Neural Networks and Learning Systems
TBMI26 / 732A55
2018

Lecture 1
Introduction

Magnus Borga
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 (tekfak (TBMI26) students only)
  - Chose group A, B or C for classes – and follow your group!
  - Chose one occasion for each lab (not connected to class)

Staff

- Examiner: Magnus Borga (magnus.borga@liu.se)
- Course admin: Anette Karlsson (anette.k.karlsson@liu.se)

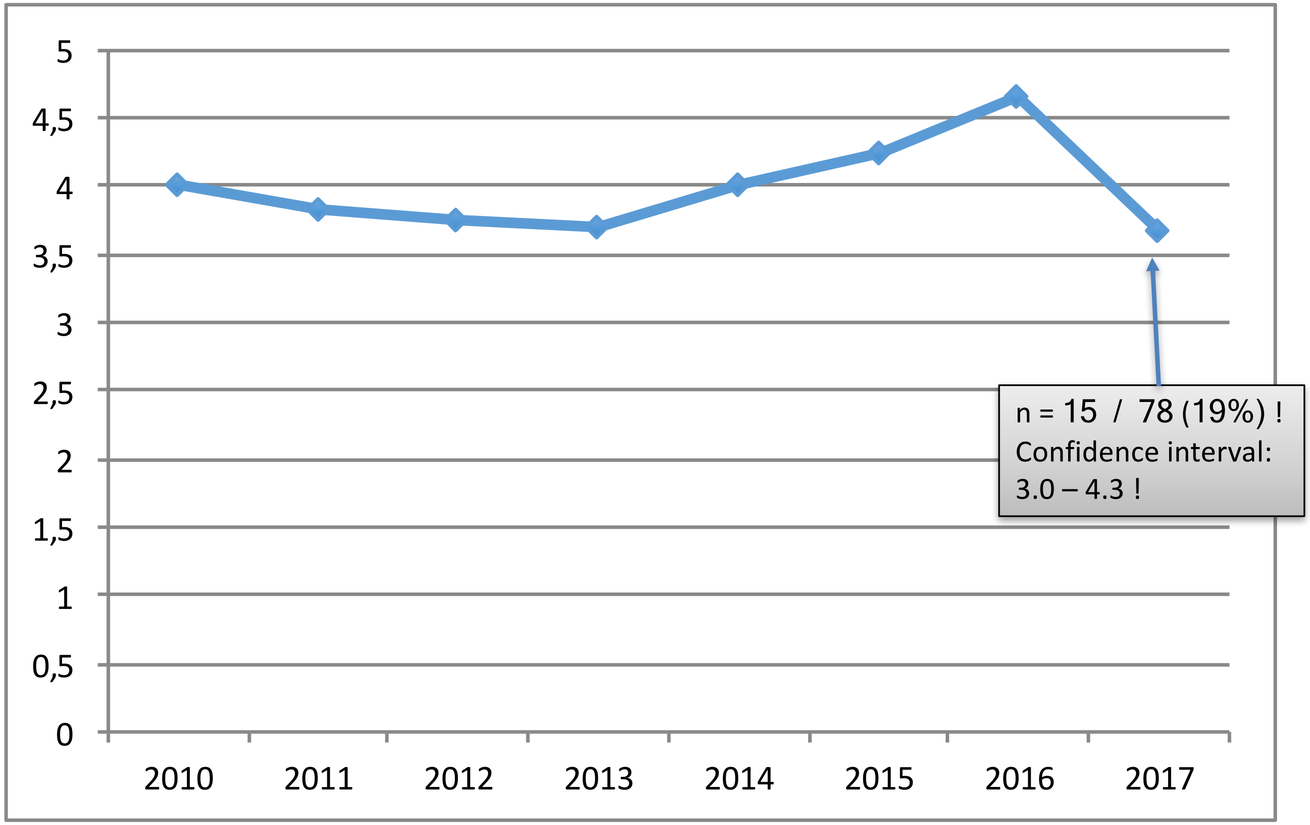
- Lectures: Magnus Borga, Ola Friman, Michael Felsberg
- Lessons: Anette Karlsson, Martin Hultman, Lasse Alfredsson
- Labs:
  – Annette Karlsson, Martin Hultman, Nikolajs Prihodko, Zhendong Wang
  – Abdelrahman Eldesokey , Mikael Persson, Hannes Ovrén

Course evaluation and development

På en femgradig skala ger jag kursen sammanfattningsbetyget / On a scale 1-5 (5 being the best) I give the overall credit to this course

Svarsalternativ	Antal svar	Andel	Visa grupp
1	1	7%	→
2	1	7%	→
3	5	33%	→
4	3	20%	→
5	5	33%	→

Overall credit over time



The Course - Overview

- 9 lectures
- 9 lessons
- 4 assignments
- 1 written exam

Must be completed

- Course language is English.

The Course - Lectures

PPT lectures, handouts on course page

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

The Course - Lessons

- One lesson after each lecture
- Pen & paper exercises
- Complementary presentations
- Preparations and help with assignments
- Three groups for TBMI26 and one group for 732A55
  - Choose group on Lisam and follow that group!

The Course - Assignments

- 4 laboratory exercises/assignments:
  1. Pattern recognition using linear classifiers and neural networks
  2. Face recognition in images using Boosting techniques
  3. Deep learning
  4. Reinforcement learning
- Matlab and some Python (#3)
- Assignments are done in pairs. (Not more than 2 students together!)
- Supervision time scheduled (“Laboration” in schedule) –
  - For each lab there are three occations for TBMI26 and one occation for 732A55
  - Choose occation on Lisam – Number of participants limited!

- Deadlines for <u>written reports</u>.
- Late reports will not be corrected until the re-exam in June.

Course literature

- Lecture notes
- Recommended reading on the Lectures-page
- Exercise collection
- Assignments

- Additional link collection on course pages
  – Video lectures
  – Online books
  – Articles, tutorials
  – Applets

Prerequisites

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

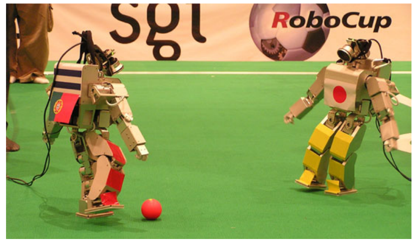
Challenges this year

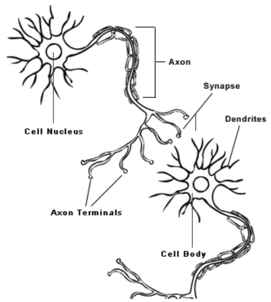
- Number of students
  - More important to follow your group schedule (lessons and labs)
  - Feedback on assignments may be delayed
- New lecture, lesson and lab on deep learning and CNN.
  - Lab not tested before
- A lot of new staff

Machine Learning

Applications of machine learning

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





What is (machine) learning?

Encyclopaedia Britannica (1964):
"Any relatively permanent change in behaviour resulting from past experience."

Bishop (2006):
"The core objective of a learner is to generalize from its experience."

Wikipedia (2015):
"Machine learning is a scientific discipline that explores the construction and study of algorithms that can learn from data. Such algorithms operate by building a model based on inputs and using that to make predictions or decisions, rather than following only explicitly programmed instructions."

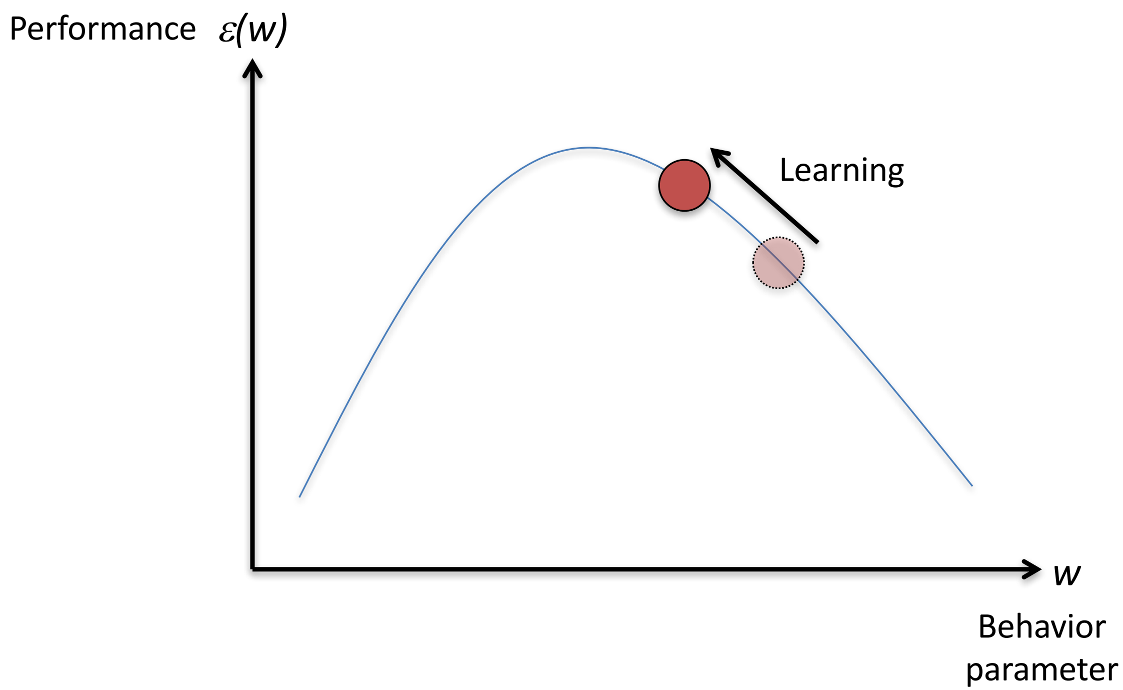
How can a machine learn?

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

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

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

Learning = optimization



Mathematical foundations of machine learning

Optimize parameters for optimal behaviour

Machine Learning

Optimization theory

Probability theory

Multidimensional calculus

Linear algebra

Model uncertainty

Why machine learning?

- Algorithm too complex for a human to implement, but we can easily generate 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.

Big companies are using it

- Microsoft
  http://research.microsoft.com/en-us/about/our-research/machine-learning.aspx
- Google
  http://research.google.com/pubs/ArtificialIntelligenceandMachineLearning.html
- IBM
  http://www.research.ibm.com/cognitive-computing/
- Apple, Facebook, Saab, Ericsson, ... (long list)

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.

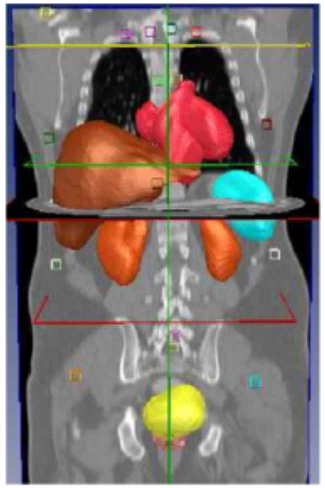
- **Reinforcement learning (active)**

  Generate policies/strategies that lead to a (possibly delayed) reward. Learning by doing.

Pattern recognition examples

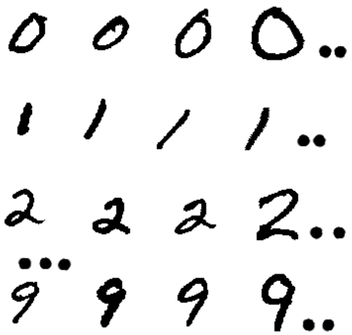
Face recognition

Organ segmentation

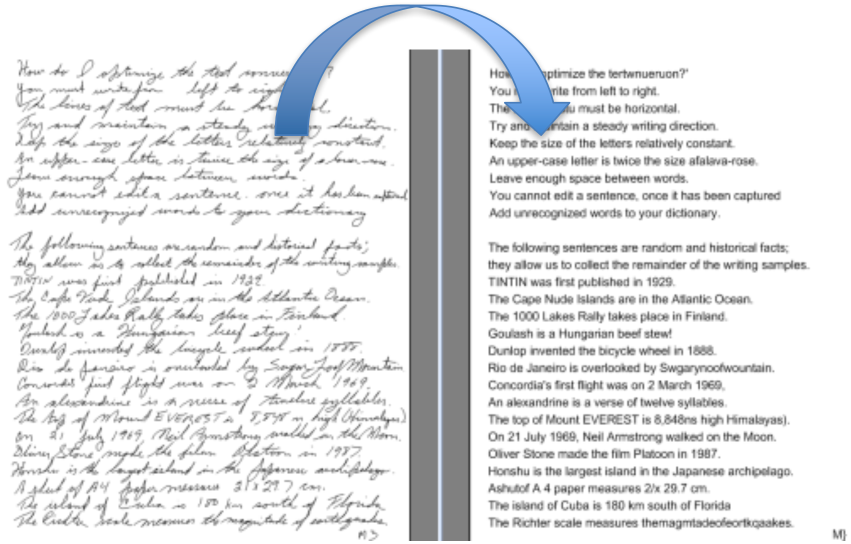


Pattern recognition examples

Optical Character Recognition (OCR)



Source: http://blog.damiles.com

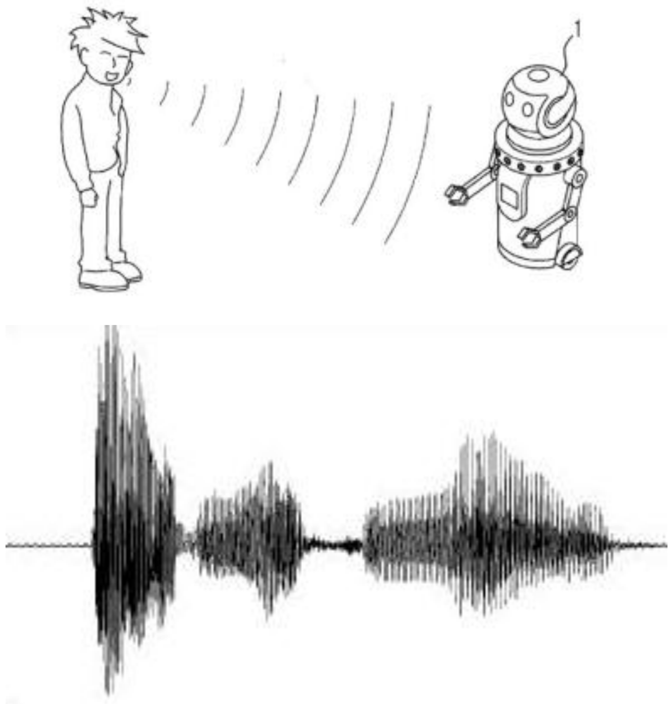


Pattern recognition examples

Xbox Kinect – Pose estimation

Pattern recognition examples

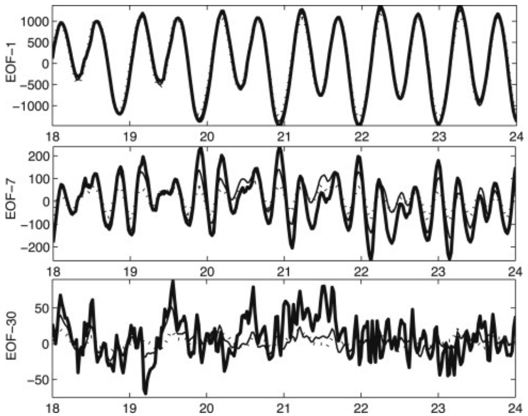
Speech recognition



Pattern recognition examples

Prediction and forecasting

Weather and natural phenomena



Financial markets

Pattern recognition examples

Game positions

Spam filters

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

Movie & music recommendation

Spotify

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*

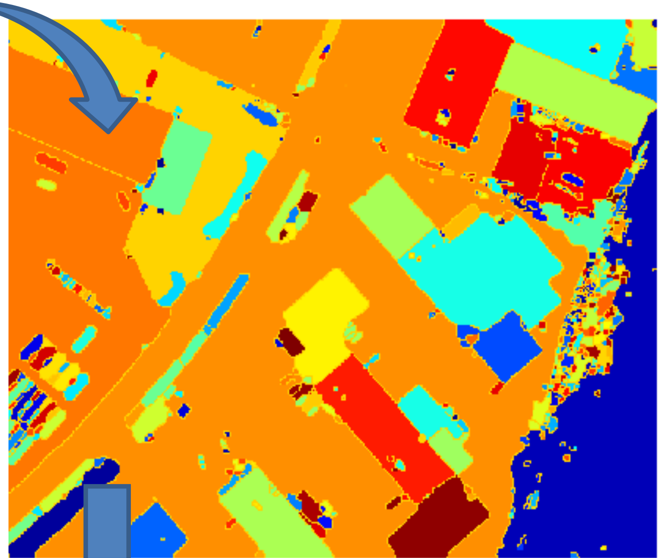
From Wikipedia

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

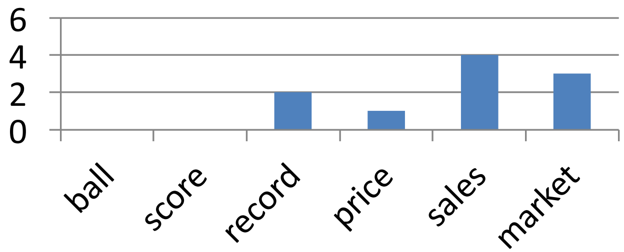
$$\mathbf{x} = \begin{bmatrix} \text{Mean intensity} \\ \text{Std intensity} \\ \text{Border regularity} \\ \text{Compactness} \end{bmatrix}$$

Features – Document analysis

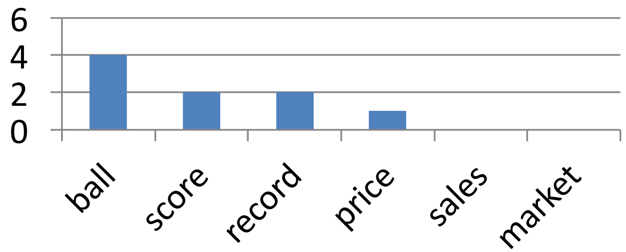
Articles, mail, web pages, ...

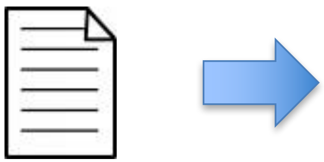
Dictionary:
{’ball’, ’score’, ’record’, ’price’, ’sales’, ’market’}

Financial document



Sports document

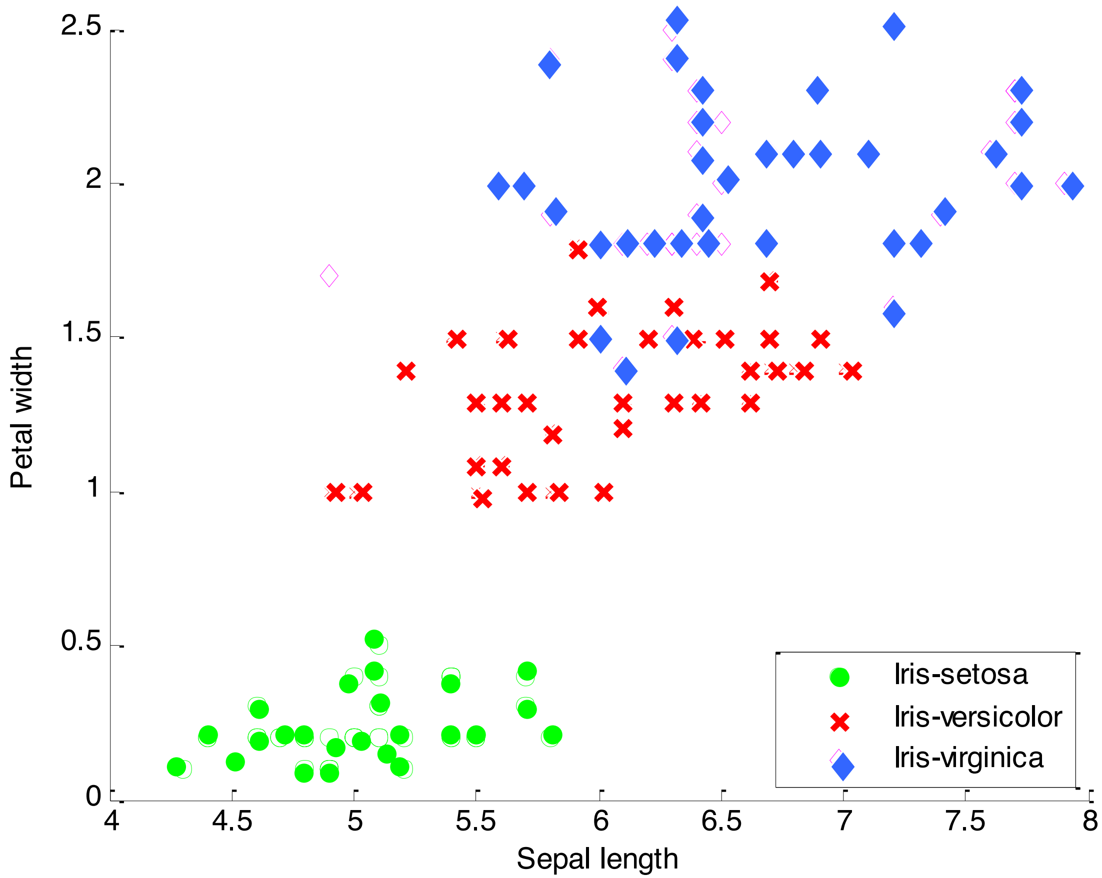




$$\mathbf{x} = \begin{bmatrix} \#'ball' \\ \#'score' \\ \#'record' \\ \#'price' \\ \#'sales' \\ \#'market' \end{bmatrix}$$

Bag-of-words model

Feature space

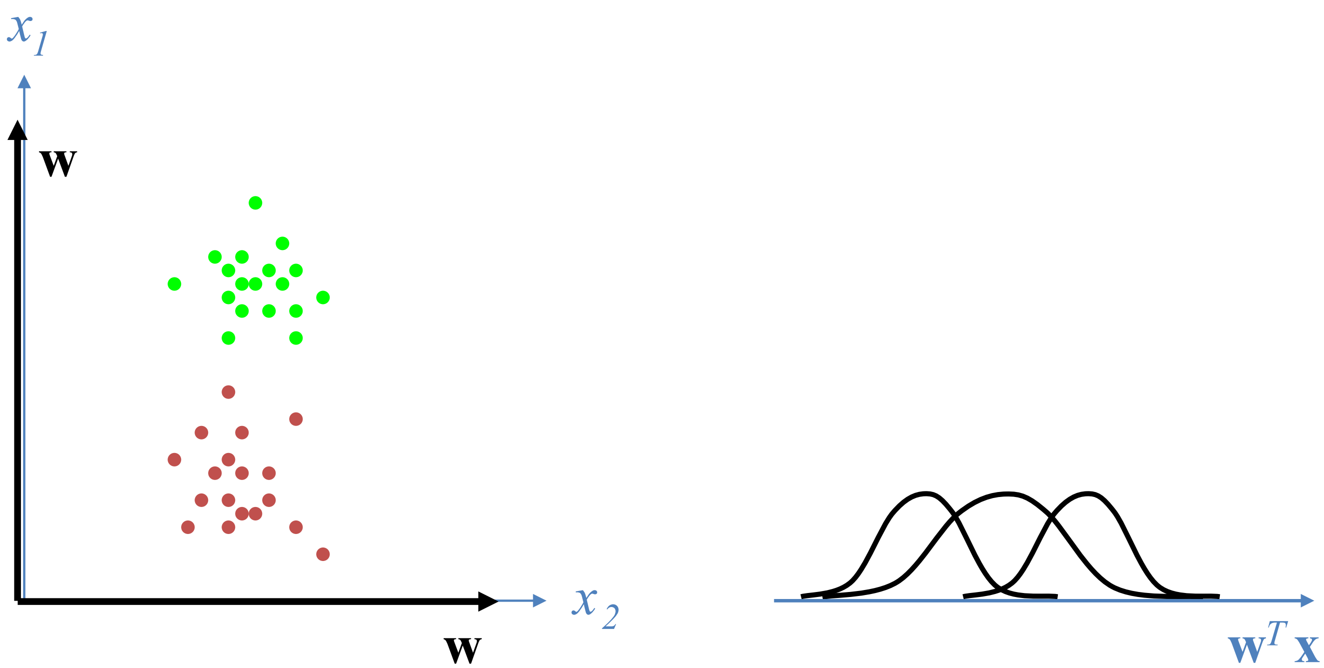



Iris setosa


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!!***

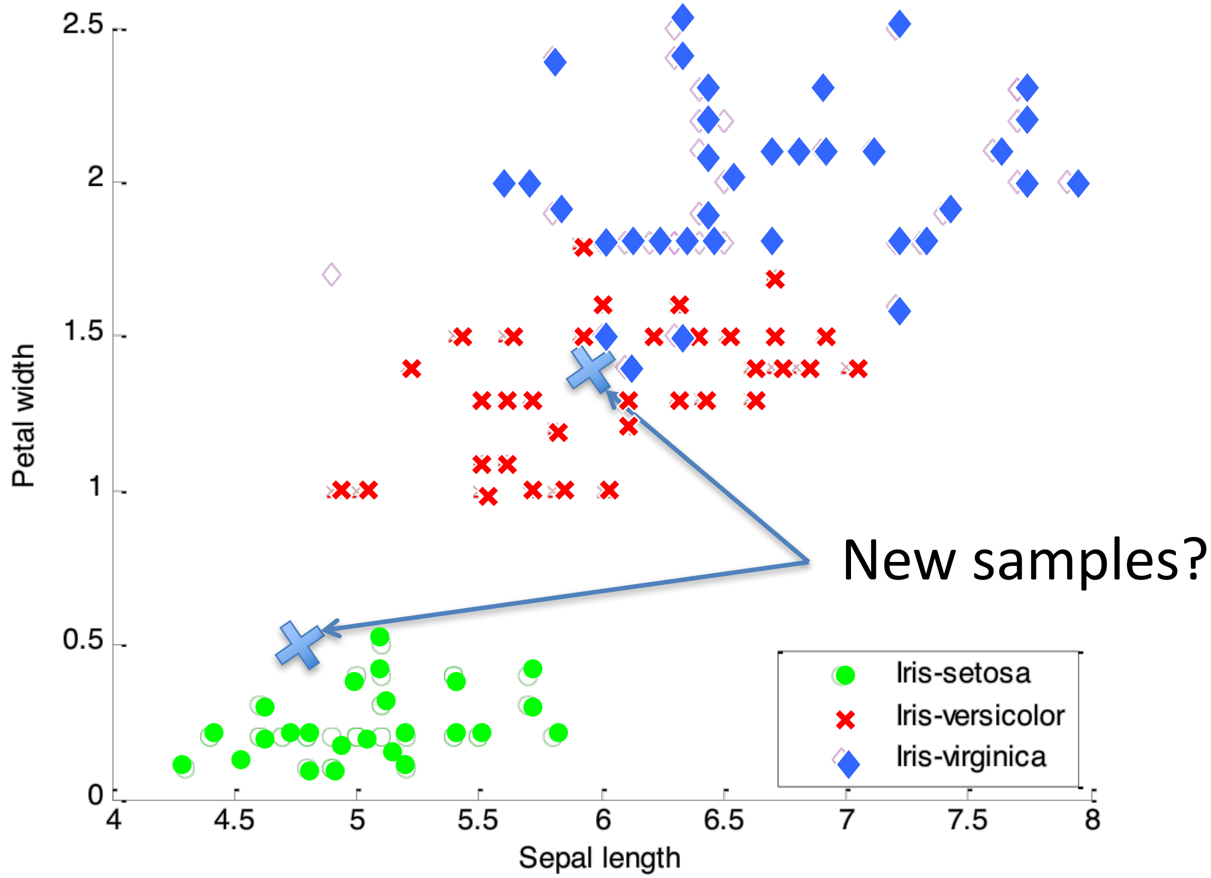
Classification vs. regression vs. ranking

- **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


2.5
2
1.5
1
0.5
0
4
4.5
5
5.5
6
6.5
7
7.5
8
Petal width
Sepal length
New samples?
Iris-setosa
Iris-versicolor
Iris-virginica

k-Nearest Neighbours (k-NN)

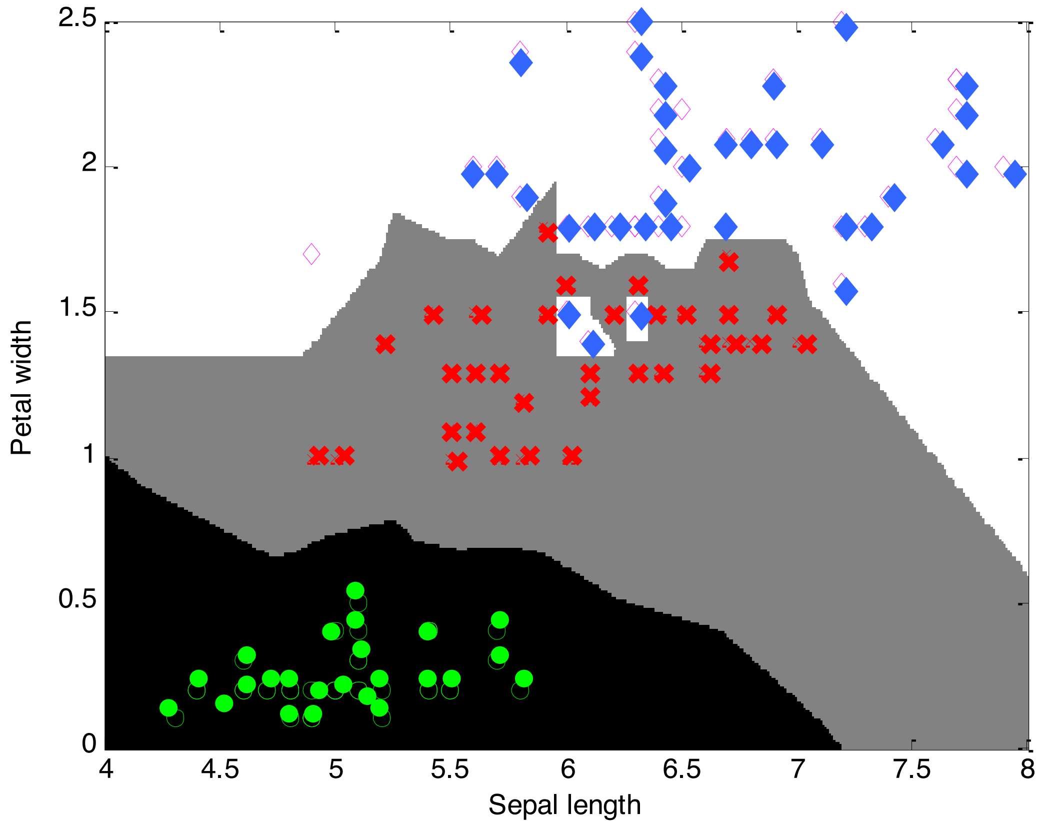
- Save all training data.
- For a new case, find similar examples among the training data.
- Requires a similarity measure (metric), for example the Euclidian distance

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

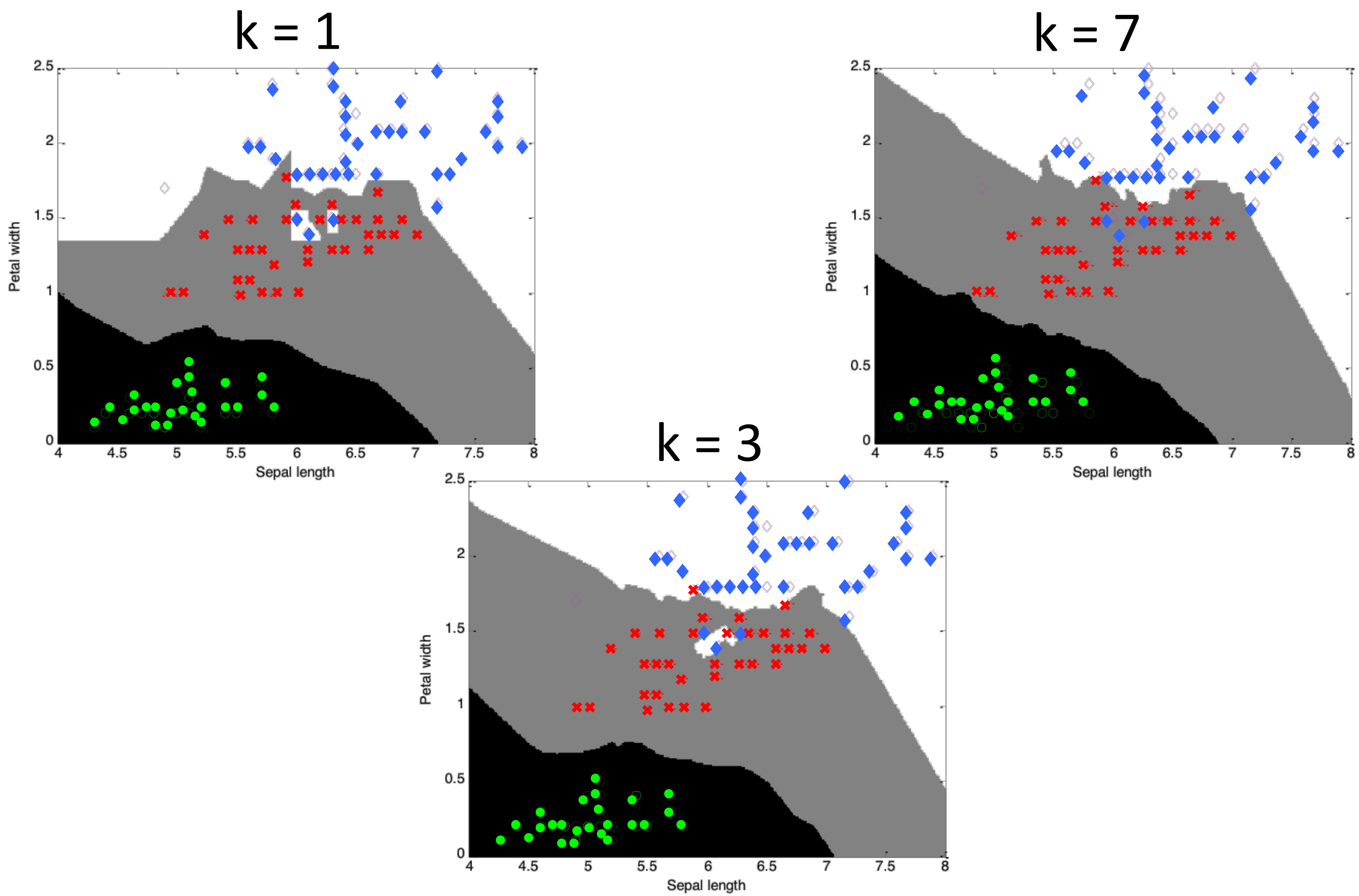
- A majority vote among the *k* nearest neighbours decides the class, where *k* can be 1,2,3,4...

Classification areas

k = 1



Classification areas



Pros and cons of k-NN

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

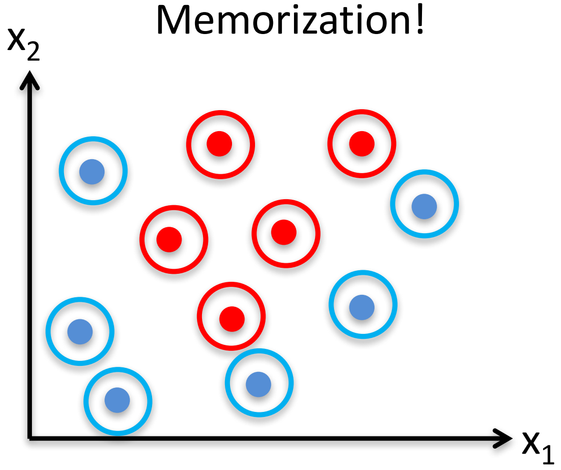
$$f(\mathbf{x};\underbrace{w_1,\ldots,w_k}_{\text{Parameters equal to training data } \mathbf{x}_i}) \rightarrow \Omega$$

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

Generalization!

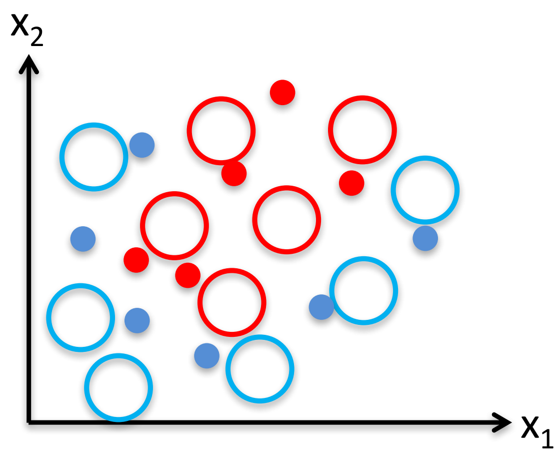
"The core objective of a learner is to generalize from its experience."

Classifying training data is trivial.
Memorization!



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

Classifying **new data** is the challenge!



Generalization error!

Evaluating classifiers

- How can we compare the performance of different classifiers?
- What happens if we use the same data for training and evaluation?
- How can we train and test a classifier if we only have a finite amount of collected data?

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

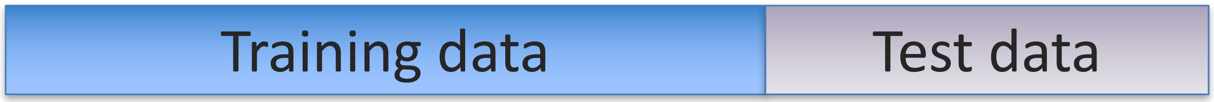
Accuracy: $$\frac{50+45+43}{150} = 92\%$$

Training data vs. test data

- A classifier must be able to <u>generalize</u>, i.e., it must be tested using <u>previously unseen</u> 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 test data.



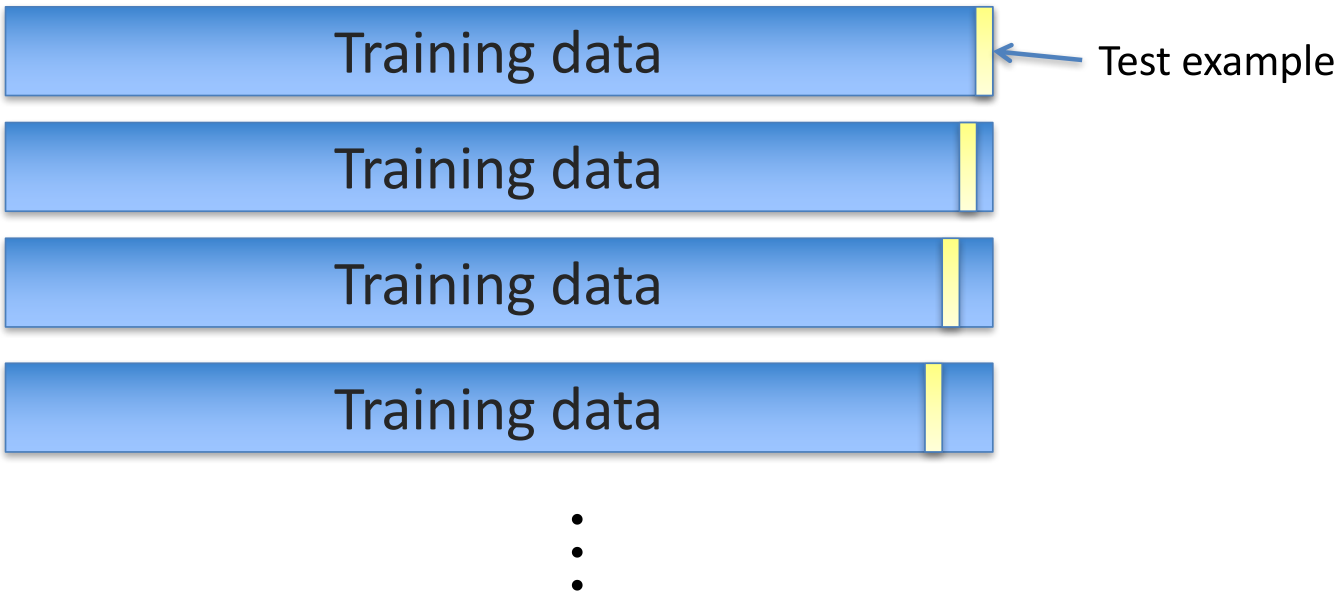
n-fold Cross-Validation

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

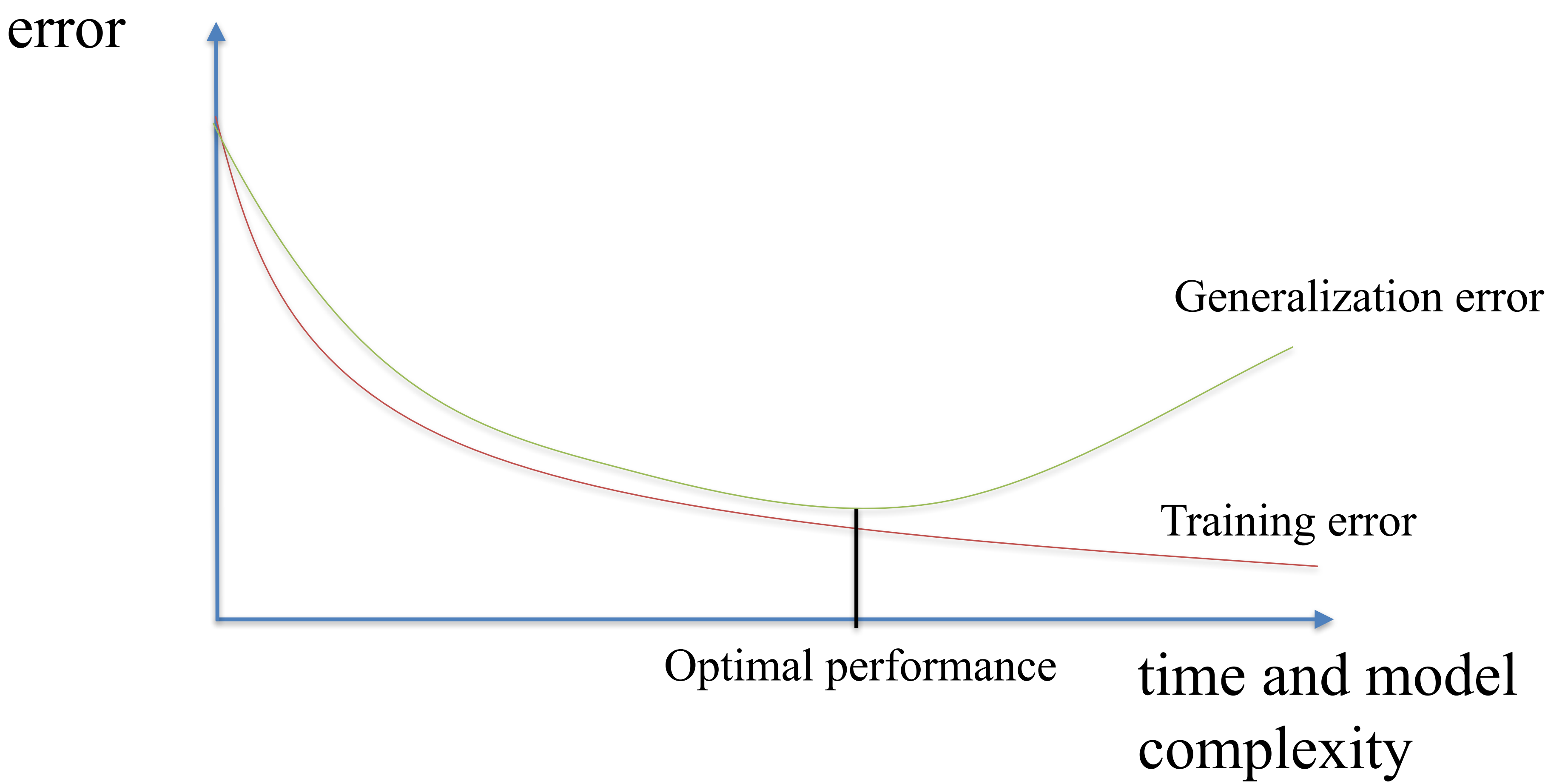
Training data	Training data	Test data
Training data	Test data	Training data
Test data	Training data	Training data

Leave-one-out

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



Overfitting



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