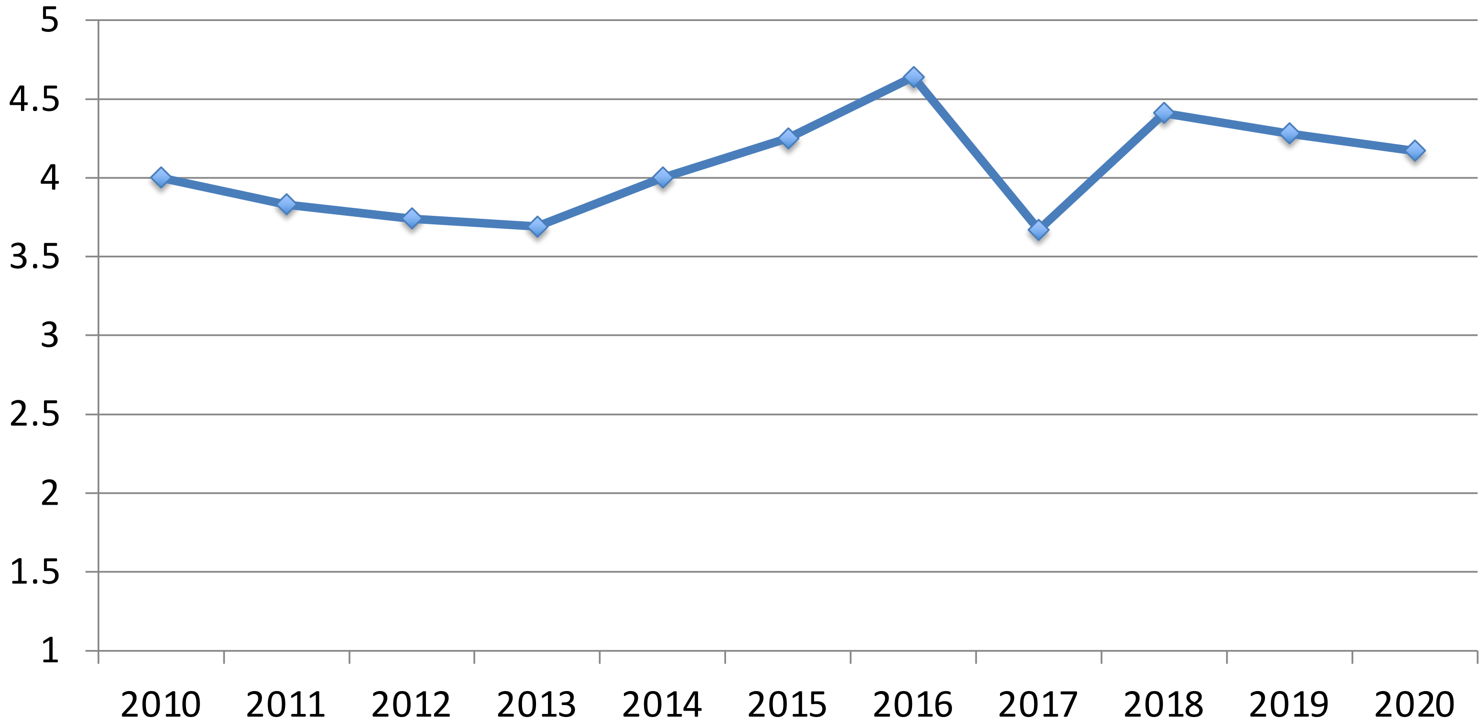


Course evaluation and development

- Main criticism 2020 was regarding the part on deep learning which was outsourced to another department.

- Changes 2021
  - All parts of the course are now given by the same team at IMT
  - New lectures, lesson and laboratory exercise on deep learning

Overall credit over time



The Course - Overview

- 9 lectures
- 9 lessons (classes)
- 4 assignments
- 1 written exam

Mandatory

- Course language is English.

The Course - Lectures

PPT lectures, handouts on course page

1. Introduction
2. Supervised learning - Linear classification
3. Neural networks
4. Deep learning
5. Ensemble learning & Boosting methods
6. Generative Adversarial Networks
7. Reinforcement learning
8. Unsupervised learning – Dimensionality reduction, Clustering
9. Kernel methods

Lectures in distance mode

- Lectures will be pre-recorded and published on Lisam
- Please email questions to the lecturer (me or Anders)
  - magnus.borga@liu.se
  - anders.eklund@liu.se
- Questions and answers will be published in the Lecture folder on Lisam

The Course - Lessons

- One lesson after each lecture
- Pen & paper exercises
- Complementary presentations
- Preparations and help with lab assignments
- Choose group (A or B) on Lisam and follow that group

The Course - Assignments

- 4 laboratory exercises/assignments:
    1. Pattern recognition using linear classifiers and neural networks
    2. Deep learning
    3. Face recognition in images using Boosting techniques
    4. Reinforcement learning
- Matlab and Python (Jupyter Notebook in lab 2)
- Assignments are done in pairs. (Not more than 2 students together!)
- Supervision time scheduled (“LA” in schedule) –
    - Choose lab group (Wednesday or Thursday) on Lisam and follow that group!

- Deadlines for <u>written reports</u>.
- Late reports may not be corrected until next re-exam (June/August).

Course literature

- Lecture notes
- Exercise collection
- Assignments

- Additional links in lecture notes (not required reading)

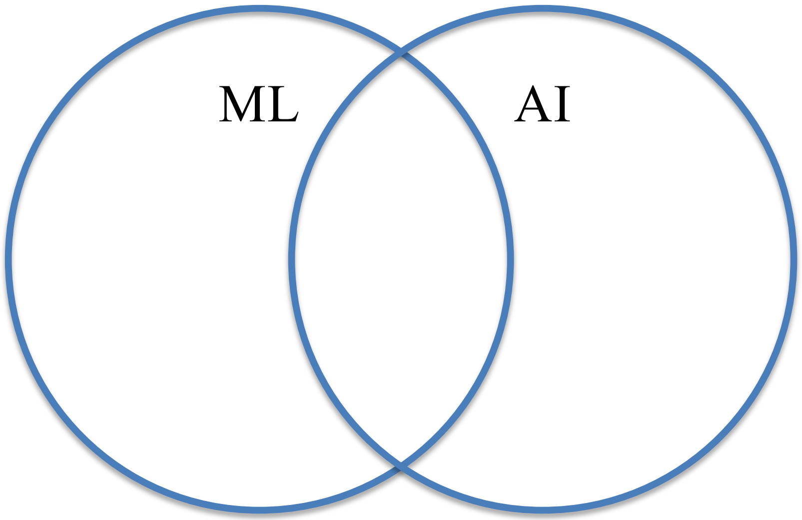
Challenges this year

- Distance mode!
- New lab-assignment on deep learning.

What is machine learning?

The ability of a system to learn from data and generalize to new data

ML vs AI



- AI aims at simulating "intelligent" behaviour, but not necessarily by learning
- ML does not always aim at doing something "intelligent"

How can a machine learn?

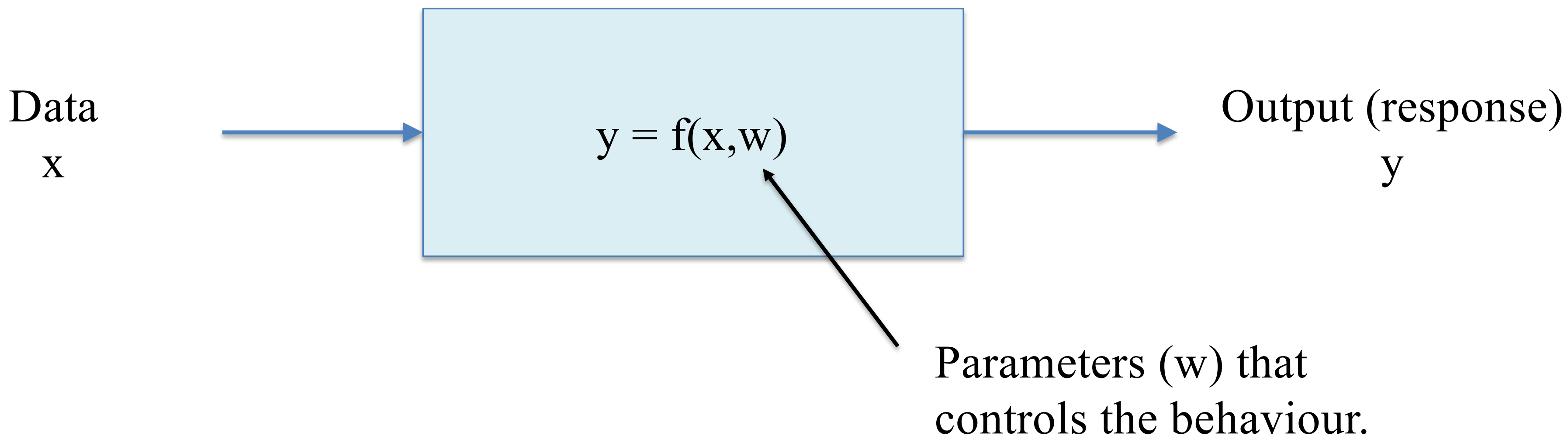
Learning: ”Any relatively permanent change in behaviour resulting from past experience.”

(Encyclopaedia Britannica 1964)

- The ”behaviour” of the machine is determined by its parameters.
- “past experience” is previously observed data.

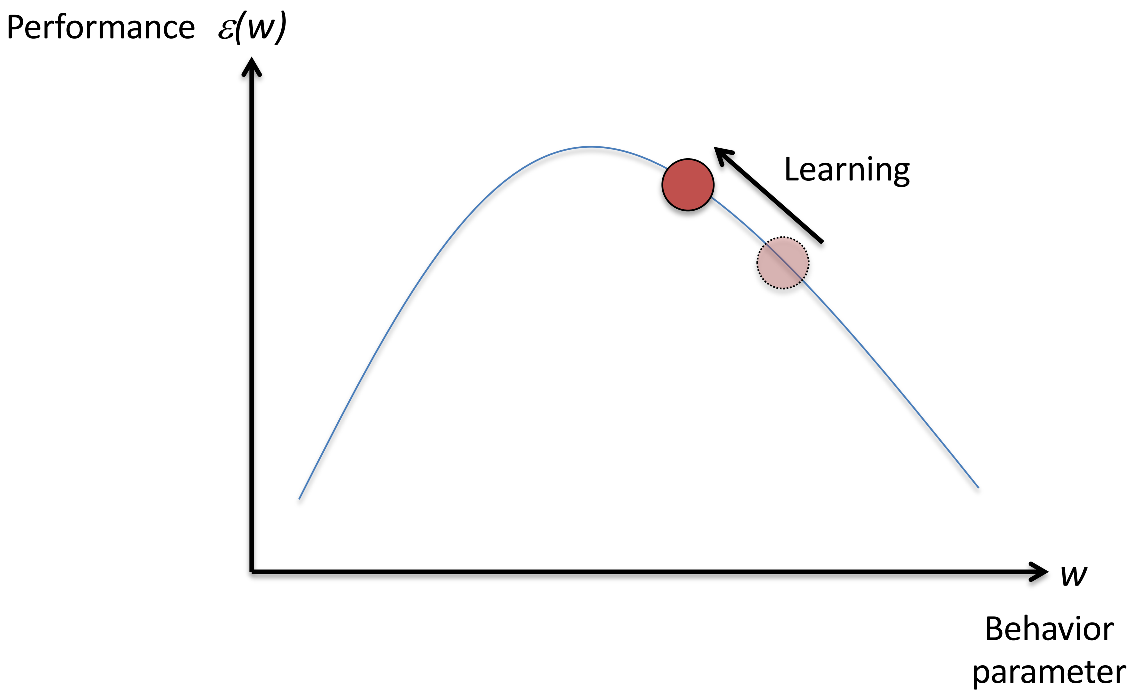
- Machine learning = changing parameter values (behaviour) as a result of observed data (experience).

Machine Learning



Machine learning: The system changes its parameters (w) in order to improve its performance

Learning = optimization of performance



What is being learnt depends on how we define performance!

Mathematical foundations of machine learning

Machine Learning

Find parameters for optimal behaviour

Optimization theory

Statistics

Model uncertainty

Multidimensional calculus

Linear algebra

Prerequisites

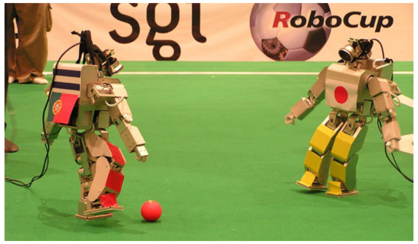
- Linear algebra
  - Vectors, scalar products, eigenvectors and eigenvalues
- Multidimensional calculus
  - Gradients, partial derivatives
- Mathematical statistics
  - Mean, variance, covariance, correlation, Gaussian distribution,
- Programming
  - Some programming experience
  - Matlab and Python experience helps a lot

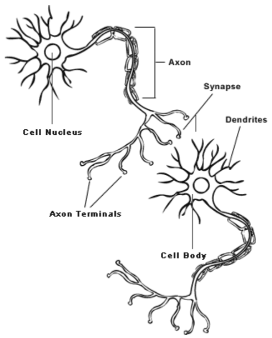
Why machine learning?

- Algorithm too complex for a human to design, but we can easily provide examples of what the algorithm should do.
- Relationships in high-dimensional data too complex for a human to see, but a computer can find these.
- The computer should learn and adapt continuously to new situations.

Applications of machine learning

- Pattern and speech recognition
- Robots & autonomous systems
- Big data
- Expert systems & decision support
- Games
- Models of the brain





Three main categories of machine learning methods

- **Supervised learning (predictive)**
  Learn to generalize and classify new data based on labelled training data.
  - Pattern recognition
  - Classification
  - Regression

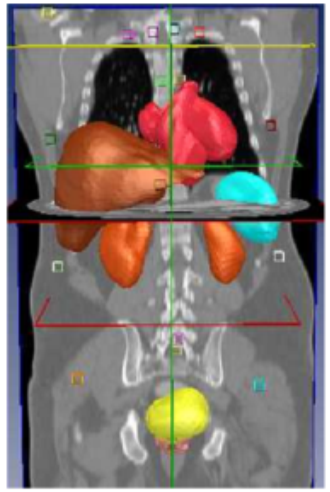
- **Unsupervised learning (descriptive)**
  Discover structure and relationships in complex high-dimensional data.

- **Reinforcement learning (active)**
  Generate policies/strategies that lead to a (possibly delayed) reward. Learning by doing.

Pattern recognition examples

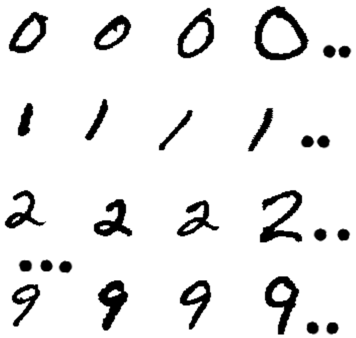
Face recognition

Organ segmentation

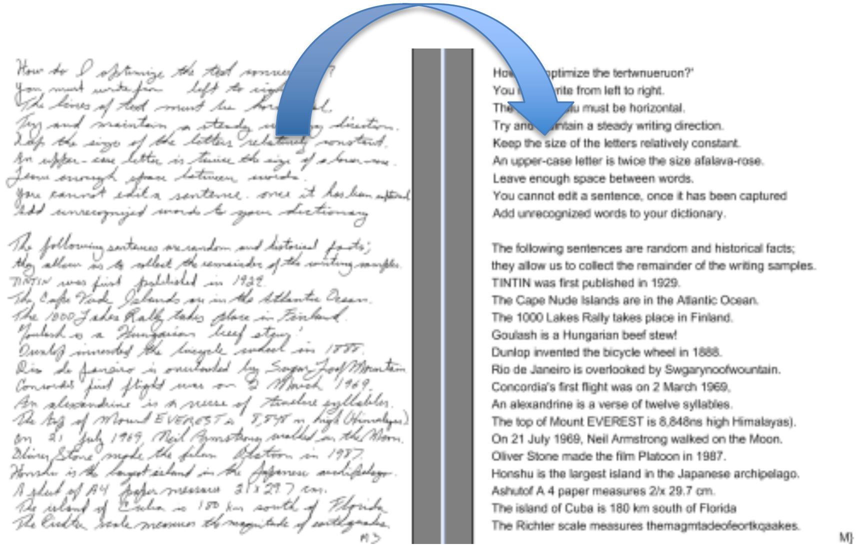


Pattern recognition examples

Optical Character Recognition (OCR)



Source: http://blog.damiles.com

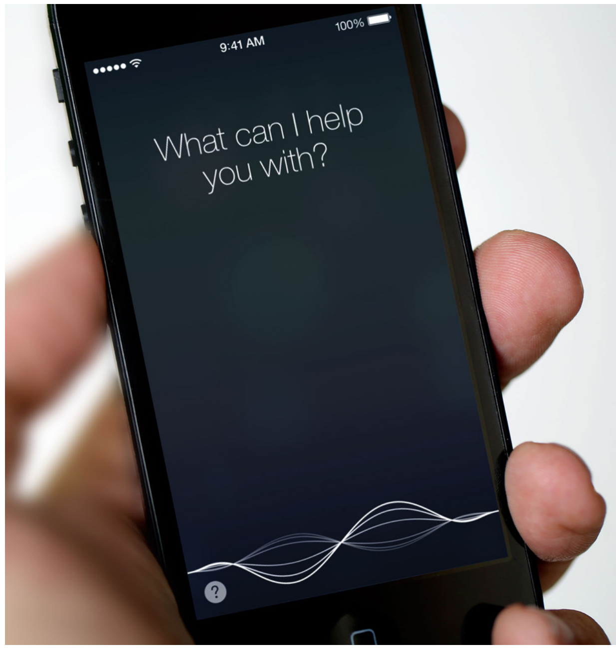


Pattern recognition examples

Xbox Kinect – Pose estimation

Pattern recognition examples

Speech recognition



Pattern recognition examples

Game positions

Spam filters

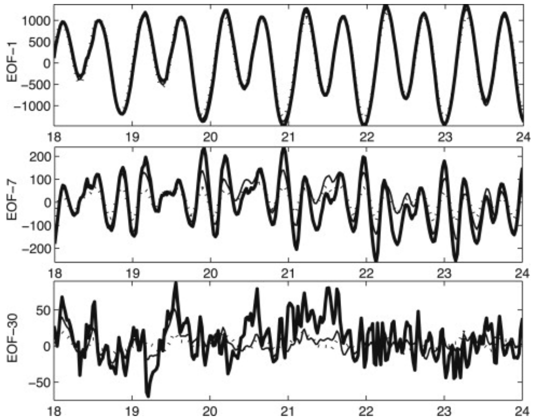
#1
100% satisfied
4U
Accept credit cards
Act Now!
Additional Income
Affordable
All natural

Movie & music recommendation

Regression examples

Prediction and forecasting

Weather and natural phenomena



Financial markets

Features

- A feature is a measurement or scalar number that describes some aspect of a phenomenon or object
  - Size, length, shape, velocity
  - Intensity and color (RGB)
  - Position (x,y)
  - Signal frequency
  - Sensor measurements (e.g., temperature)
  - Game piece present at certain location (yes/no)
  - Word present in an email (yes/no)

- Feature extraction is the process of measuring features from data.

Features – Iris dataset

Iris setosa

Iris versicolor

Iris virginica

Fisher's Iris Data

Sepal Length ♦	Sepal Width ♦	Petal Length ♦	Petal Width ♦	Species ♦
5.7	4.4	1.5	0.4	*I. setosa*
5.8	2.6	4.0	1.2	*I. versicolor*
5.8	2.7	5.1	1.9	*I. virginica*

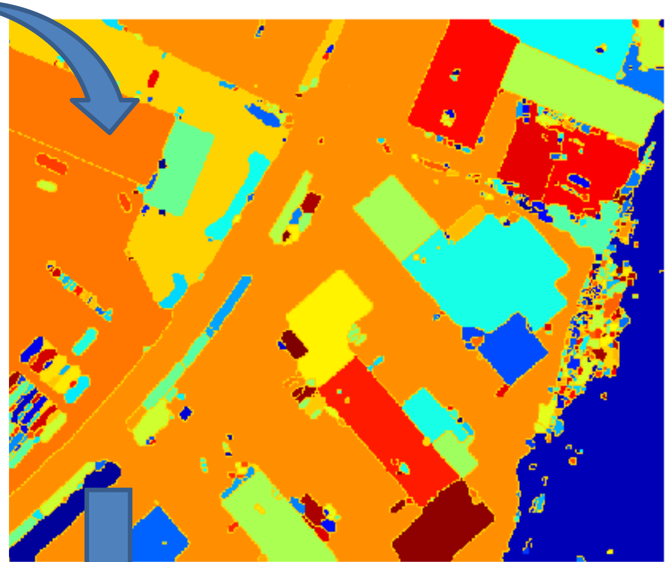
From Wikipedia

Feature vectors:

$$\mathbf{x}_1 = \begin{bmatrix} 5.7 \\ 4.4 \\ 1.5 \\ 0.4 \end{bmatrix} \quad \mathbf{x}_2 = \begin{bmatrix} 5.8 \\ 2.6 \\ 4.0 \\ 1.2 \end{bmatrix} \quad \mathbf{x}_3 = \begin{bmatrix} 5.8 \\ 2.7 \\ 5.1 \\ 1.9 \end{bmatrix}$$

Features – Image classification

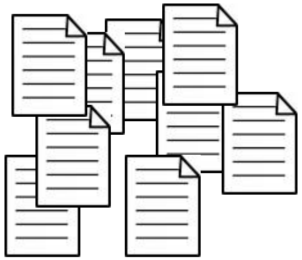
Watershed transform



Extract intensity & shape features

$$\mathbf{x} = \begin{bmatrix} \text{Mean intensity} \\ \text{Std intensity} \\ \text{Border regularity} \\ \text{Compactness} \end{bmatrix}$$

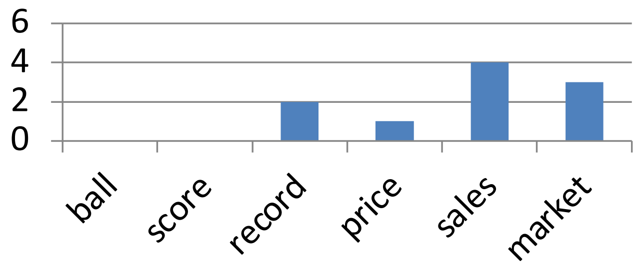
Features – Document analysis



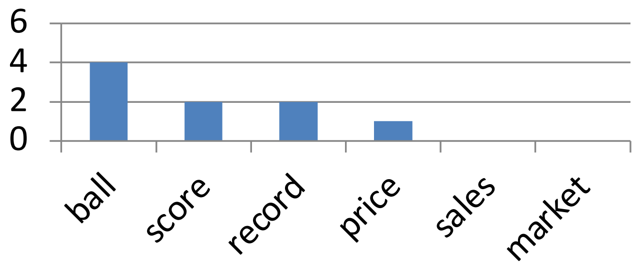
Articles, mail, web pages, ...

Dictionary:
{'ball', 'score', 'record', 'price', 'sales', 'market'}

Financial document



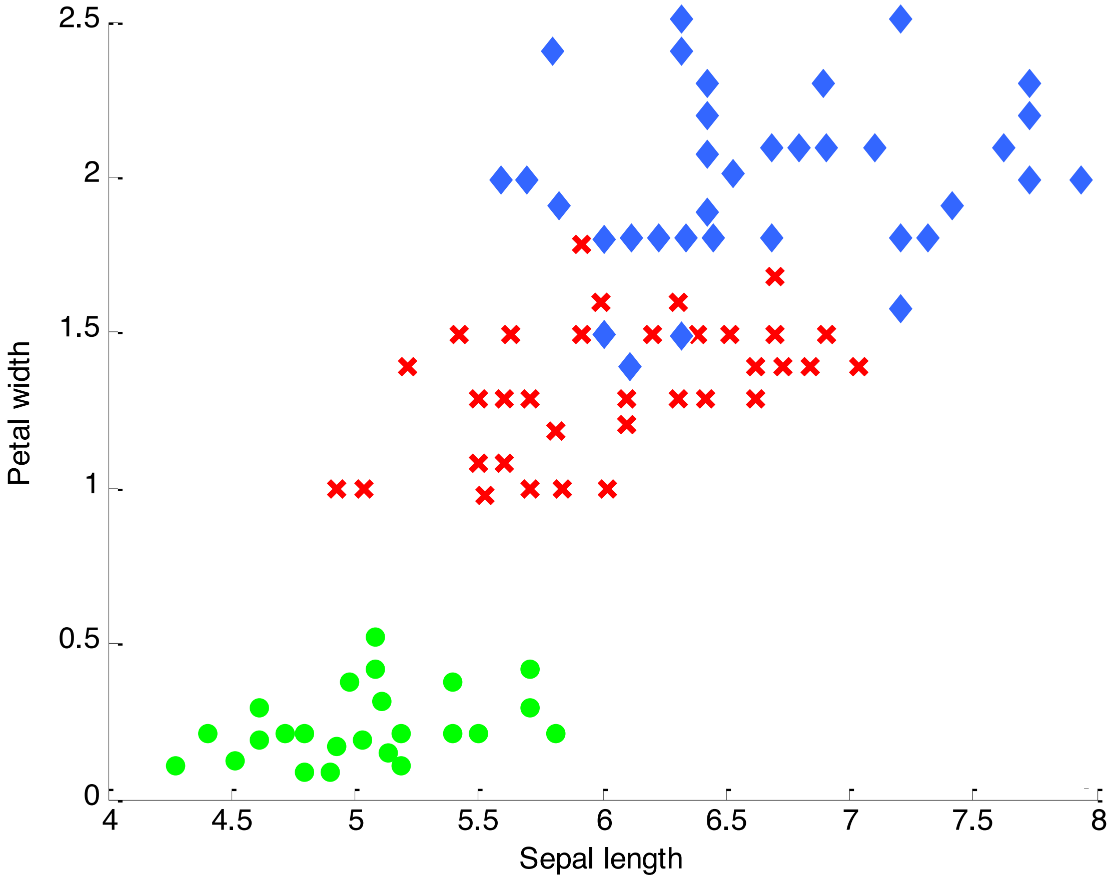
Sports document



$$\mathbf{x} = \begin{bmatrix} \#'ball' \\ \#'score' \\ \#'record' \\ \#'price' \\ \#'sales' \\ \#'market' \end{bmatrix}$$

Bag-of-words model

Feature space

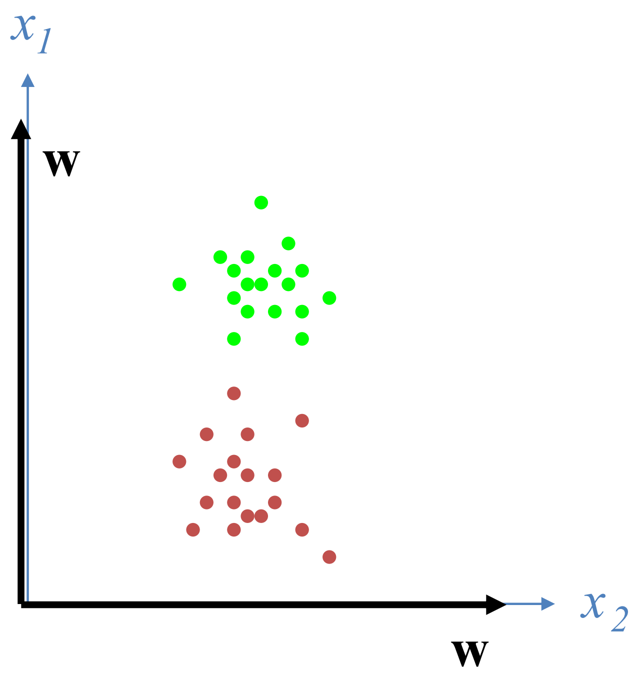


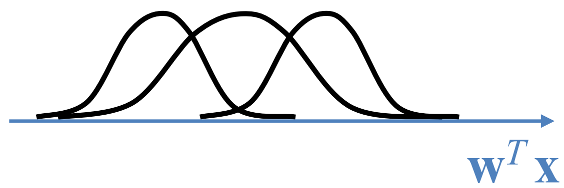

Iris setosa


Iris versicolor


Iris virginica

Representation





Supervised learning

- **Task:** Learn to predict/classify new data from labeled examples.
- **Input:** Training data examples {$\mathbf{x}_i$, y_i}, i=1...N, where $\mathbf{x}_i$ is a feature vector and y_i is a class label in the set Ω.
- **Output:** A function $f(\mathbf{x};w_1,\ldots,w_k) \rightarrow \Omega$

Find a function *f* and adjust the parameters $w_1,\ldots,w_k$ so that new feature vectors are classified correctly. ***Generalization!!***

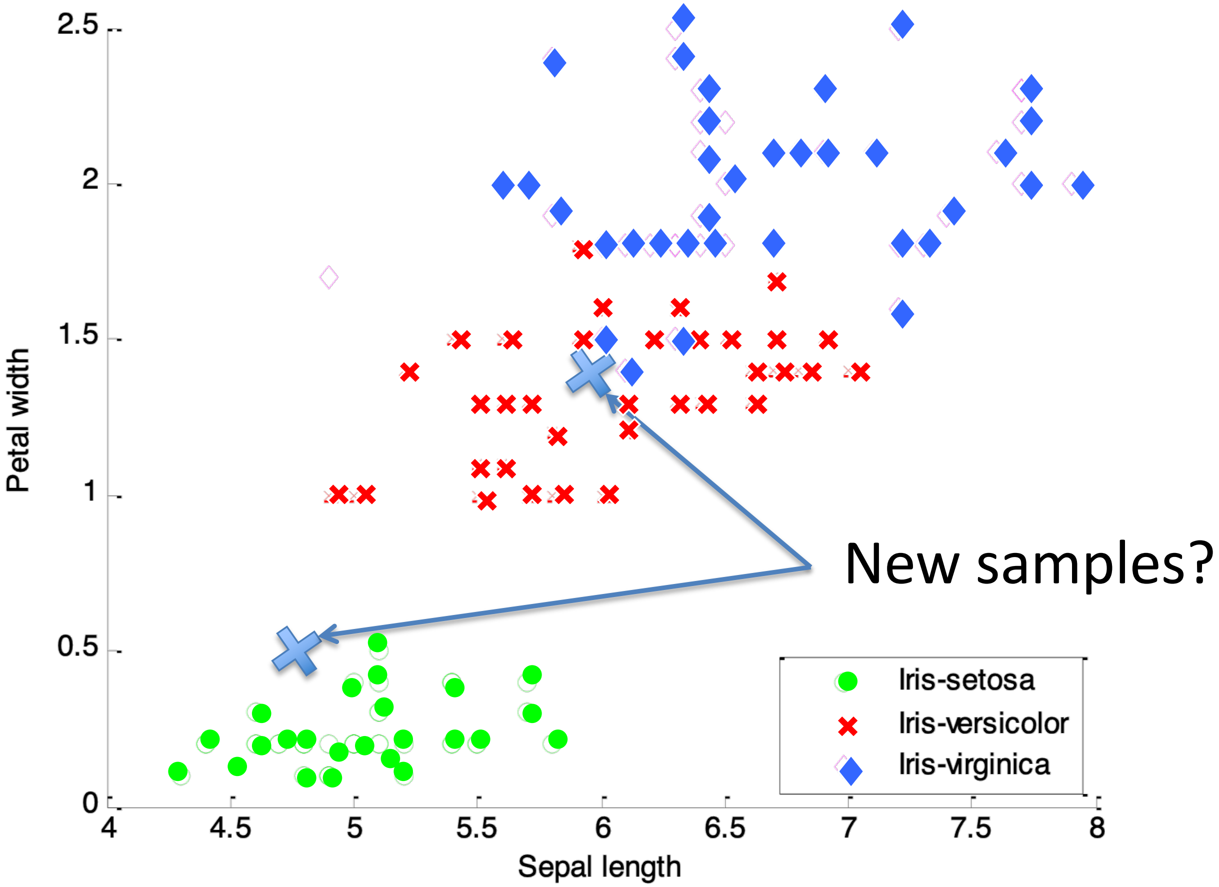
Classification vs. regression vs. ranking

- **Classification**: Select one of a discrete set of classes (the set Ω is discrete).
  - Which horse is going to win this race?
  - Which letter does this image depict?
  - Is this email spam (yes/no)?

- **Regression**: Learn to predict a continuous value ($\Omega = \mathbb{R}$).
  - Learn to predict the temperature tomorrow.
  - What is the *probability* that this image depicts the letter ‘a’?

- **Ranking**: Learn to rank a set of items ($\Omega = \mathbb{R}$).
  - Rank webpages, movies, etc.

Example - Classification



k-Nearest Neighbours (k-NN)

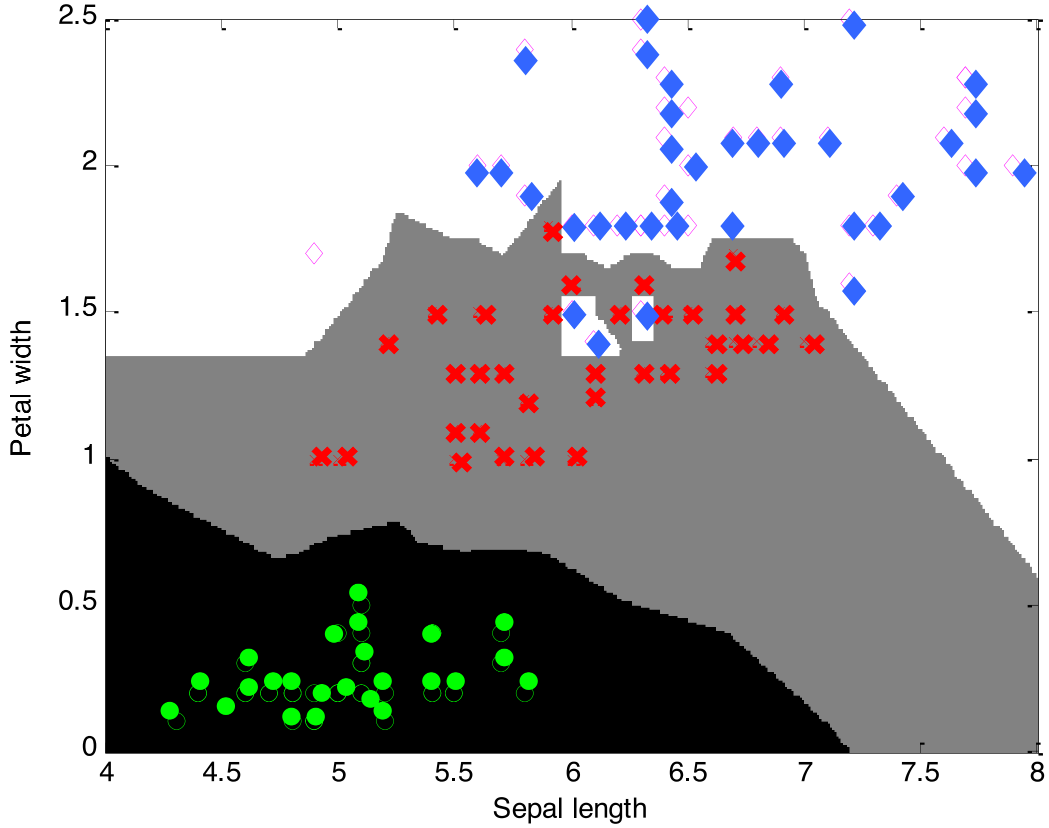
- Save all training data.
- For a new case, find similar examples among the training data.
- Requires a similarity measure (metric), for example the Euclidian distance

$$\|\mathbf{x}-\mathbf{y}\| = \sqrt{\sum_i (x_i - y_i)^2}$$

- A majority vote among the *k* nearest neighbours decides the class, where *k* can be 1,2,3,4...

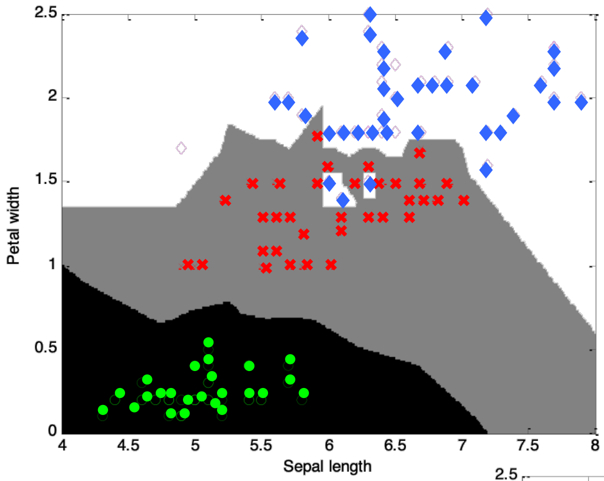
Classification areas

k = 1

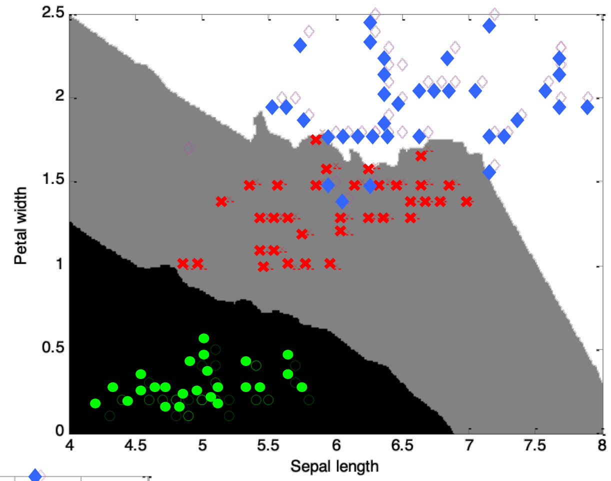


Classification areas

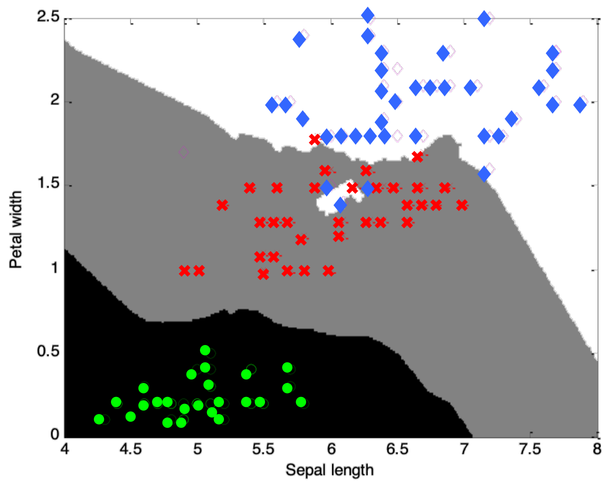
k = 1



k = 7



k = 3



Pros and cons of k-NN

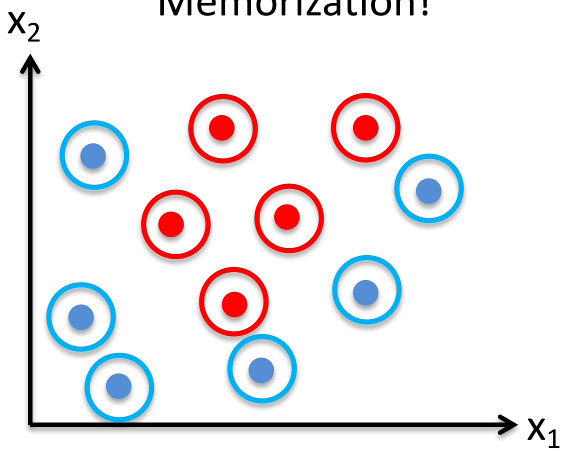
- Very simple – no “training“ or modeling required
- Must store all training data – problem for large data sets:

$$f(\mathbf{x}; \underbrace{w_1,\ldots,w_k}_{\text{Parameters equal to training data } \mathbf{x}_i}) \rightarrow \Omega$$

- Slow classification for large data sets – must compare new samples with all stored samples.

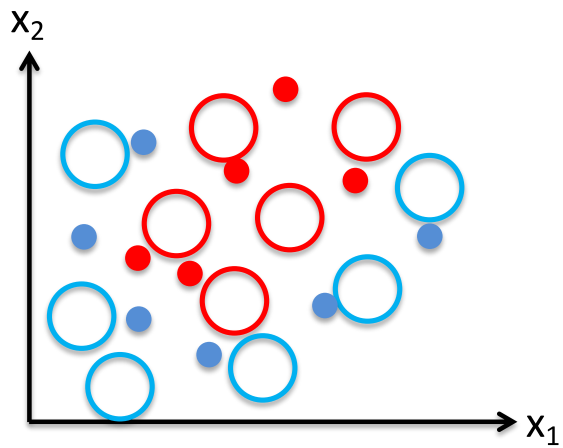
Generalization!

Classifying training data is trivial.
Memorization!



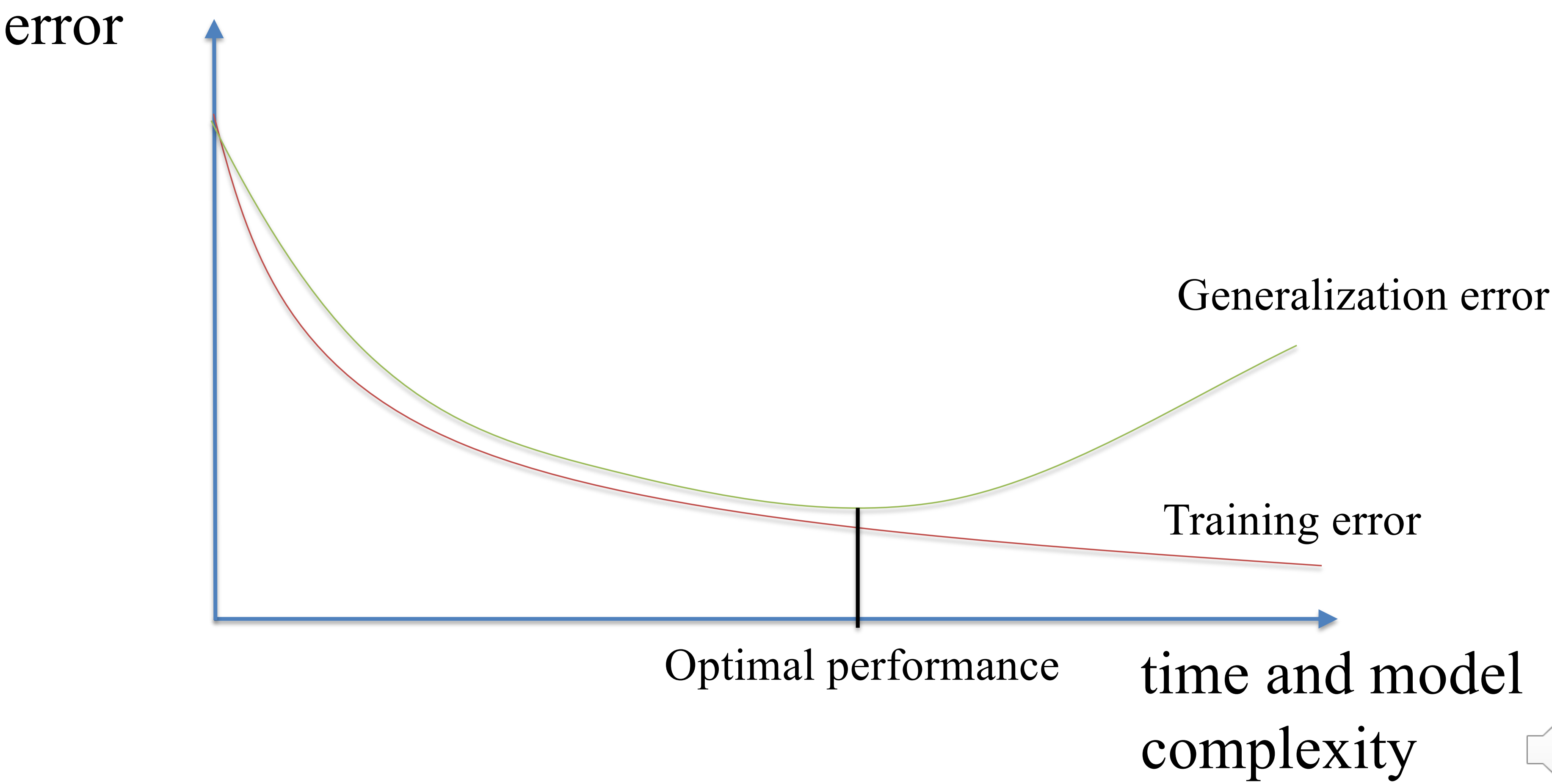
$f(\mathbf{x}; w_1, \ldots, w_k) \rightarrow \Omega$

Classifying **new data** is the challenge!



Generalization error!

Overfitting

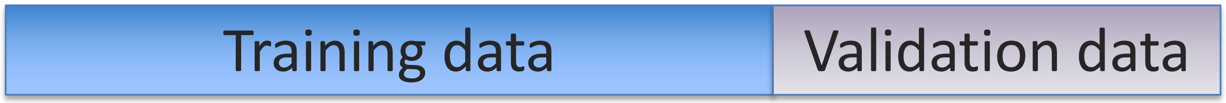


Training data vs. validation data

- A classifier must be able to generalize, i.e., it must be evaluated using previously unseen data.
- Evaluating using training data will give an overly optimistic accuracy.
- Three ways to perform the evaluation:
  - Hold out
  - Cross validation
  - Leave one out

Hold out

- Simplest approach, hold out one part of the entire data set as validation data.



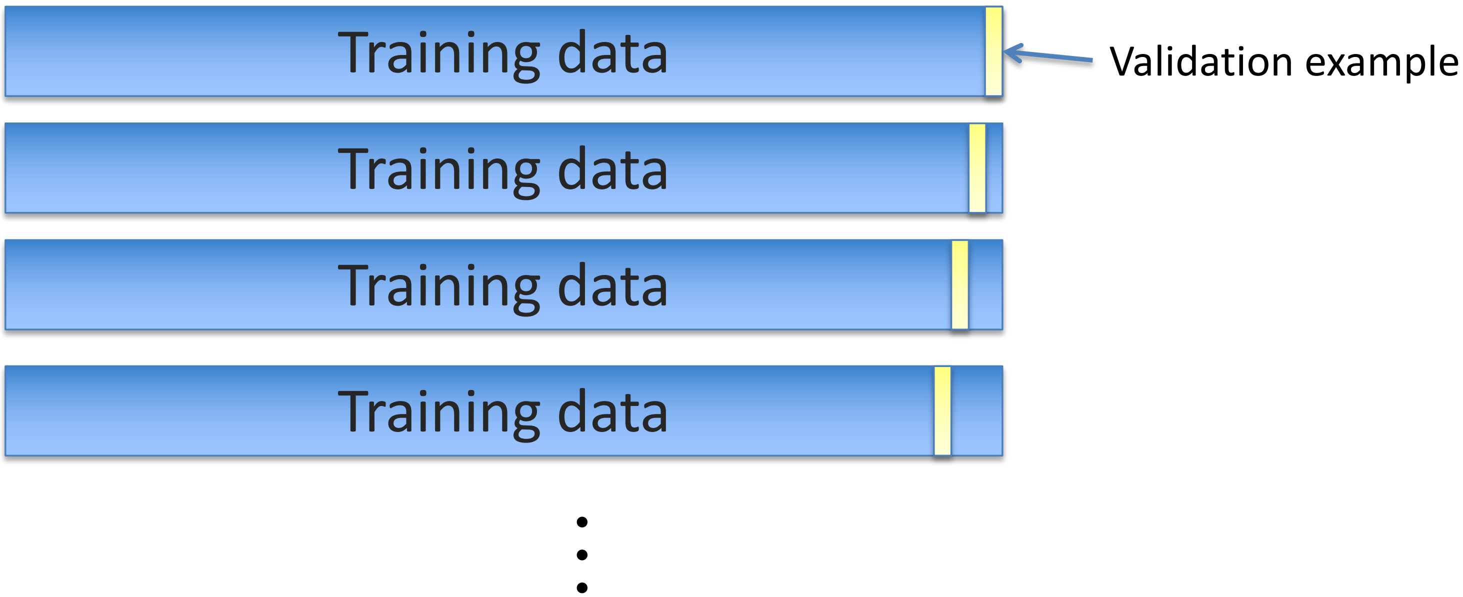
n-fold Cross-Validation

- Divide data set into n segments. Train using n-1 segments and validate using the n:th.
- Example of 3-fold Cross-Validation:

Training data	Training data	Validation data
Training data	Validation data	Training data
Validation data	Training data	Training data

Leave-one-out

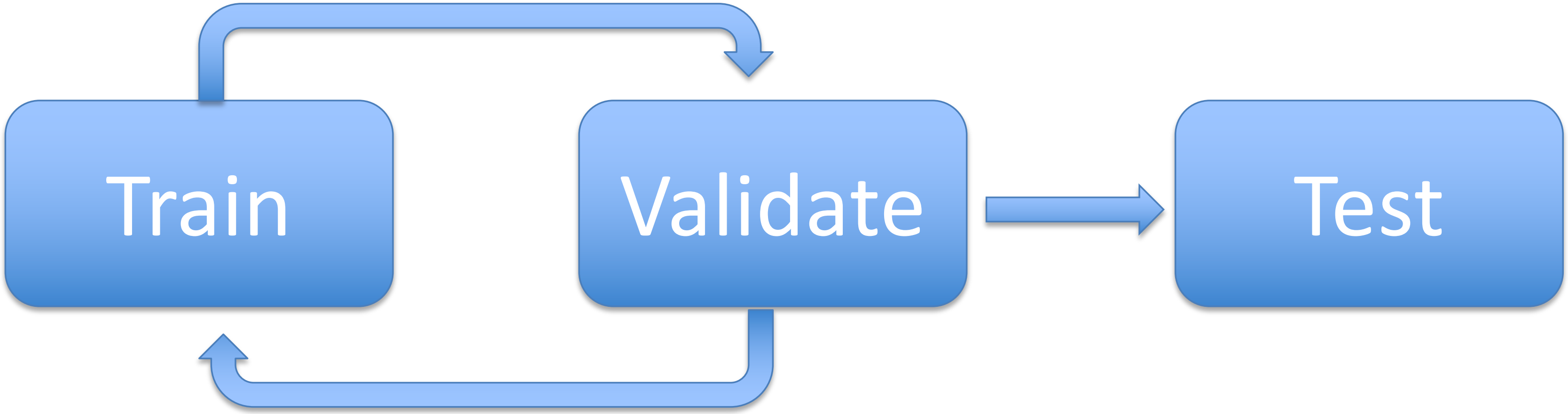
- Extreme case of Cross-Validation: Use all data but one example for training and use the last one to evaluate



How can we find the minimum generalization error?

- What happens if the generalization error is not low enough?
- Modify the classifier (change the model) and train again…
- But – then the validation data is used to select the model!
- How do we know how well the new model generalizes?
- Need new validation data to test the final model – this dataset is called test data.
- Test data must <u>never</u> be used more than once!

Training – Validation –Testing



Evaluating classifiers – The Confusion matrix


Iris setosa


Iris versicolor


Iris virginica

Predicted class

Actual class	Setosa	Versicol.	Virginica
Setosa	50	0	0
Versicol.	0	45	5
Virginica	0	7	43

Accuracy: $$\frac{50+45+43}{150}=92\%$$