Machine Learning
Introduction

Prof. Matthias Hein

Machine Learning Group
Department of Mathematics and Computer Science
Saarland University, Saarbrücken, Germany

Lecture 1, 18.10.2013

Organization of the lecture

Core course, 4+2 hours, 9 credit points

- Organization: Shyam Sundar Rangapuram
- Exercises:
  - weekly exercises, theoretical and practical work (roughly alternating),
  - practical exercises will be in Matlab,
  - 50% of the points in the exercises are needed to take part in the exams.
- Exams:
  - End-term: to be done
  - Re-exam: to be done
- Grading: An exam is passed if you get at least 50% of the points. You have to pass one exam. The grading is based on the best passed exam.

Roadmap of the lecture

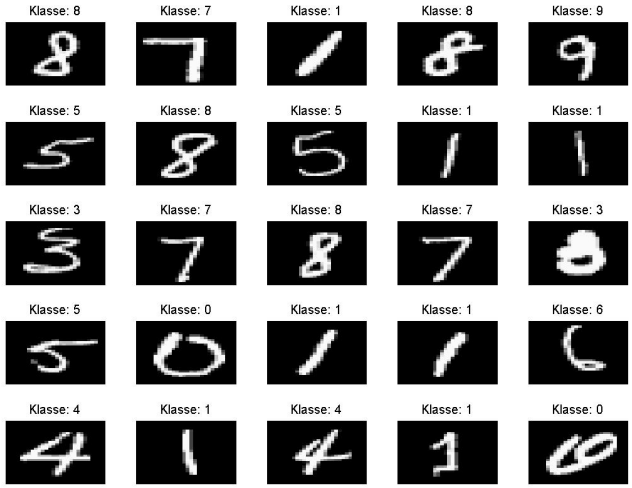
- Introduction to machine learning
- Recap of basic probability notions
- Bayesian decision theory
- Linear methods for regression and classification (SVM etc.) and quick intro into (convex) optimization
- Kernel methods (... going nonlinear)
- Evaluation/Comparison of classifiers, Model selection
- Feature selection
- Boosting and decision trees, Neural networks, prototype methods
- Semi-supervised learning
- Unsupervised learning (Clustering, Dimensionality Reduction)
- Large Scale Learning (Online Learning/Stochastic Gradient Descent)
- Statistical learning theory

Roadmap for today

- What is machine learning ?
- Inductive Inference and machine learning
- Types of learning
- Statistical learning
- Discriminative versus generative learning
- Challenges: Curse of dimensionality and over- versus underfitting
- Is there a best learning algorithm ?

What is machine learning ?

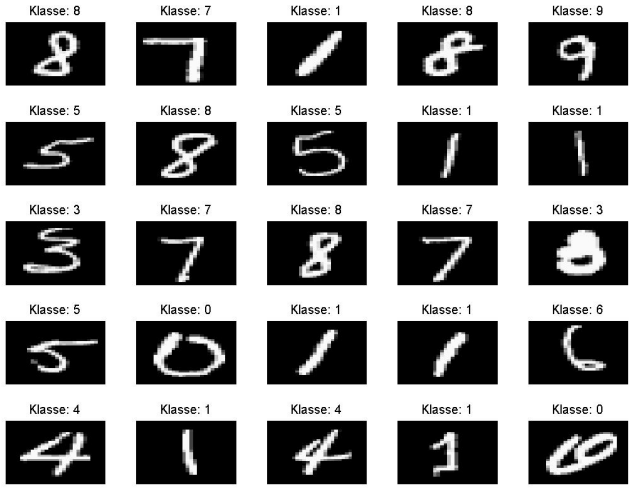
Learning the terminology of machine learning with an example:



input	pixel representation of the digits (image is in $\mathbb{R}^{28\times 28} = \mathbb{R}^{784}$)
feature	one specific property of the input (single dimension of the input)
output	class label (ten digits) in $\{0, 1, \ldots, 9\} \Longrightarrow$ multi-class problem
classifier	a function from input to output, that is $f : \mathbb{R}^{784} \to \{0, 1, \ldots, 9\}$.

What is machine learning ?

Learning the terminology of machine learning with an example:



training	construction of the classifier (usually optimization problem)
testing	count errors on unseen cases
generalization	classifier predicts well on unseen cases
model	parameterized function class in which the classifier is chosen

Inference I

In the **natural sciences**, we are doing **inference**.

We differentiate between two types:

- **Inductive inference:** Learning general principles from observations.
- **Deductive inference:** Deriving specific assertions from general principles.

Inference II

Inductive inference

Inductive inference is at the heart of all natural sciences.

General steps

1. Collecting observations/data.
2. Construction of a model.
3. Prediction.

Falsification

Inductive results can only be **falsified** but not **verified** !

Machine Learning tries to **automate** the process of inductive inference.

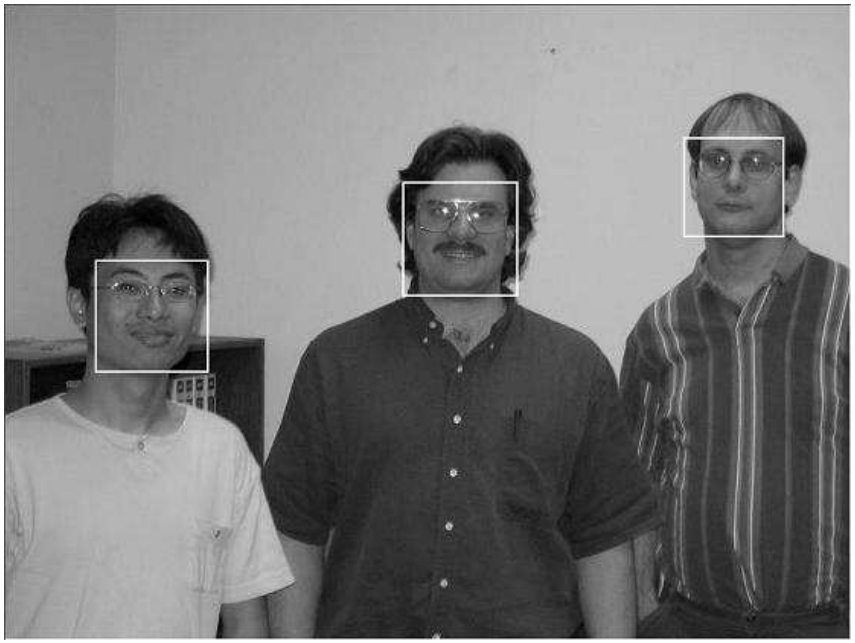
Applications of machine learning

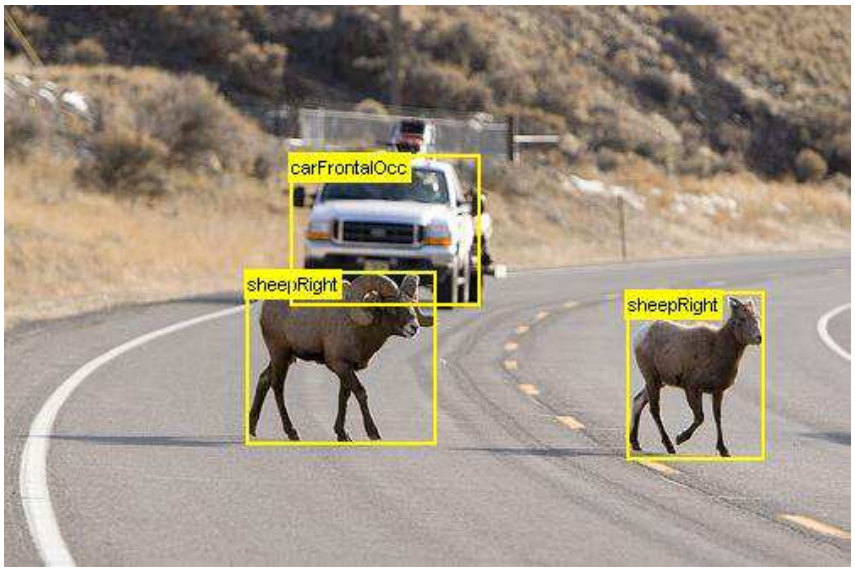
Most important application areas:

- bioinformatics,
- computer vision/image processing/computer graphics/robotics,
- information retrieval/collaborative filtering,
- natural language processing
- economics,
- other: spam filter/intrusion detection,
- machine learning in computer games and software engineering.

Ever more data is collected in different areas. Humans cannot analyze it.
$\Longrightarrow$ **Increasing demand for machine learning.**

Applications I - Computer Vision





Object categorization in computer vision:

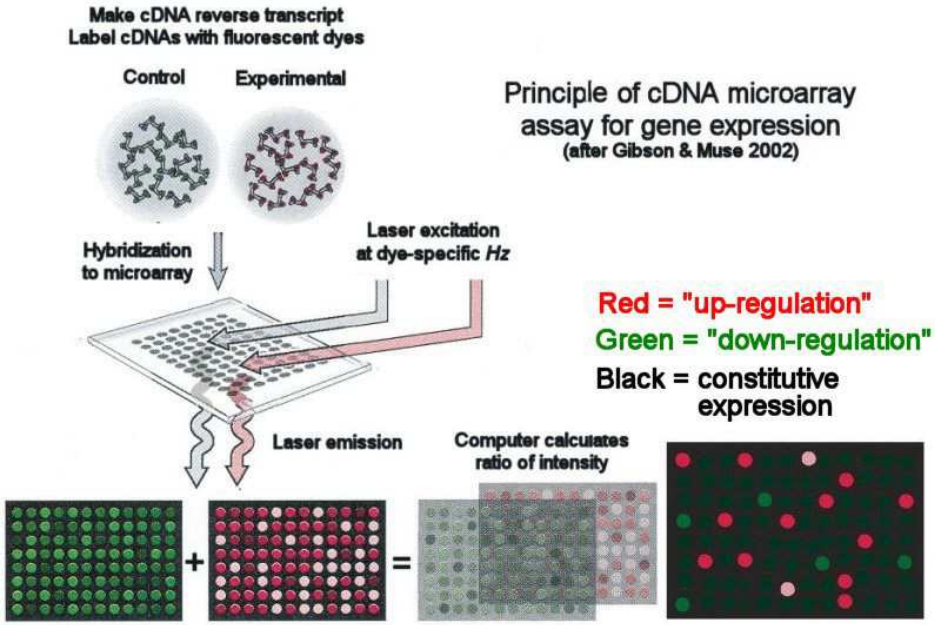
- Face detection (now in digital cameras - works well as long as you look straight into the camera)
- General Object Categorization (Competitions today with more than 20 classes).

Applications II - Robotics

Machine learning for autonomous driving:

- Stanford won the DARPA Grand Challenge 2005 (Left: the navigation),
- The Grand Urban challenge 2007 has been won by CMU (Right: crashes of other cars).
- S-class of Mercedes features semi-automatic driving 2013

Applications III - Bioinformatics



Bioinformatics:

- diagnosis of diseases etc. using sequencing data

User-guided image segmentation - Example of Semisupervised Learning:

Left: Input Image with user labels, Right: Image segmentation

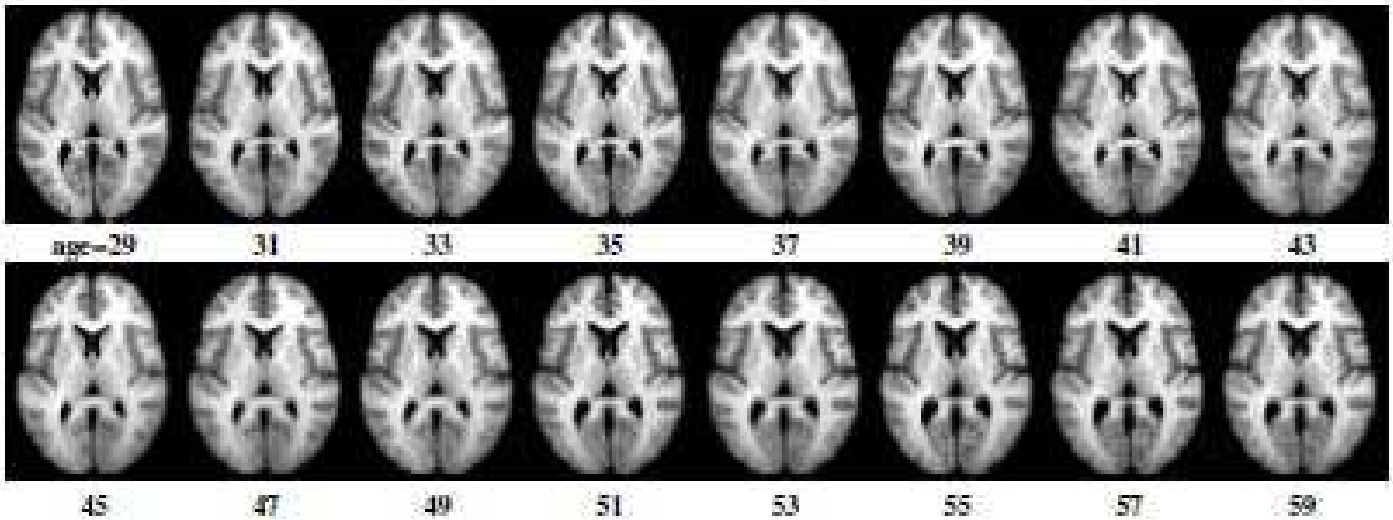
Regression

Regression:

Learning of a general function $f : M \to N$.

Examples

- Prediction of temperature, wind direction in weather forecast,
- Prediction of whole voxel images,



How does the brain change when one gets older ?

Types of learning

We distinguish between three main types of learning:

- **supervised** learning,
- **semi-supervised** learning,
- **unsupervised** learning.

In the following:

$\mathcal{X}$ is the **input space**, X_i are the training inputs,

$\mathcal{Y}$ is the **output space**, Y_i are the training outputs.

Supervised Learning

Supervised Learning:
Given n observations $T = (X_i, Y_i)_{i=1}^n$ construct function $f_n : \mathcal{X} \to \mathcal{Y}$.

- output space $\mathcal{Y}$ discrete $\Longrightarrow$ **classification**.
- output space $\mathcal{Y} = \mathbb{R}$ or $\mathcal{Y} = \mathbb{R}^d \Longrightarrow$ **(multivariate) Regression**.
- general output space $\mathcal{Y} \Longrightarrow$ **Learning with structured output**.

Unsupervised learning

Unsupervised Learning:
Given a set of input points $(X_i)_{i=1}^n$:

- **Clustering:** Construction of a grouping of the points into sets of *similar* points, the so called *clusters*.
- **Density Estimation:** Estimation of the distribution of the input points over the input space $\mathcal{X}$. Related problem **outlier detection**.
- **Dimensionality Reduction:** Construction of a mapping $\phi : \mathcal{X} \to \mathbb{R}^m$, where the dimensionality m of the target space is usually much smaller than that of the input space $\mathcal{X}$. Generally, the mapping should preserve properties of the input space $\mathcal{X}$ e.g. distances.

Batch, online and active learning

- **batch**: all training data is given at once and classifier is also trained only once,
- **online**: the training data arrives sequentially which leads then to sequential updates of the learning rule.
- **active learning**: a variant of (semi)-supervised learning. The learning algorithm can actively query some input points to be labeled.

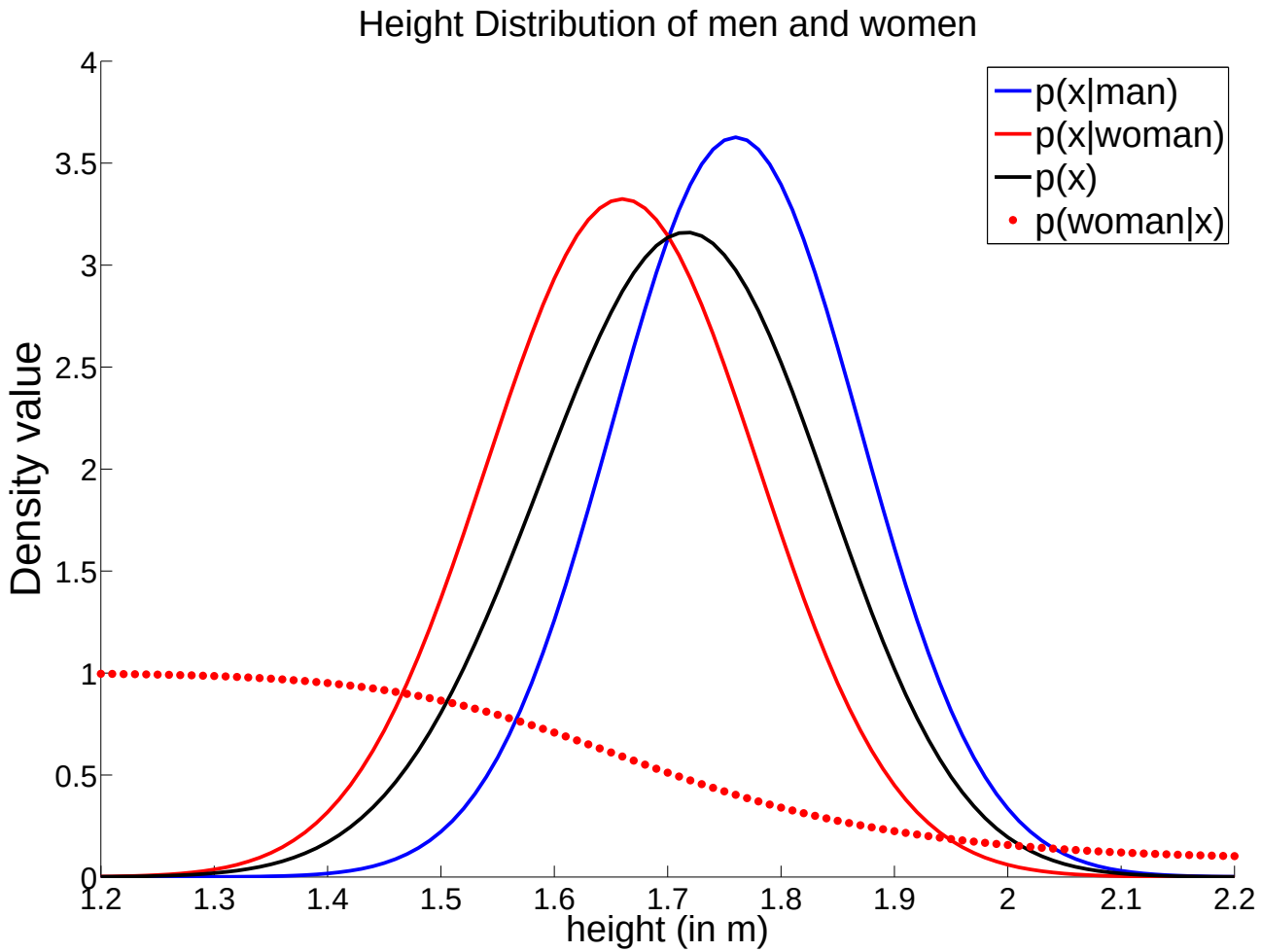
Statistical learning I

Assumption: Data is generated by sampling from a **probability measure** P on $\mathcal{X} \times \mathcal{Y}$.

What does that mean ?

1. Training data is a **random sample** from P,
2. The labels $y \in \mathcal{Y}$ are **non-deterministic**, that means there exists not necessarily a function $y = g(x)$. Instead for a given input x, there exists a distribution over the possible values in $\mathcal{Y}$.
3. Since the training data underlies statistical fluctuations, the classifier should be relatively stable under small changes of the training data.

Learning problem: Predict the sex of a person, $Y = \{\text{male}, \text{female}\}$, using the measured height of the person as a feature (input space is $\mathcal{X} = \mathbb{R}$).



Marginal distribution

$$p(x) = p(x|\text{male})p(\text{male}) + p(x|\text{female})p(\text{female}).$$

Using Bayes law we get the conditional distribution $p(y|x)$,

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}.$$

Classification rule: classify x as female if $p(\text{female}|x) \geq \frac{1}{2}$ and otherwise as male.

$\Longrightarrow$ From the plot, female if $x < 1.71$ and otherwise male.

Discriminative versus generative learning

Two main types of approaches to solve a (semi)-supervised statistical learning problem:

- **Generative Learning:** Estimation of the joint distribution $p(x, y)$ in particular the class conditional probabilities $p(x|y)$
  $\Longrightarrow$ Using Bayes rule compute the conditional probability $p(y|x)$.
  Advantage: One can create synthetically new data points.
  Disadvantage: One has to solve a harder problem
  $\Longrightarrow$ predictions often worse than in discriminative learning.
- **Discriminative Learning:** Just model the conditional distribution $p(y|x)$ (or even only the set $p(y|x) = \frac{1}{2}$ in binary classification).
  Support vector machines, neural networks and k-nearest neighbor classifiers are prominent representatives of discriminative learning.

Challenges in machine learning

Challenges in machine learning:

- choice of features
- integration of prior knowledge
- computational complexity
- curse of dimensionality
- over- and underfitting

Curse of dimensionality I

Often we have a lot of features $\Longrightarrow$ input/feature space is high-dimensional.

Our first classifier: Naive histogram estimator on $\mathcal{X} = [0,1]^d$.

- partition each dimension into k equal parts,
- there are k^d different bins
- classify each bin with the label occuring most often in that bin (majority vote).

In order to do classification we need at least one sample in each bin, that means $n = k^d$ samples.

Number of required samples increases exponentially with the number of dimensions !

Curse of dimensionality !

Curse of dimensionality II

Statistical setting: Assume uniform marginal distribution over $\mathcal{X} = [0,1]^d$.

$$\mathrm{P}(\text{bin } s \text{ empty}) = \left(1 - \frac{1}{k^d}\right)^n.$$

The probability that some bin contains no sample is given as

$$\mathrm{P}(\text{some bin empty}) = \mathrm{P}(\cup_{s=1}^{k^d} \text{bin } s \text{ empty}) \leq k^d \left(1 - \frac{1}{k^d}\right)^n \leq k^d \, e^{-\frac{n}{k^d}},$$

using $\mathrm{P}(\cup_{i=1}^m A_i) \leq \sum_{i=1}^m \mathrm{P}(A_i)$ and $(1 + \frac{x}{n})^n \leq e^x$ for $|x| \leq n$.
$A = \{\text{every bin is occupied}\}$ is the complement of the event
$B = \{\text{some bin empty}\}$

$$\mathrm{P}(\text{every bin is occupied}) \geq 1 - k^d \, e^{-\frac{n}{k^d}}.$$

In order that every bin is occupied with probability p, $n > k^d \log\left(\frac{k^d}{1-p}\right)$ samples are sufficient.

Curse of dimensionality III

In 10 dimensions we need for $k = 10$ already approximately $n = 10^{10}$ samples.
$\Rightarrow$ learning is impossible (for the naive histogram estimator).

How can we avoid the curse of dimensionality ?

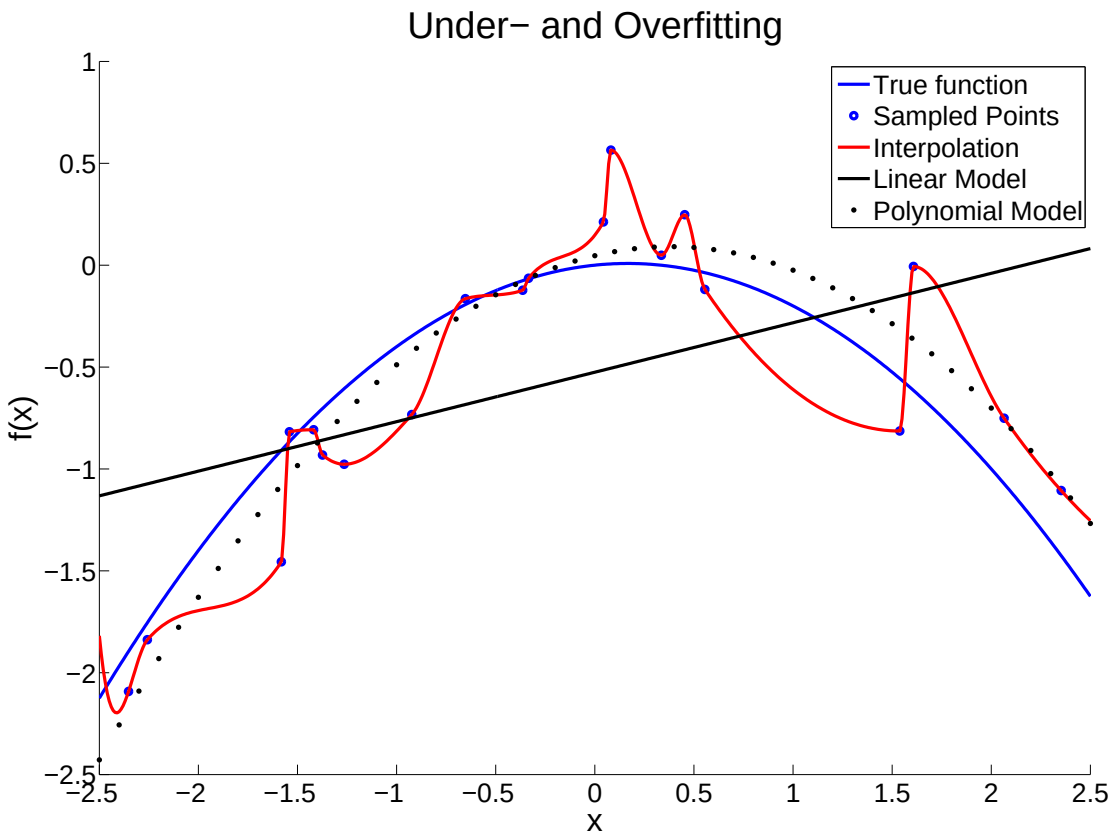
- input dimensions have dependencies,
- the output is "nice" e.g. varies smoothly with respect to the input.

$\Rightarrow$ in these cases learning is still possible.

Support vector machines avoid to some extent the curse of dimensionality since they rely on the **Maximum margin principle**.

Overfitting and underfitting I

Regression: input $\mathcal{X} = \mathbb{R}$, output $\mathcal{Y} = \mathbb{R}$, training data $(X_i, Y_i)_{i=1}^n$.



blue curve: true function, blue circles: 20 noisy samples of the true function, red curve: interpolation of the training points **black solid line:** fitted linear model, **dotted black line:** polynomial model.

Overfitting and underfitting II

- using **interpolation** techniques there **always** exists a function $f : \mathcal{X} \to \mathcal{Y}$ which fits the data perfectly ! (given that there are no contradictions, that is if $X_i = X_j$, $i \neq j$, then $Y_i = Y_j$.)
  $\Longrightarrow$ **overfitting** of the data.
  $\Longrightarrow$ **no generalization !**
- using a very simple regression model e.g. a linear one will often lead to **underfitting**, which means that the learned function can hardly represent the functional relationship given by the data.
  $\Longrightarrow$ **generalization but poor performance !**

Purpose of different learning methods

One learning method for all purposes ?

- Key result of statistical learning theory: there exists no universally best learning method. On "average" (this has to be defined carefully !) they perform all the same.
- Nature is "nice" to us - most problems are not of pathological nature.

Purpose of different learning methods ?

- Each learning method implicitly or explicitly models different prior assumptions about the input-output relationship. The art of machine learning is to choose the method which best fits the data generating process.
- **Data can never replace prior knowledge.**