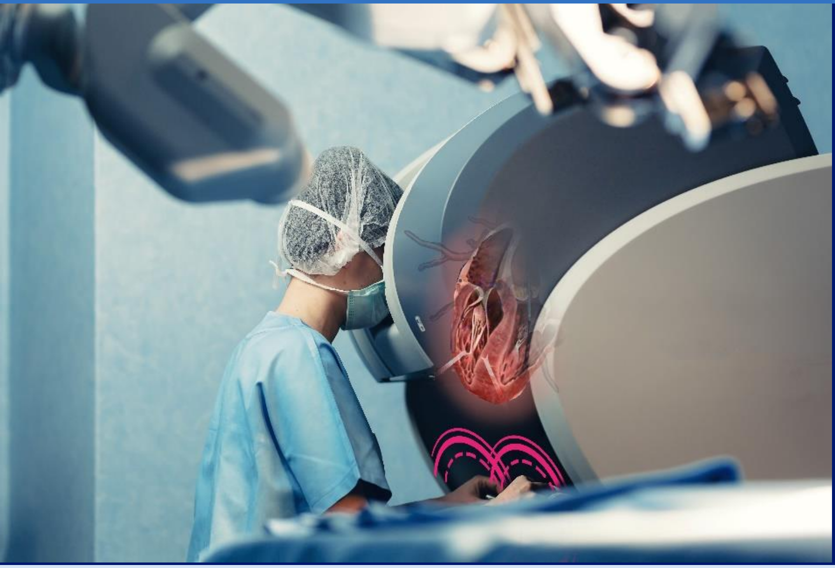
Computer- and robot-assisted Surgery



NATIONALES CENTRUM
FÜR TUMORERKRANKUNGEN
PARTNERSTANDORT DRESDEN
UNIVERSITÄTS KREBSCENTRUM UCC

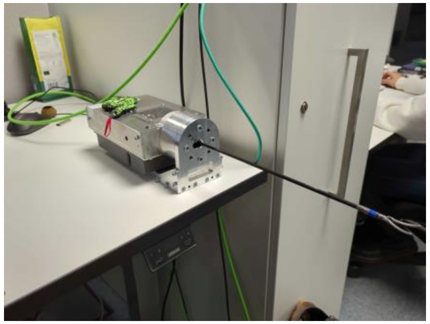
getragen von:
Deutsches Krebsforschungszentrum
Universitätsklinikum Carl Gustav Carus Dresden
Medizinische Fakultät Carl Gustav Carus, TU Dresden
Helmholtz-Zentrum Dresden-Rossendorf

Sebastian Bodenstedt

Basics of Machine Learning

SHK – Development and maintainance of a surgical robot testbed

- Help developing control software for our robot testbed i.e.:
  - Industrial 7-DOF robot (franka emika - panda / universal robots - ur5e)
  - 3D input device (forcedimension - lambda 7)
  - Custom end effector for controlling laparoscopic instruments



- Goal: system to perform surgical tasks with e.g. for reinforcement / imitation learning applications

<u>Prerequisites</u>

C/C++

(ROS/ROS2)

Send us your application!

Martin Lelis:
martin.lelis@nct-dresden.de
Ariel Rodriguez:
ariel.rodriguezjimenez@nct-dresden.de

Master/Diploma thesis - Recognition and/or prediction of fine grained surgical actions in a laparoscopic setting using deep neural networks

- Supervised Learning task using deep neural networks
- Usage of temporal video and sensor data

Prerequisite: Python

Useful prior knowledge:

Pytorch | Machine Learning fundamentals

Write me if you’re interested:

Martin Lelis
martin.lelis@nct-dresden.de

Content

- Introduction
  - What and why
- Supervised Learning
  - Introduction
  - Linear Methods for Regression and Classification
  - Tree-Based Methods
- Unsupervised Learning
  - Introduction
  - Cluster Analysis
- Model Assessment and Selection

What is machine learning?

The subfield of computer science that "gives computers the ability to learn without being explicitly programmed".

Arthur Samuel, 1959

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.

Tom Mitchell 1997

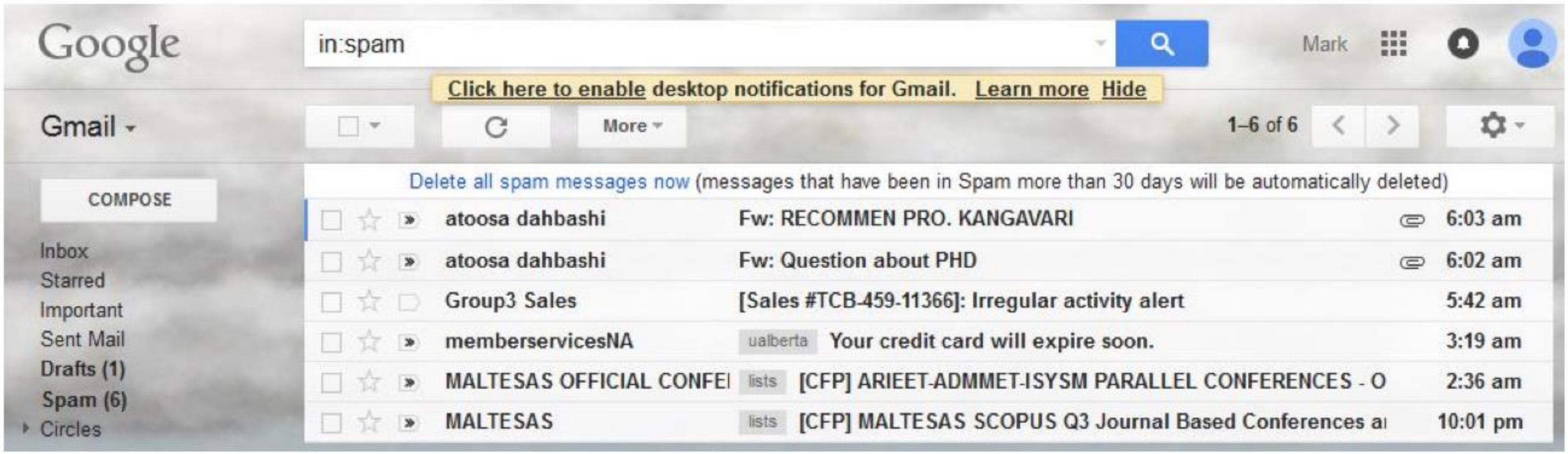
What is machine learning?

- A subset of artificial intelligence
- Mathematical models build on sample data
- Make predictions or decisions without being programmed to do so
- Locate patterns in data

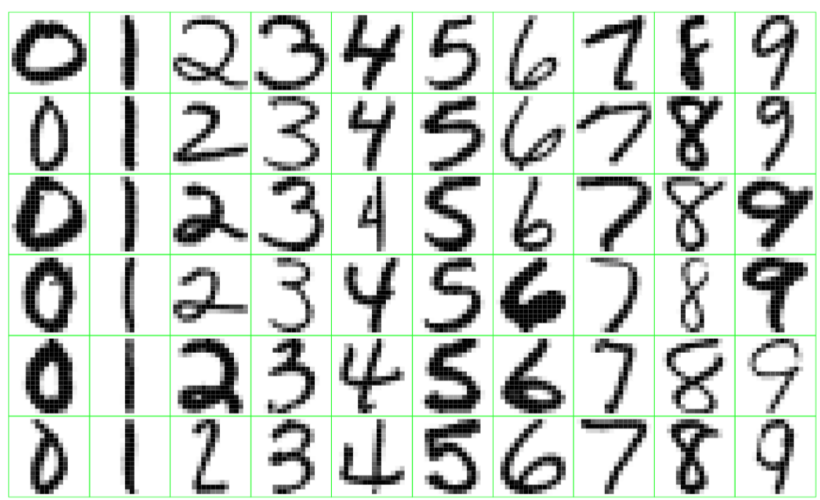
NCT

Introduction – Applications

- *Spam filtering*: Try to predict whether an email is junk email

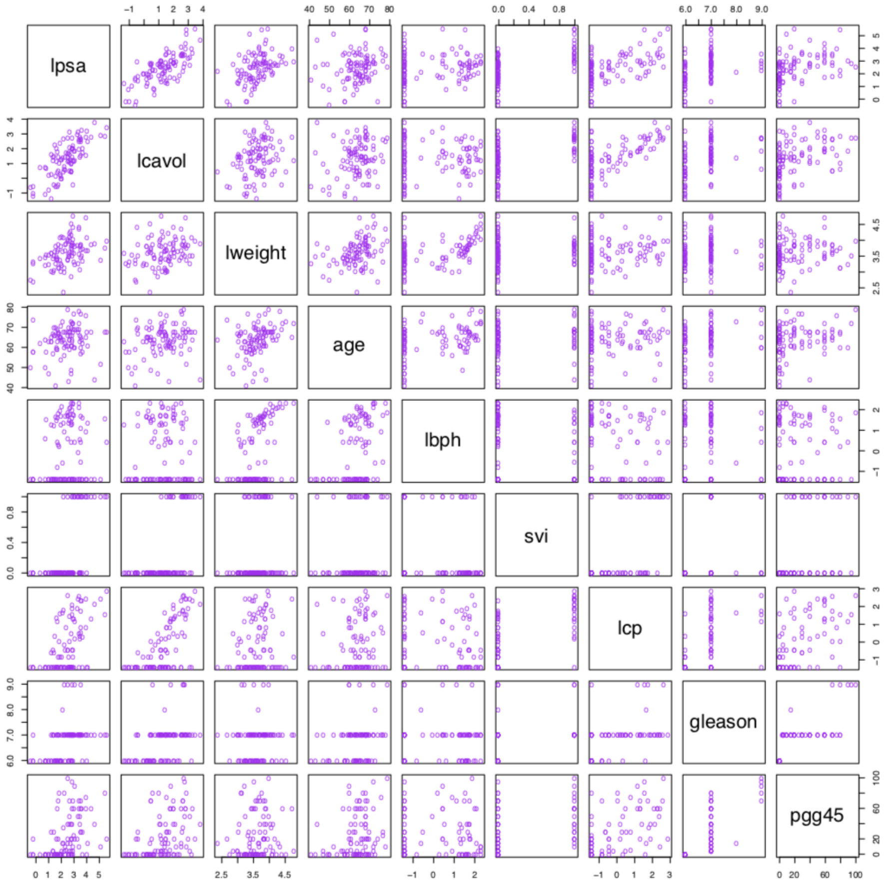


- *Handwritten digits*:
  Identify the numbers in a handwritten ZIP code, from a digitised image

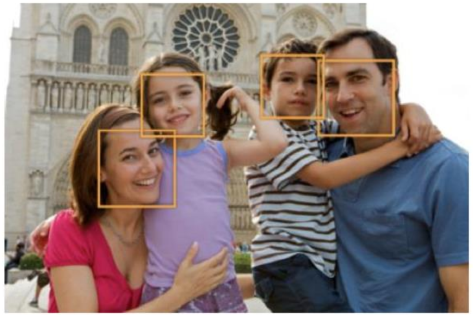


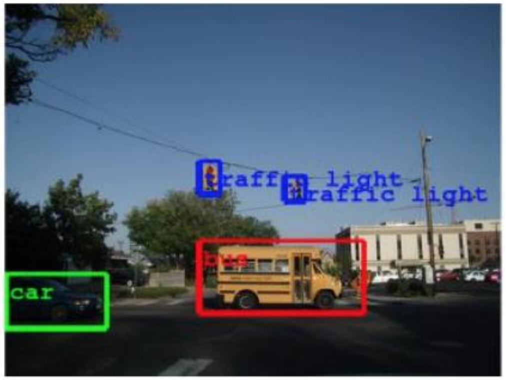
Introduction – Applications

- *Prostate cancer*: Identify the risk factors for prostate cancer, based on clinical and demographic variables



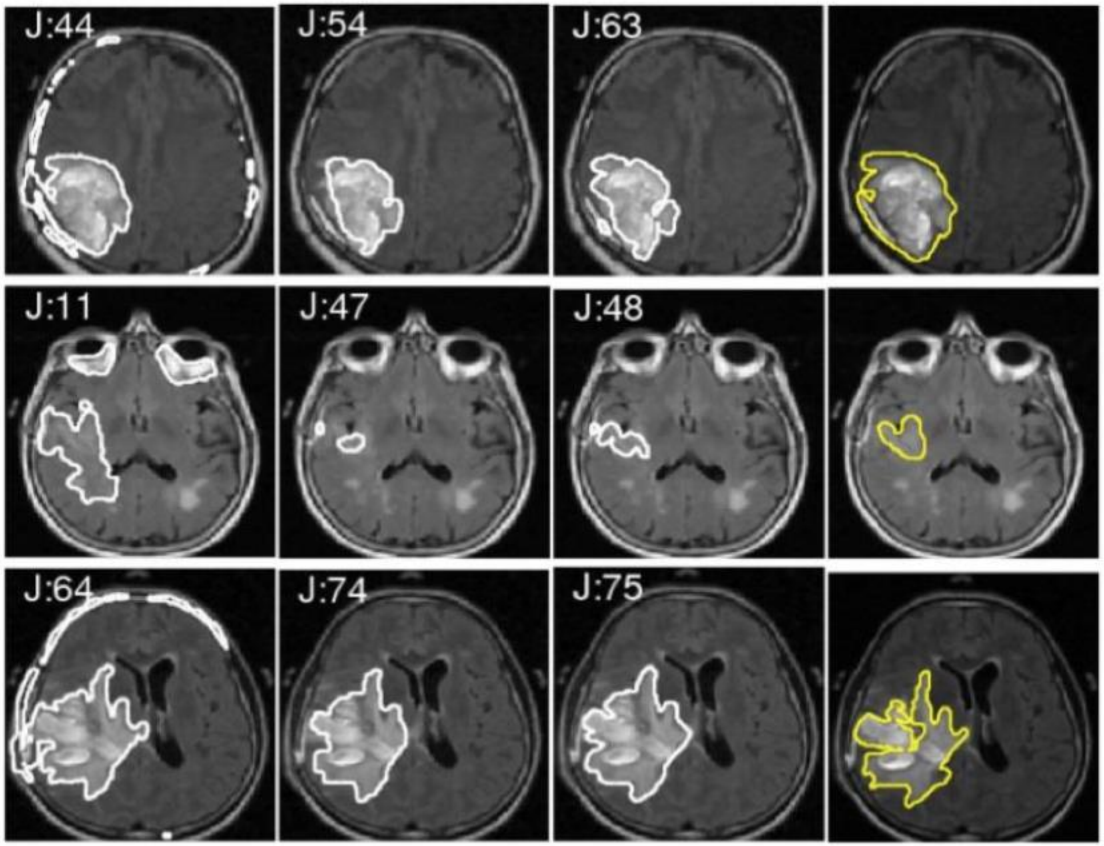
- *Face & Object detection*:





Introduction – Applications

- *Medical Imaging*:
  Tumour segmentation



- *Self driving cars*:

Classification Machine Learning

Classification Machine Learning

Mode of inference: Deductive vs inductive

- Deductive reasoning
  - Logical conclusions (Top-down logic), e.g. using Modus ponens:

    If P*, then* Q

    P

    Therefore Q
  - Example:

    If today is Saturday, then it is the weekend

    Today is Saturday

    Therefore it is the weekend

Classification Machine Learning

Mode of inference: Deductive vs inductive

- Inductive reasoning
  - Generalization from given samples (bottom-up logic)

    The proportion Q *of the sample has attribute* A

    Therefore, the proportion Q *of the population has attribute* A
  - Example:

    A sample of 5 balls from a urn with 20 balls has 4 black balls and 1 white ball

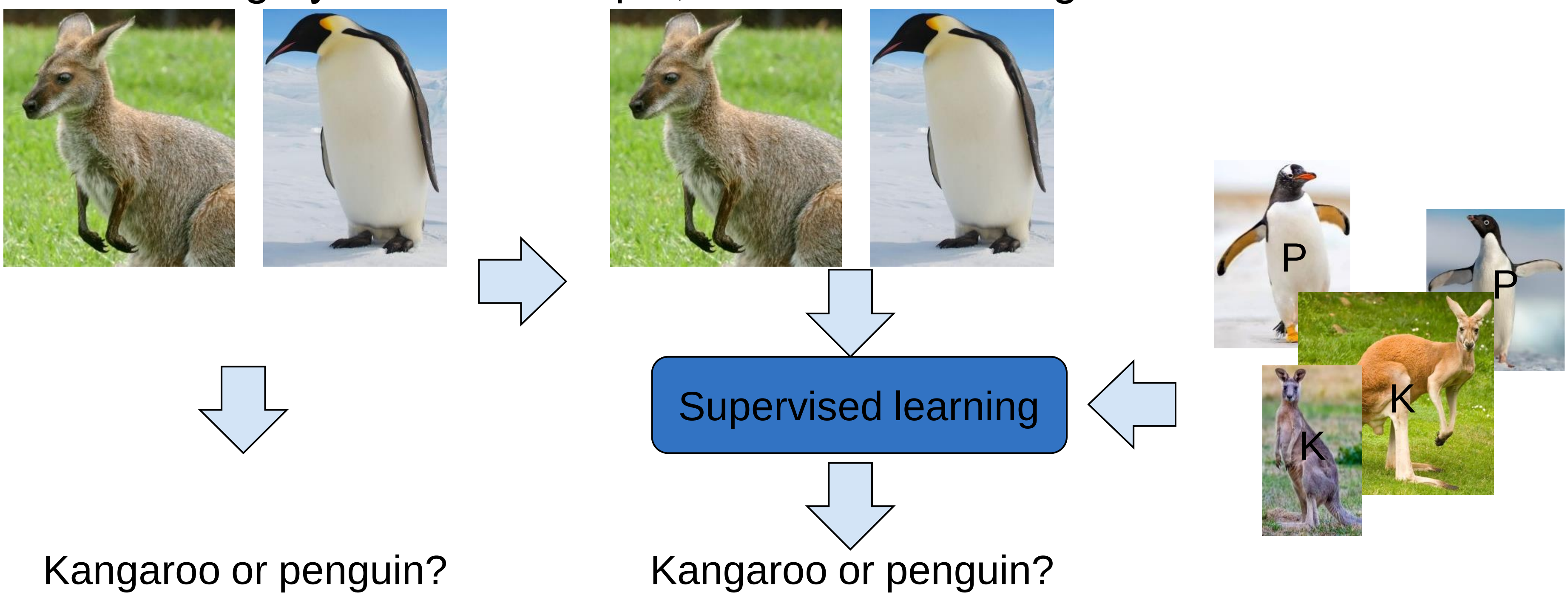
    Therefore, the urn contains 16 black balls and 4 white balls

=> Learning through data analysis

Classification Machine Learning

Category: Supervised learning (concept learning)

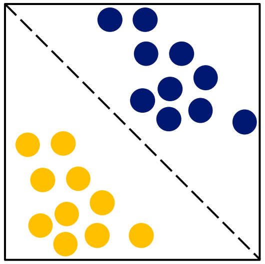
- Learning by labeled example, i.e. we “tell” the algorithm what to learn

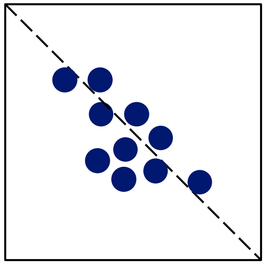


Classification Machine Learning

Category: Supervised learning (concept learning)

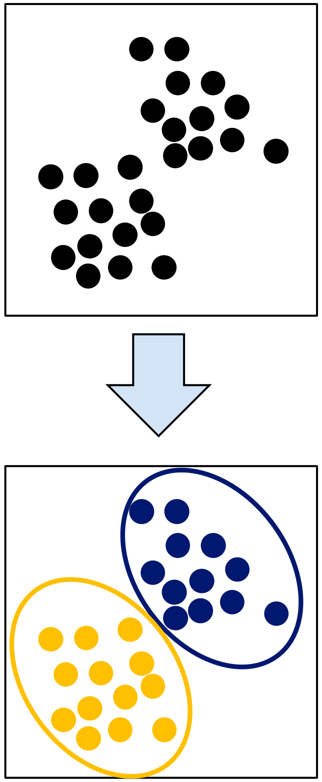
- Learning by labeled example, i.e. we “tell” the algorithm what to learn
- Two forms of output
  - Symbolic
    - Output is a discrete value/category, e.g. kangaroo or penguin
  - Subsymbolic/regression
    - Output is continuous value, e.g. age or temperature





Classification Machine Learning

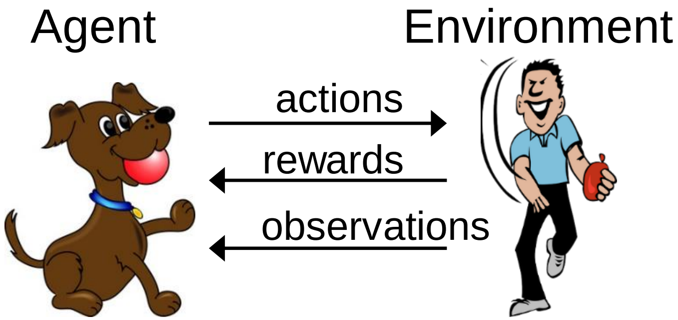
- Category: Unsupervised learning
  - Detect previously unknown patterns in unlabeled data
    - No desired output!
  - Example applications
    - Find anomalies in data, e.g. credit card usage
    - Find clusters, e.g. grouping pictures



Classification Machine Learning

Category: Reinforcement learning

- A software agent learns how to interact (e.g. take actions) in an environment in order to maximize a reward function
  - Environment gives positive or negative rewards based on action(s)
- Examples:
  - Teaching algorithms to play (video) games
  - Robot navigation
  - Modeling dopamine-based learning in the brain

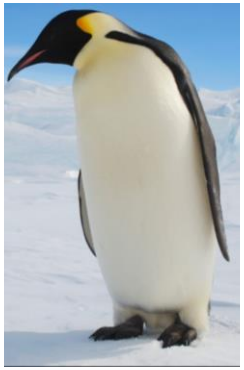


Steps for Machine Learning

- Collect data
  - Most problems require (expert-)labeled data
  - Quantity: Many algorithms require large amounts of data
  - Quality: Is the collected data representative? Are there biases?
- Feature selection
  - Feature: individual measurable property or characteristic of a phenomenon being observed
  - Represents the data
  - Present the data in a more useful manner to help algorithms learn
- Select algorithm
- Train algorithm
- Perform predictions

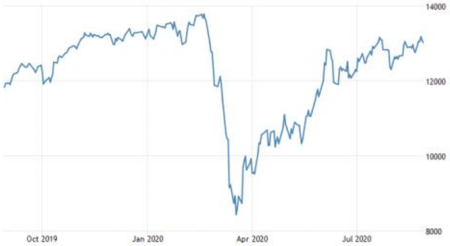
Feature selection

- Examples low level features
    - Images

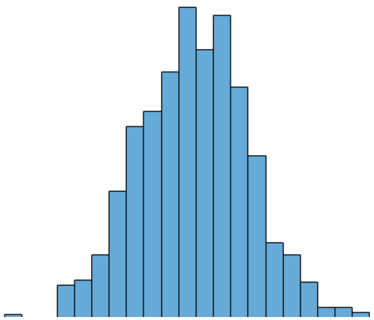
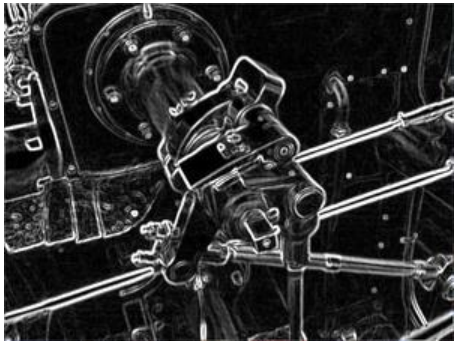
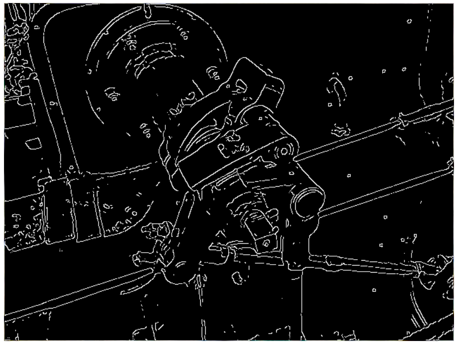


    - Time series

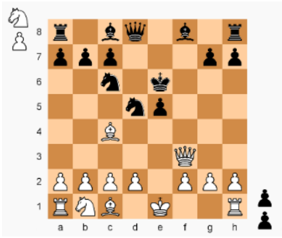



Feature selection

- Examples higher level features
  - Image histogram, gradient, lines, geometric primitives

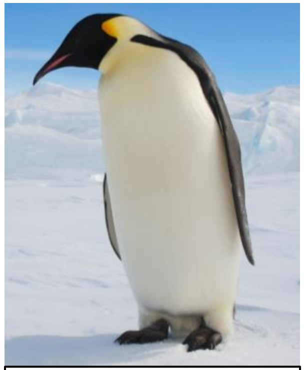
  - Environment states, e.g. state of a game, position of robot on a map, …



Feature selection

- Examples higher level features
  - Attributes

 # legs: 5
Avg. height: 1.8m
Political spectrum:
Communist
Fav. Food:
Schnapps
chocolates



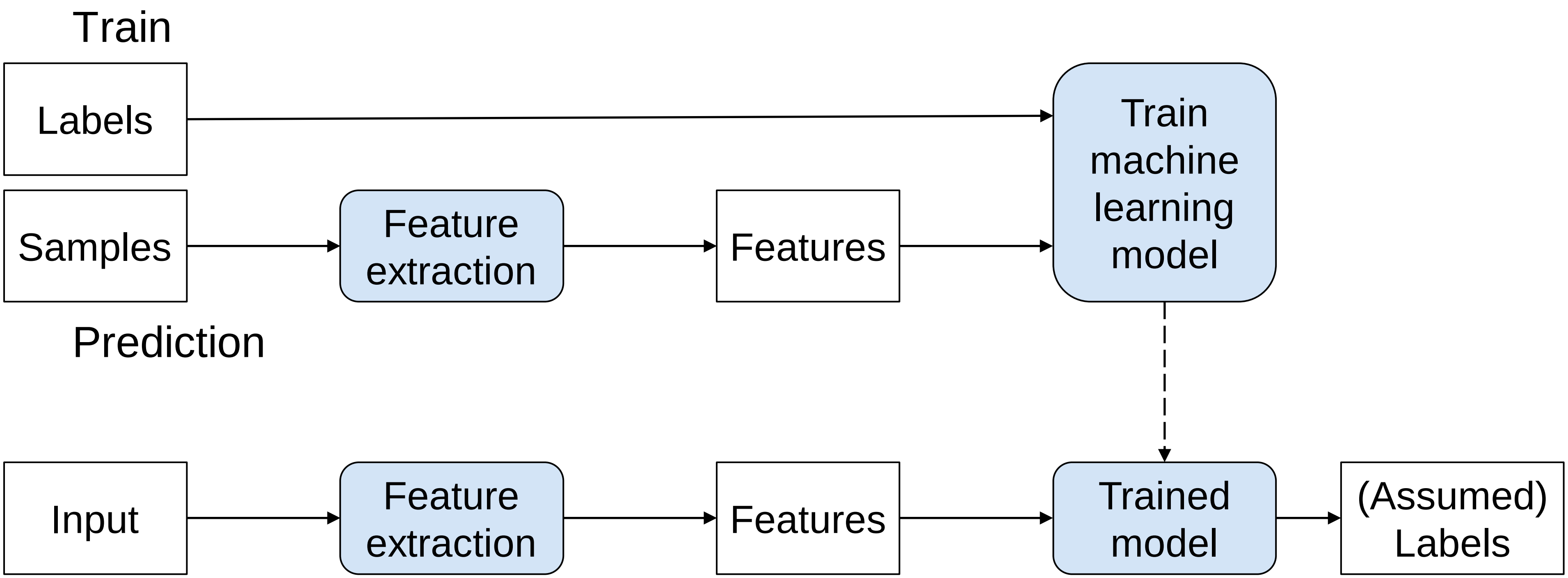
 # legs: 2
Avg. height: 1.1m
Political spectrum:
Capitalist
Fav. Food:
Tea sausage

 # legs: 4
Avg. height: 0.7m
Political spectrum:
Social democrat
Fav. Food:
Eucalyptus

 # legs: 4
Avg. height: 1m
Political spectrum:
Green
Fav. Food:
Steak

 # legs: 4
Avg. height: 0.25m
Political spectrum:
Monarchist
Fav. Food:
Fish

Train and predict



Inductive Machine Learning Algorithms

Supervised

- k-Nearest Neighbor
- Linear classifier
- Bayes classifier
- Decision Trees
- Random Forests
- Support Vector Machine
- Neural Networks (Deep Learning)

Unsupervised

- PCA
- Hierarchical clustering
- k-Means
- DBSCAN

Reinforcement Learning

- SARSA-λ
- Q-Learning

Inductive Machine Learning Algorithms

Supervised

- *k-Nearest Neighbor (in tutorial)*
- **Linear classifier**
- Bayes classifier
- **Decision Trees**
- **Random Forests**
- Support Vector Machine
- *Neural Networks (Deep Learning)*

Unsupervised

- *PCA (in tutorial)*
- Hierarchical clustering
- **k-Means**
- DBSCAN

Reinforcement Learning

- SARSA-λ
- Q-Learning

Supervised Learning

Supervised Learning - Introduction

- "… task of learning a function that maps an input *X* to output *Y* based on example input-output pairs."
  - Trainings data set of *N* instances: $\{X_{1:N}, Y_{1:N}\}$
  - Each input $X_i \in \mathbb{R}^{1xp}$ is a vector with *p* attributes/predictors/features

- (1) Training (learning):

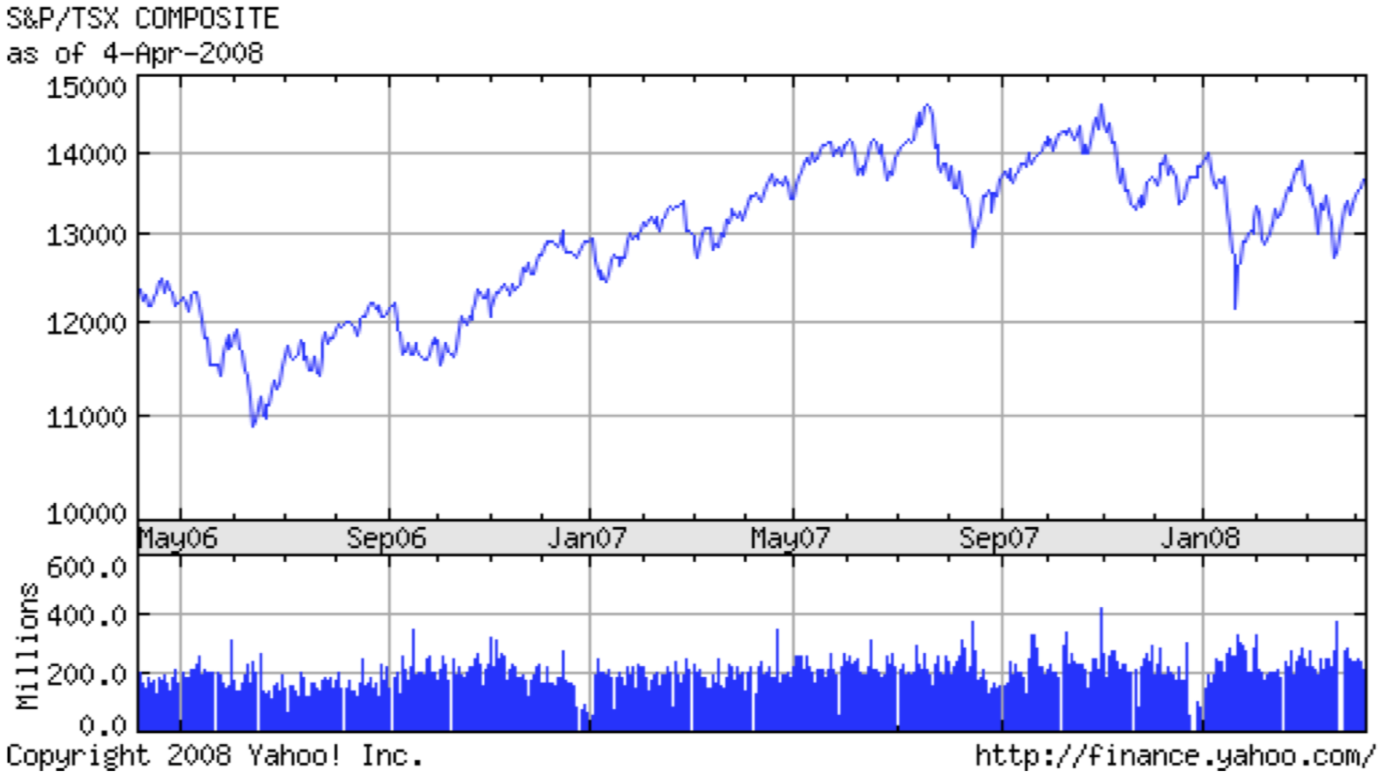
$\{X_{1:N}, Y_{1:N}\}$ → **Learner** → Θ (Model parameter)

- (2) Testing (prediction):

$\{X_{N+1}, \Theta\}$ → **Learner** → $\{\hat{Y}_{N+1}\}$

Supervised Learning – Linear Methods for Regression

- Many real processes can be approximated with linear models
- Linear problems can be solved analytically



Supervised Learning – Linear Methods for Regression

- Input vector $X^T = (X_1, X_2, \ldots, X_p)$ and we want to predict a real-valued output Y
- Linear regression model has the form:

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$$

Unknown parameters/coefficients

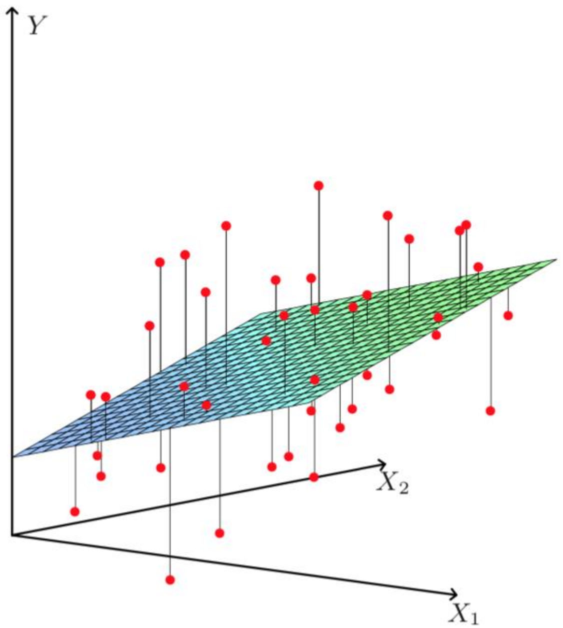
- Input variables X_j can come from different sources:
  - quantitative inputs,
  - transformation of quantitative inputs,
  - coding of the levels of qualitative (class labels) inputs,
  - basis expansions

- No matter of the source of the X_j, model is linear in the parameters

Supervised Learning – Linear Methods for Regression

- Typically, using a training set $(x_1, y_1) \dots (x_N, y_N)$ from which to estimate the parameters β.
- Each $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})^T$ is a factor of measurements of *i*-th case
- Popular estimation method is *least squares*, which minimise the residual sum of squares:

$$\begin{aligned} \mathrm{RSS}(\beta) &= \sum_{i=1}^{N} (y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \Big)^2. \end{aligned}$$



NCT

Supervised Learning – Linear Methods for Classification

- Goal in classification is to take input *x* and assign it to one of K discrete classes C_k
- C_k typically disjoint (unique class membership)

$$\boldsymbol{x}_i =$$  $$\boldsymbol{t}_i = (0,0,0,1,0,0,0,0,0,0)$$

- Training set $\{(x_1,t_1),\ldots,(x_N,t_N)\}$

- Learning problem is to construct a “good” function y(x) from these by:
  - Discriminant function
  - Probabilistic generative models
  - Probabilistic discriminative models

Supervised Learning – Linear Methods for Classification

- Generalised linear model for classification:

$$y(\boldsymbol{x}) = f(\boldsymbol{w}^T\boldsymbol{x} + w_0)$$

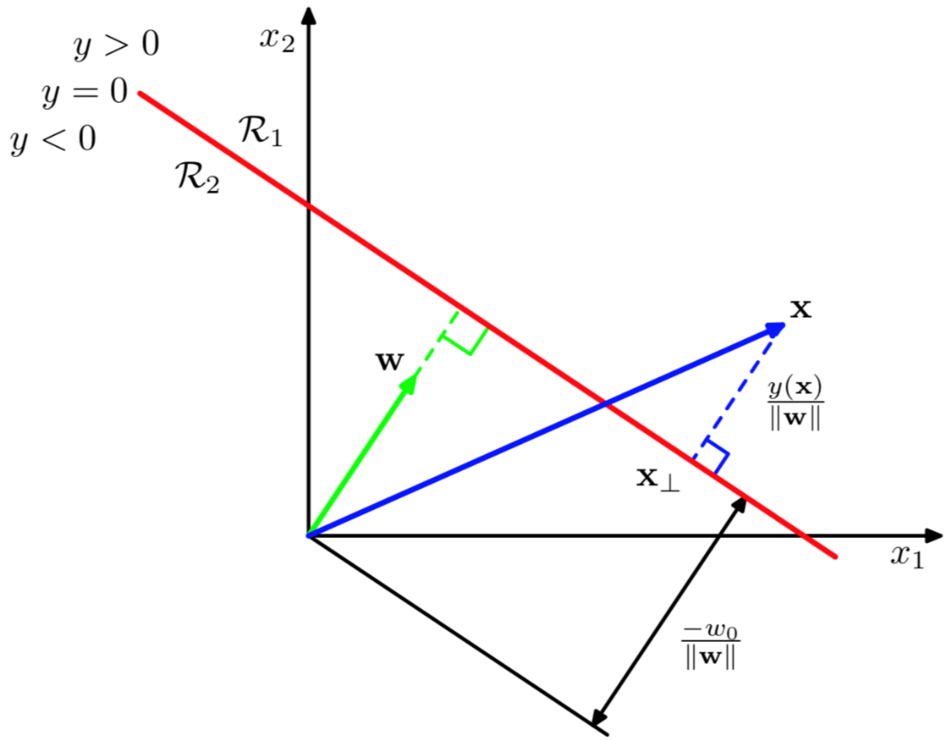
- f(·) is a fixed non-linear function
  - No longer linearity in the parameters
  - More complex analytical and computational properties

$$f(u) = \begin{cases} 1 \text{ if } u \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

- Decision boundary between classes will be a linear function of *x*

Supervised Learning – Linear Methods for Classification

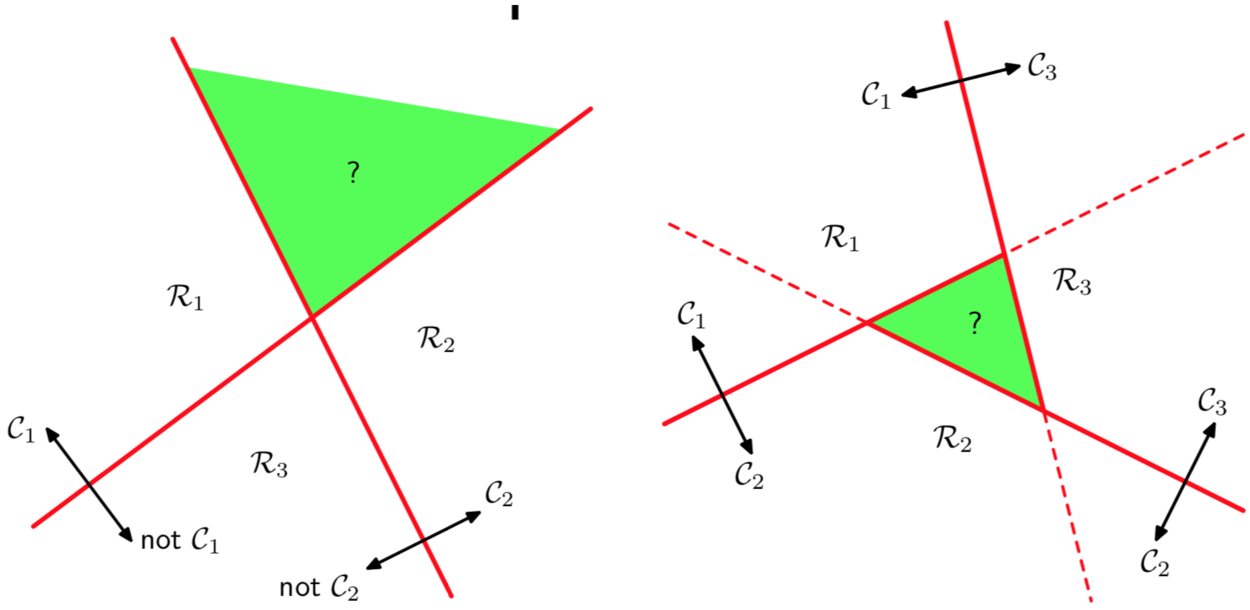
- Discriminant functions for two classes: $y(x) = w^T x + w_0$
  - Takes an input vector x_i and assigns it to on of K classes C_k
  - If $y(x) \geq 0 \rightarrow$ assigned to class C_1 and to class C_2 otherwise



- 2 class problem,
  $t \in \{0,1\}$
- Simple linear discriminant
  $y(x) = w^T x + w_0$
- Apply threshold function to get classification

Supervised Learning – Linear Methods for Classification

- Discriminant functions with for multiple classes:
  - One-versus-the-rest method: build K − 1 classifiers, between C_k and all others
  - One-versus-one method: build K(K − 1)/2 classifiers, between all pairs



Supervised Learning – Linear Methods for Classification

- Learning of *K* discriminant functions:

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}$$

- Assign *x* to class $\arg\max_k y_k(\boldsymbol{x})$

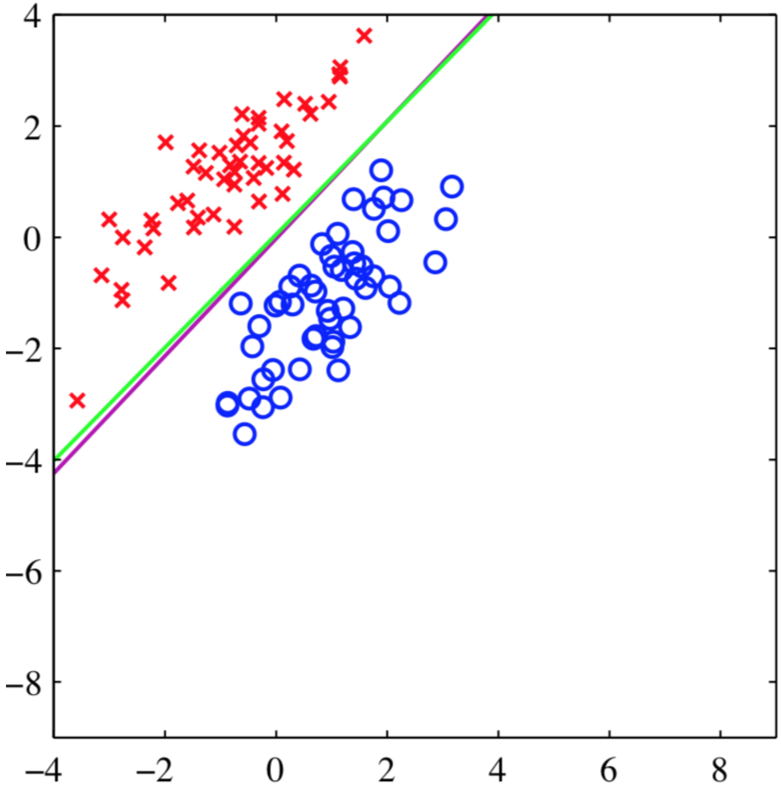
- How do we learn the decision boundaries (w_k,w_{k0})?

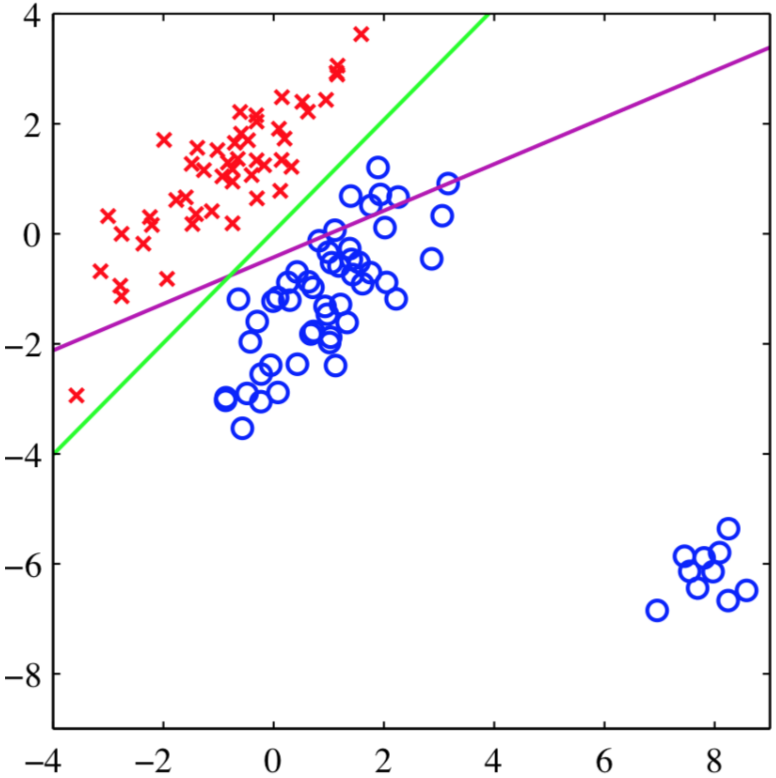
$$E(\boldsymbol{W}) = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} (y_k(\boldsymbol{x}_n) - t_{nk})^2$$

- Use least squares, to find *W* which minimise squared error over all examples and all class labels

Supervised Learning – Linear Methods for Classification

- Problem with least square method:

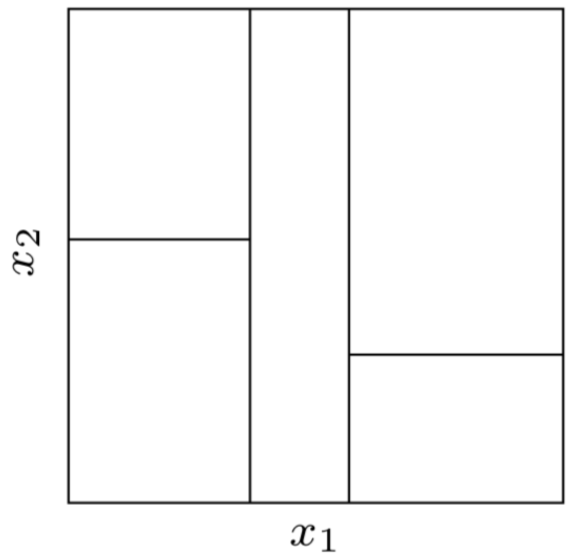




- Points far away from the decision boundary will cause a large error → boundary is moved

Supervised Learning – Tree-Based Methods

- In both regression and classification settings we seek a function y(**x**) which maps the input **x** into a prediction.
- One **flexible** way is to partition the input space into disjoint regions and fit a simple model in each region.



- **Classification:** Majority vote within the region.
- **Regression:** Mean of training data within the region.

Supervised Learning – Tree-Based Methods

- Challenge: find a good partition
- Instead, we use a “greedy” approach: **recursive binary splitting**.

- 1. Select one of the inputs x_j and a cut-point s. Partition the input space into two half-spaces,

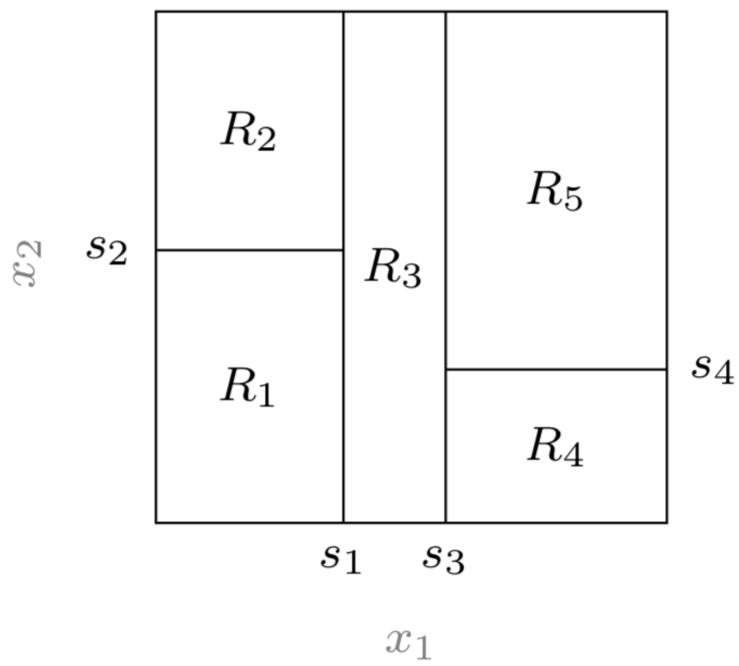
$$\{\mathbf{x}: x_j < s\} \qquad \{\mathbf{x}: x_j \geq s\}$$

- 2. Repeat this splitting for each region until some stopping criterion is met (e.g., no region contains more than 5 training data points).

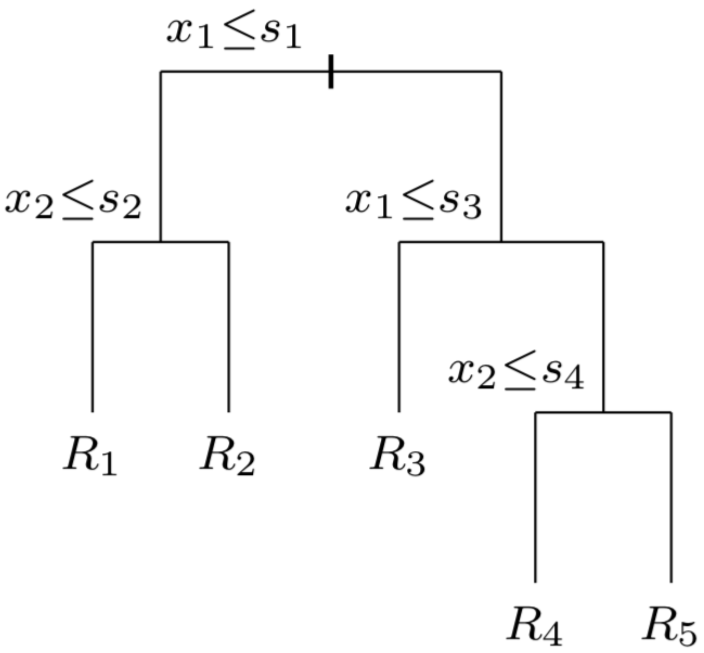
- Generally a metric is used to calculate the benefit of a spilt, e.g. information gain: Information gain = Entropy before split – Entropy after split

Supervised Learning – Tree-Based Methods

Partitioning of input space



Tree representation



Supervised Learning – Classification Tree

- The class prediction for each region is based on the proportion of data points from each class in that region.

$$\widehat{\pi}_{mk} = \frac{1}{n_m} \sum_{i:\mathbf{x}_i \in R_m} \mathbb{I}\{y_i = k\}$$

- Proportion of training observations in the *m*-th region that belong to the *k*-th class.

- Approximation of the class probability:

$$p(y = k \,|\, \mathbf{x}) \approx \sum_{m=1}^{M} \widehat{\pi}_{mk} \mathbb{I}\{\mathbf{x} \in R_m\}$$

Supervised Learning – Improving tree-based Models

- The performance of a tree-based model is often unsatisfactory

- To improve the practical performance:
  - **Pruning** – grow a deep tree which is then pruned into a smaller one
  - **Ensemble methods** – average or combine multiple trees.
    - Bagging and Random Forests
    - Boosted trees

Supervised Learning – Random Forest

- A **random forest** is constructed by bagging, but for each split in each tree only a ***random subset*** of q ≤ p inputs are considered as splitting variables.

- Rule of thumb: q = √p for classification trees and q = p/3 for regression trees.

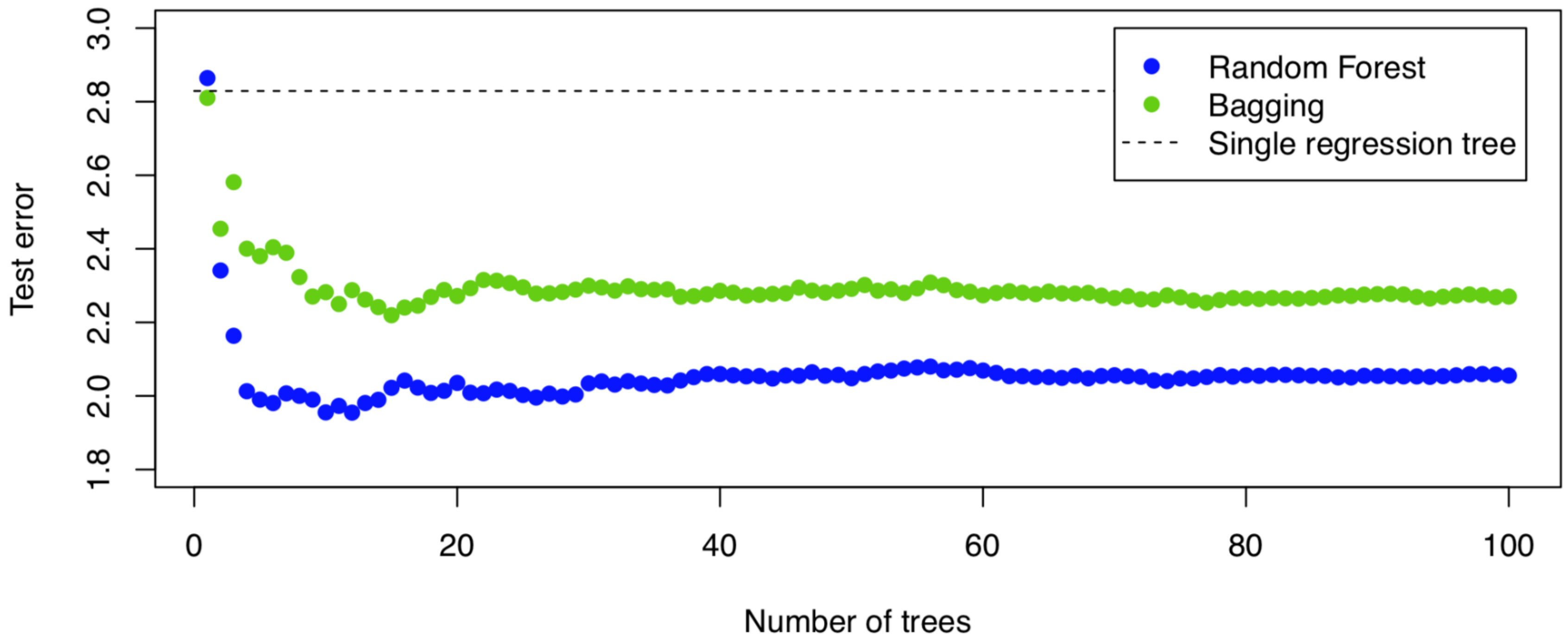
Algorithm Random forest for regression

---

1. For $b = 1$ to B *(can run in parallel)*
   (a) Draw a bootstrap data set $\widetilde{\mathcal{T}}$ of size n from $\mathcal{T}$.
   (b) Grow a regression tree by repeating the following steps until a minimum node size is reached:
      i. Select q out of the p input variables uniformly at random.
      ii. Find the variable x_j among the q selected, and the corresponding split point s, that minimizes the squared error.
      iii. Split the node into two children with $\{x_j \leq s\}$ and $\{x_j > s\}$.
2. Final model is the average the B ensemble members,

$$\widehat{y}_{\star}^{\text{rf}} = \frac{1}{B}\sum_{b=1}^{B} \widetilde{y}_{\star}^{b}.$$

Supervised Learning – Random Forest



Unsupervised Learning

Unsupervised Learning – Introduction

- Supervised: We are given input/output samples (X, y) which we relate with a function y = f(X).
- We would like to "learn" f, and evaluate it on new data.

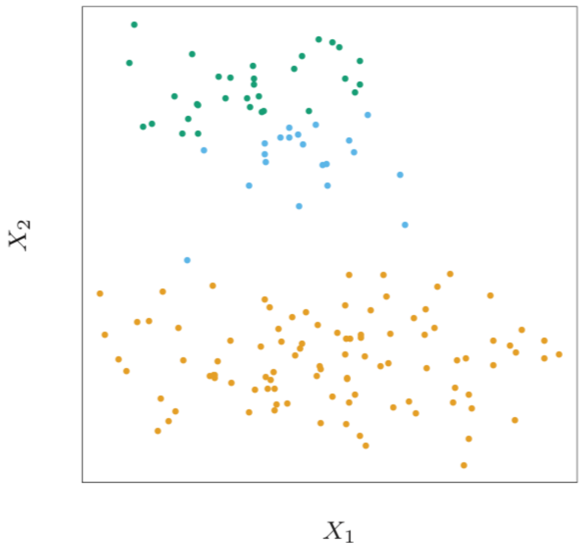
- Types:
  - Classification: y is discrete (e.g. class labels).
  - Regression: y is continuous, e.g. linear regression.

- Unsupervised: Given only samples X of the data, we compute a function f such that y = f(X) is "simpler".
  - y is discrete: Clustering
  - y is continuous: Matrix factorisation, Kalman filtering, unsupervised neural networks

- Example: Clustering for segmentation
  - Break an image into regions of points with similar features

Unsupervised Learning – Cluster Analysis

- Identifying groups, or clusters of data points
- Objects within each cluster are more closely related to one another than objects assigned to different clusters
- Object can be described by a set of measurements, or by its relation to other objects



Unsupervised Learning – Cluster Analysis

- To measure the relation between objects similarities or dissimilarities can be used

- Most often we have measurements x_{ij} for i = 1,2, …,N , on variables j = 1,2,…,p (attributes/features)

- Dissimilarity between objects i and i′ is:

$$D(x_i, x_{i'}) = \sum_{j=1}^{p} d_j(x_{ij}, x_{i'j})$$

- Common choice is squared distance:

$$d_j(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$$

Unsupervised Learning – Cluster Analysis

- Further dissimilarities-based methods:

  - Absolute differences:

$$d(x_i, x_{i'}) = l(|x_i - x_{i'}|).$$

  - Correlation:

$$\rho(x_i, x_{i'}) = \frac{\sum_j (x_{ij} - \bar{x}_i)(x_{i'j} - \bar{x}_{i'})}{\sqrt{\sum_j (x_{ij} - \bar{x}_i)^2 \sum_j (x_{i'j} - \bar{x}_{i'})^2}}$$

Unsupervised Learning – Clustering Algorithms

- Centroid-based clustering:
  - *K*-means
  - Kernel *k*-means
  - Fuzzy *C*-means
- Hierarchical clustering:
  - Agglomerative: bottom-up
  - Divisive: top-down
- Distribution-based clustering:
  - Gaussian mixture model (GMM)
- Density-based clustering:
  - Mean-shift

Unsupervised Learning – *K*-means Clustering

- The basic idea is to describe each cluster by its mean value.

- The goal of k-means is to assign data to clusters and define these clusters with their means.

- An iterative clustering algorithm:
  - Initialize: Pick *K* random points as cluster centres
  - Alternate:
    - 1. Assign data points to closest cluster centre
    - 2. Change the cluster centre to the average of its assigned points

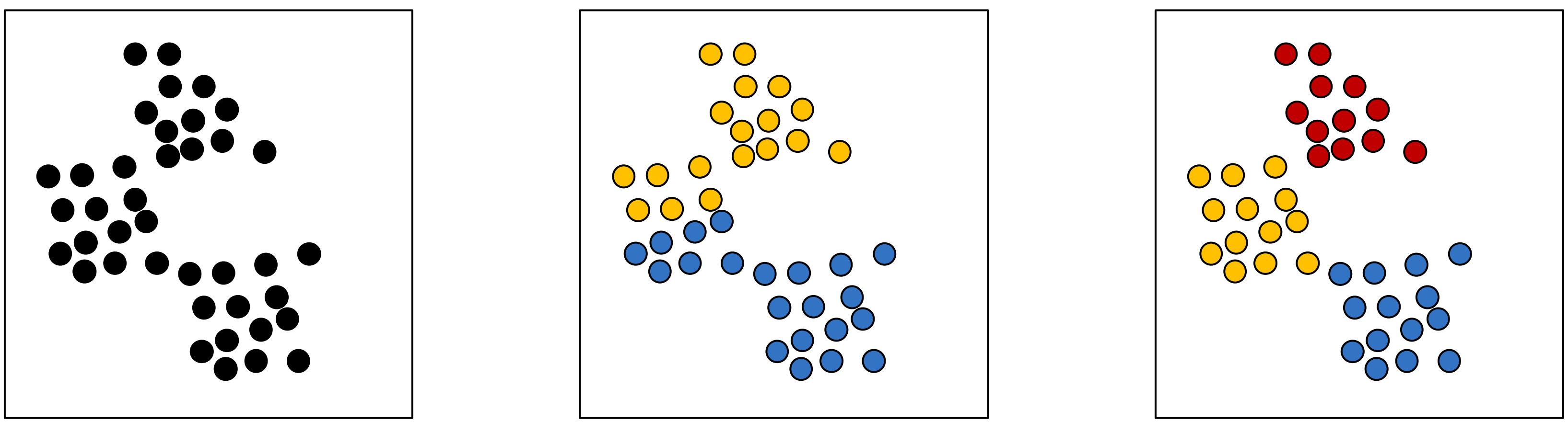
- Stop when no points' assignments change

K-Means

Problem

- Given: *N* data points
- Wanted: Divide data points into *K* clusters

K = 1 K = 2 K = 3

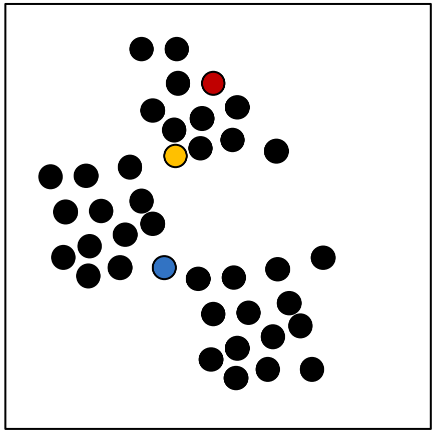


K-Means

Approach

- Select *K* initial points (centroids)

K = 3

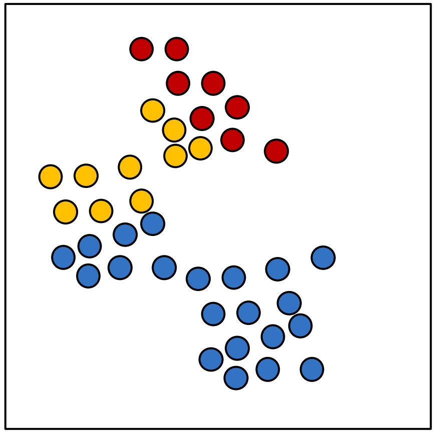


K-Means

Approach

- Select *K* initial points (centroids)
- Assign each point to the closest centroid

K = 3

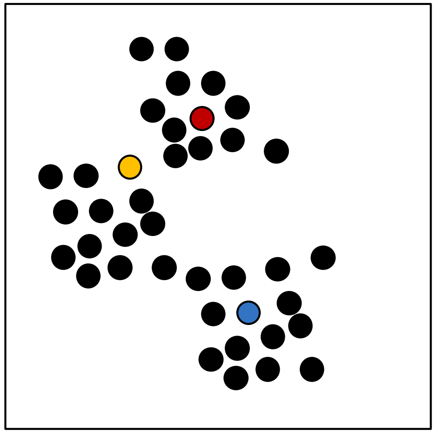


K-Means

Approach

- Select *K* initial points (centroids)
- Assign each point to the closest centroid
- Compute new centroid for each cluster

K = 3

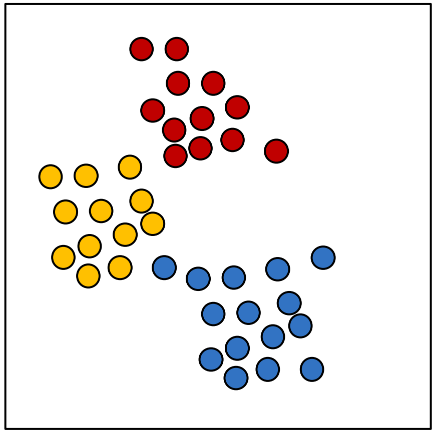


K-Means

Approach

- Select *K* initial points (centroids)
- Assign each point to the closest centroid
- Compute new centroid for each cluster
- Repeat (for a fixed amount of times or until convergence)

K = 3

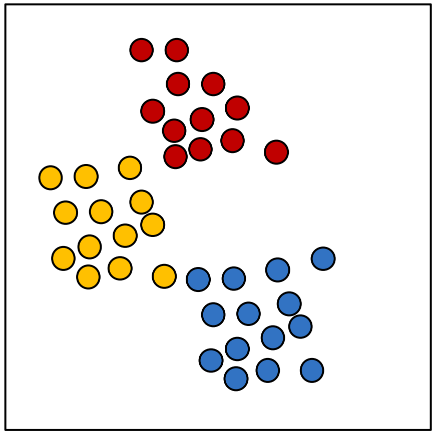


K-Means

Summary

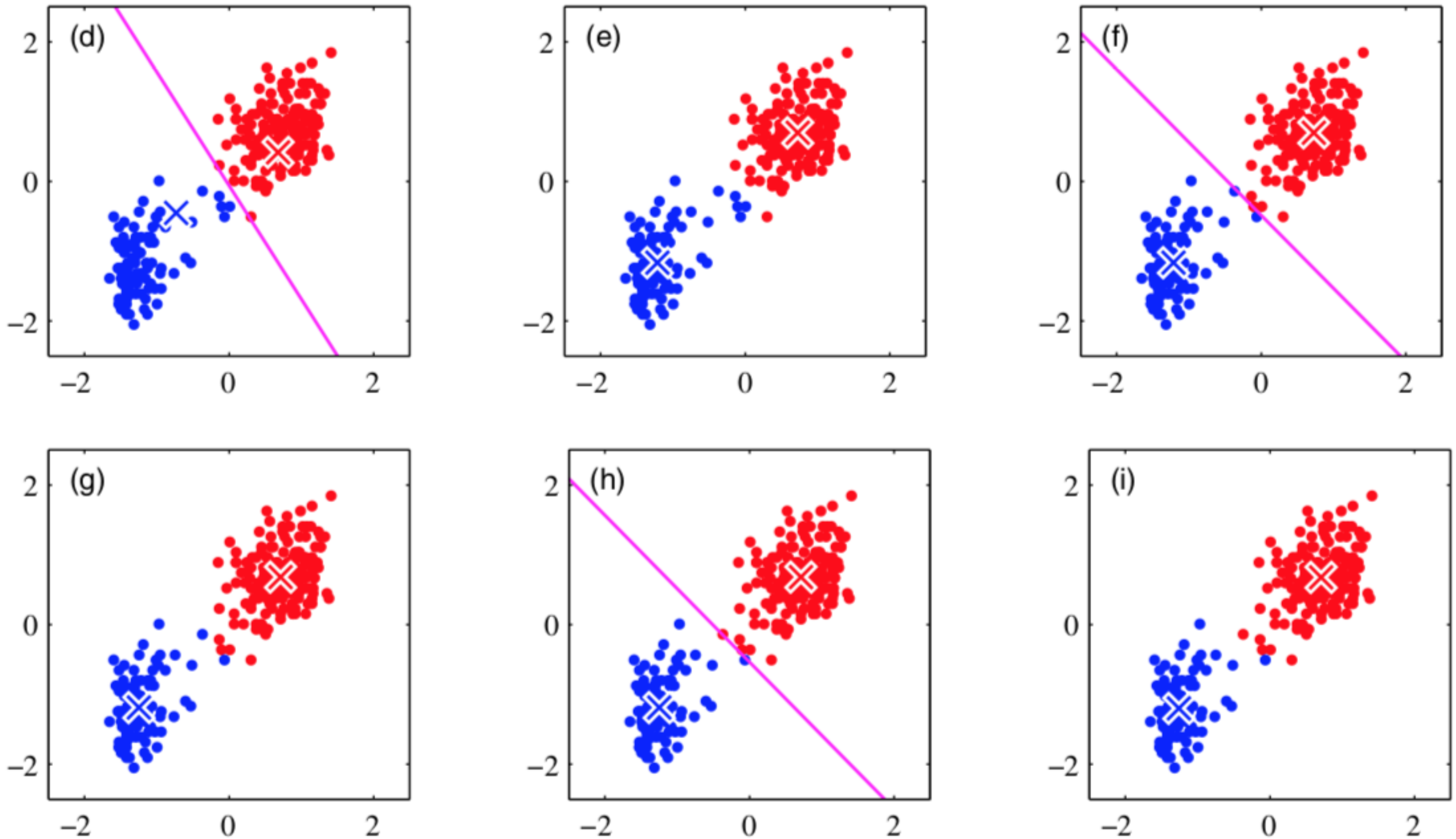
- Simple method for dividing data into clusters
- No supervision required
- Results depended on choice of initial points
- => not guaranteed to converge into global optimum

K = 3



Unsupervised Learning – *K*-means Clustering

Repeat until convergence:



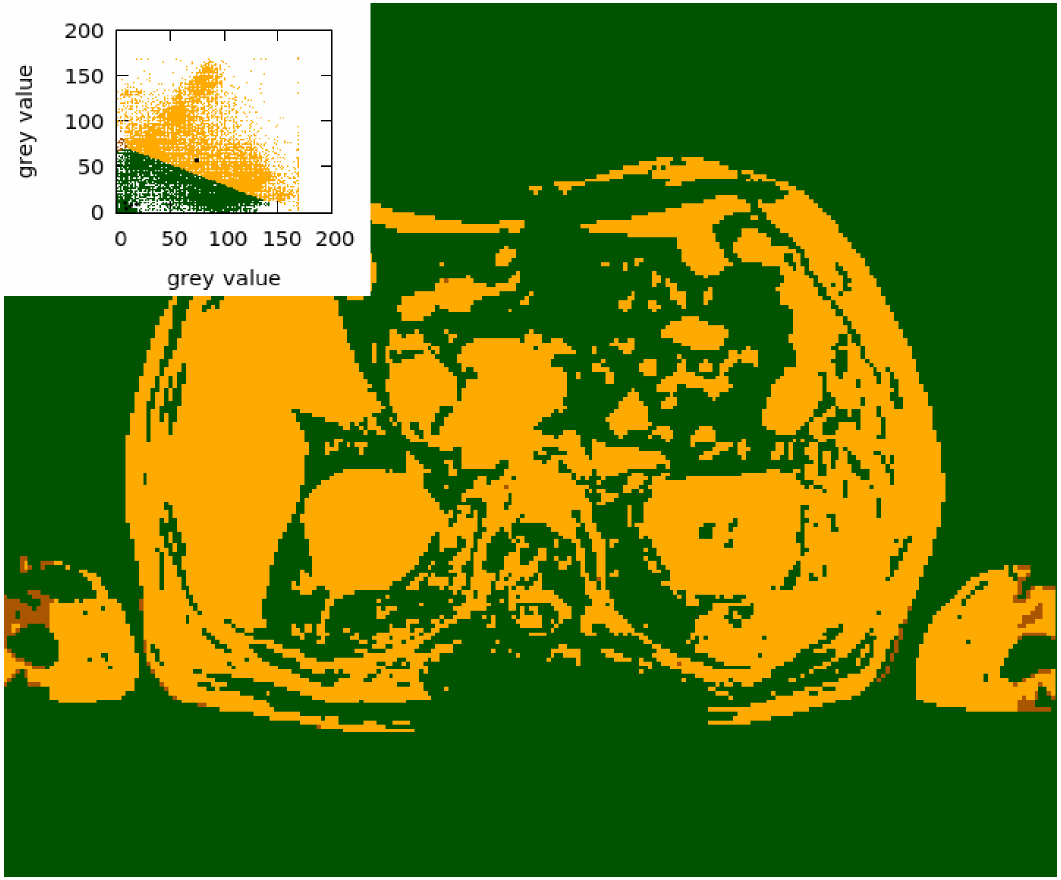
Unsupervised Learning – *K*-means Clustering

Segmentation example:

K=2 | K=3 | Original

Unsupervised Learning – *K*-means Clustering

Segmentation example:



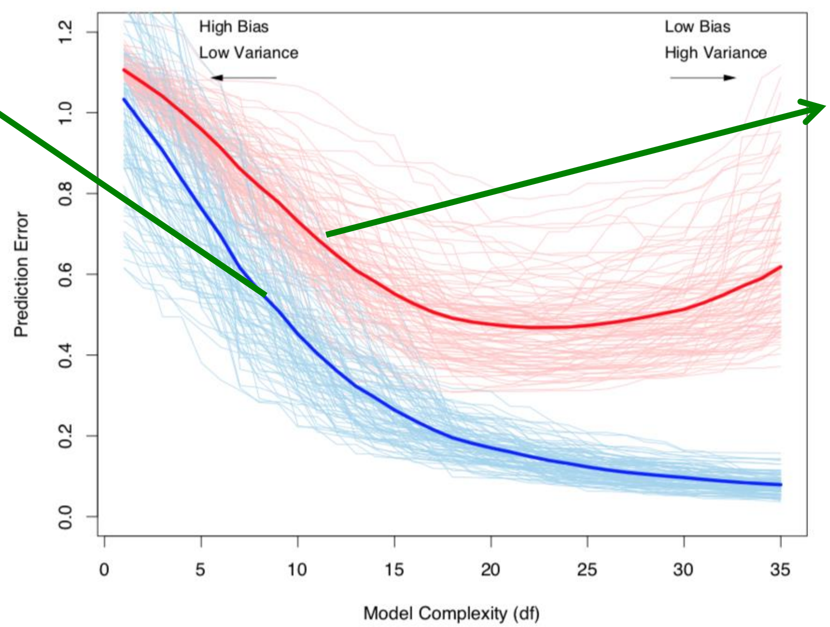
Model Assessment and Selection

Model Assessment and Selection

- Assessment of this performance is extremely important in practice:
  - It guides the choice of learning method or model
  - Measure of the quality of the chosen model

- Generalisation performance of a learning method relates to its prediction capability on independent test data

Train error



Test error

Model Assessment and Selection

- Tuning parameters α varies the model complexity
- Find value of α which minimise the average test error

- Objectives:
  - Model selection: estimating the performance of different models in order to choose the best one
  - Model assessment: estimating the prediction error on new data using the chosen model

Training	Validation	Test

Model Assessment and Selection

- *K*-Fold Cross-Validation:
  - Split data into K roughly equal-sized parts

- Example for *K*=5:
  - Fit model to *K*-1 parts and calculate the prediction error using *k*-th part

1	2	3	4	5
Train	Train	Validation	Train	Train

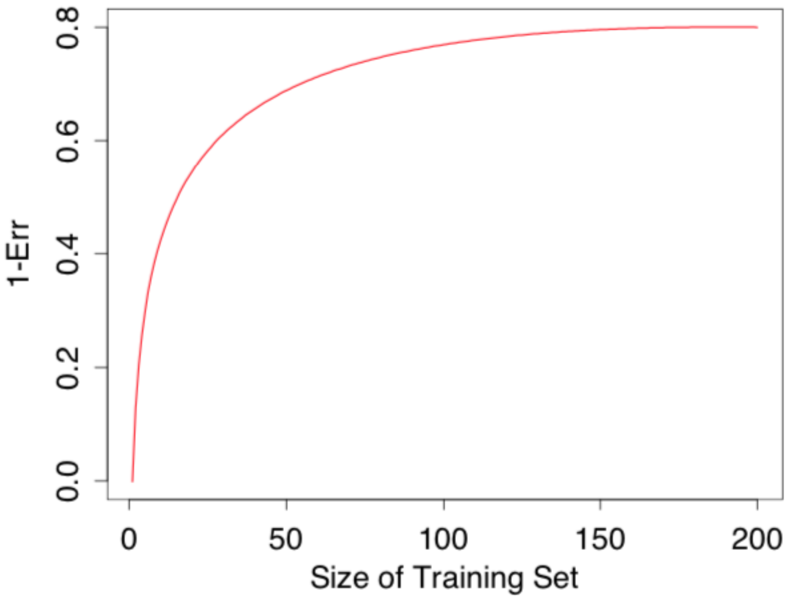
- Cross-validation estimate of prediction error:

$$\mathrm{CV}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i)).$$

Model Assessment and Selection

- What value should we choose for *K*?
  - K=N: Unbiased for the true (expected) prediction error but can have high variance AND high computational burden
  - K=5: Has lower variance but bias could be a problem

- Hypothetical "learning curve"
  - Only small benefit if increasing the number of observation from 100 to 200
  - Trainings set N=50: 5-Fold CV leads to an underestimation



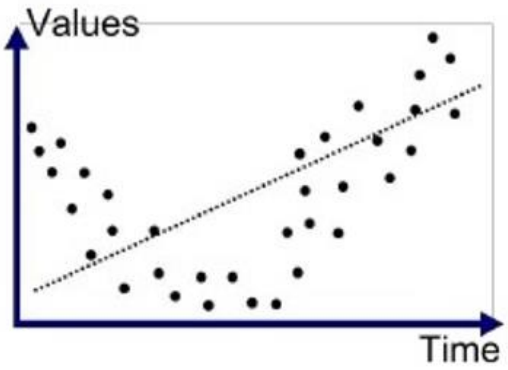
- For practice:
  - 5-Fold or 10-Fold CV are recommended as a good compromise

Model Assessment and Selection

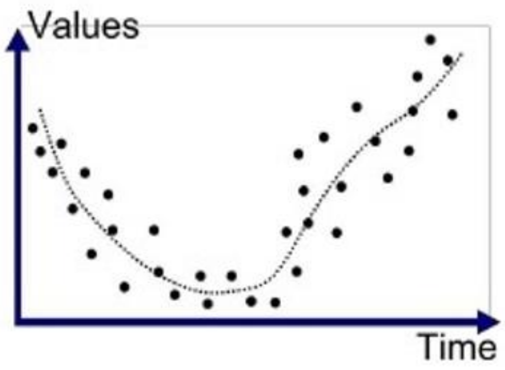
- Right Way to do Cross-validation:
  - Divide samples into K cross-validation folds at random
  - For each fold *k*=1,2,…,K
    - Find a subset of „good“ predictors using all the samples expect those in fold *k*
    - Using this subset and build classifier on all samples except those in fold *k*
    - Use this classifier to perform prediction for the samples in fold *k*

Model Assessment and Selection

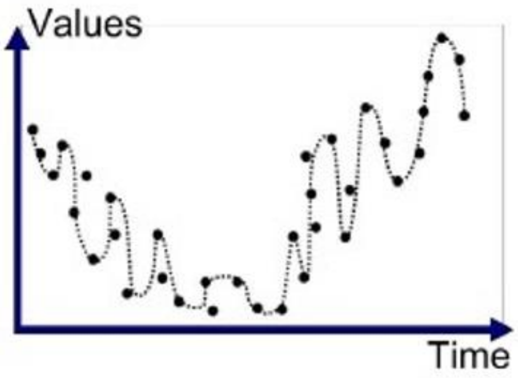
- Underfitting
  - Model is too simple to fit the given data
- Overfitting
  - Model is not able to generalize, as it also learns noise and outliers



Underfitted



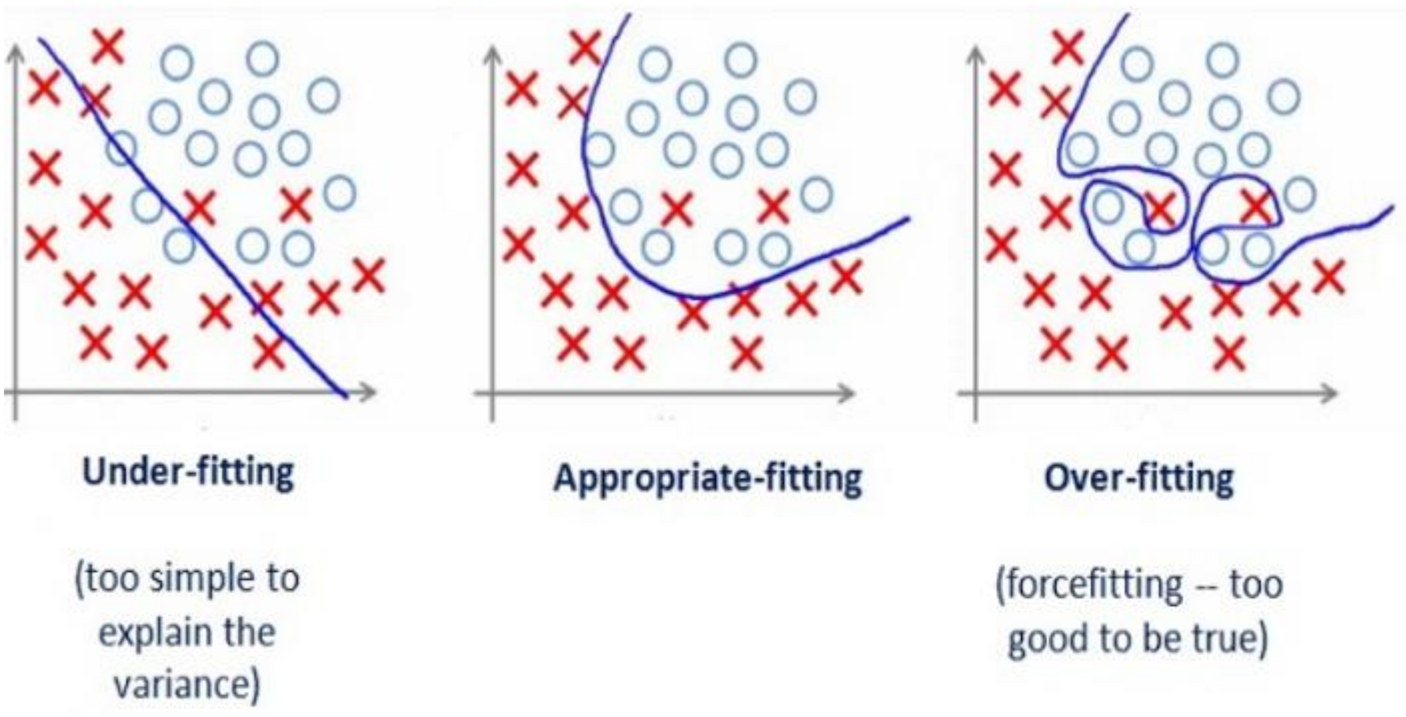
Good Fit/Robust



Overfitted

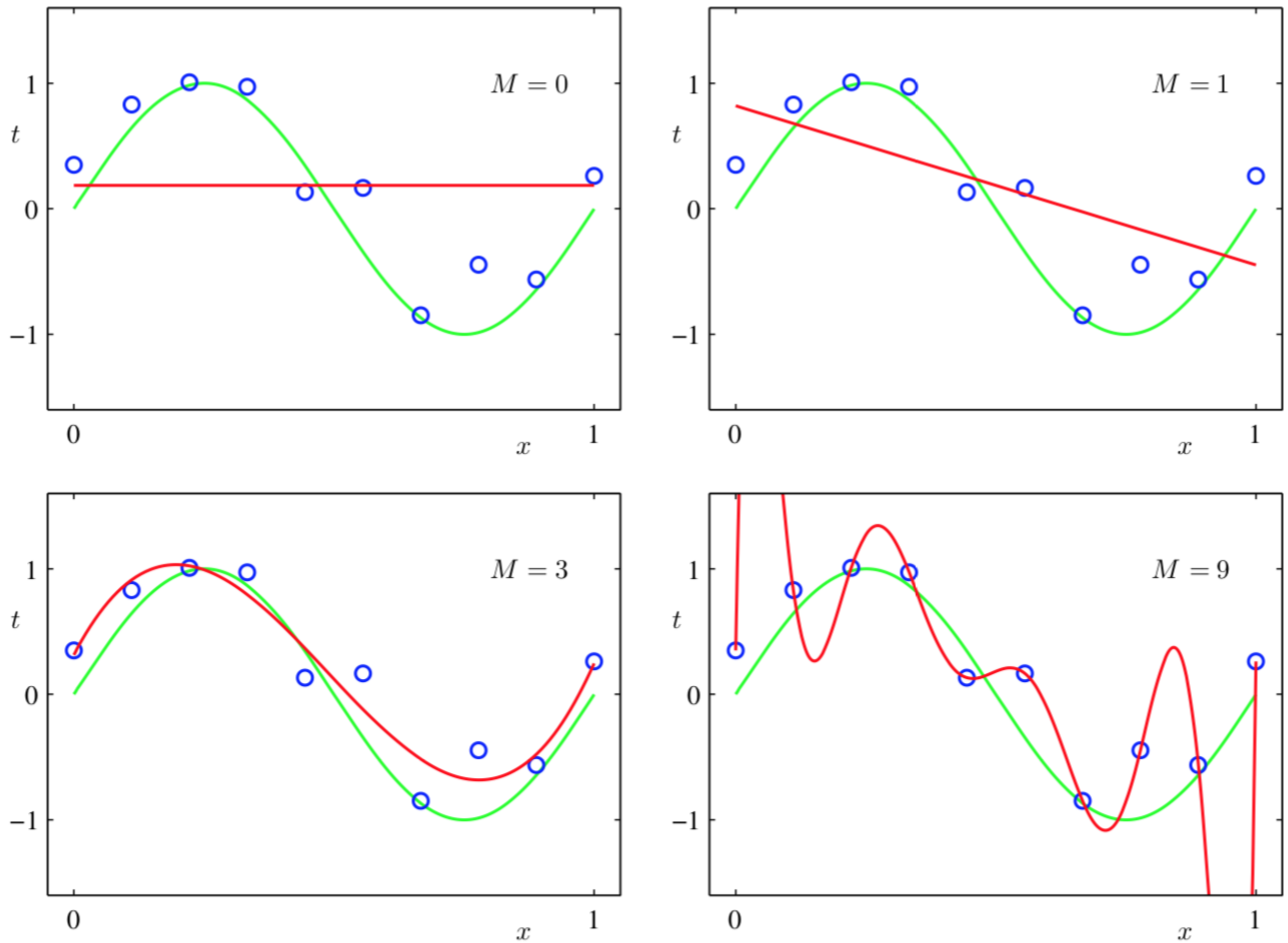
Model Assessment and Selection

- Underfitting
  - Model is too simple to fit the given data
- Overfitting
  - Model is not able to generalize, as it also learns noise and outliers



Model Assessment and Selection

- Polynomial function: $y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$



Literature

- Bishop et al., Pattern Recognition and Machine Learning
- Barber et al., Bayesian Reasoning and Machine Learning
- Hastie et al. The Elements of Statistical Learning