Artificial Intelligence

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Thorsten Schmidt
Abteilung für Mathematische Stochastik

www.stochastik.uni-freiburg.de
thorsten.schmidt@stochastik.uni-freiburg.de
SS 2017

Our goal today

Motivation

Overview

A hierarchy

Machine learning examples

Introduction

Basics

Supervised learning
Unsupervised learning
Semi-supervised learning
Reinforcement learning

Machine Learning Basics

Literature (incomplete, but growing):

- Ian Goodfellow, Yoshua Bengio und Aaron Courville (2016). **Deep Learning**. http://www.deeplearningbook.org. MIT Press
- D. Barber (2012). **Bayesian Reasoning and Machine Learning**. Cambridge University Press
- Richard S. Sutton und Andrew G. Barto (1998). **Reinforcement Learning : An Introduction**. MIT Press
- Gareth James u. a. (2014). **An Introduction to Statistical Learning: With Applications in R**. Springer Publishing Company, Incorporated. ISBN: 1461471370, 9781461471370
- Trevor Hastie, Robert Tibshirani und Jerome Friedman (2009). **The Elements of Statistical Learning**. Springer Series in Statistics. Springer New York Inc.

Motivation

- Artificial Intelligence includes machine learning as one exciting special case
- Machine Learning is nowadays used at many places (Google, Amazon, etc.)
- It is a great job opportunity ! It needs maths and probability !
- Many applications are surprisingly successful (speech / face recognition) and currently people are seeking further applications
- Here we want to learn about the foundations, discuss implications and what can be done by ML and what not
- The lecture is an open forum for discussions and will be developed during the semester. Slides will be available online, one day ahead. The exercises will include computational projects, in particular towards the end.

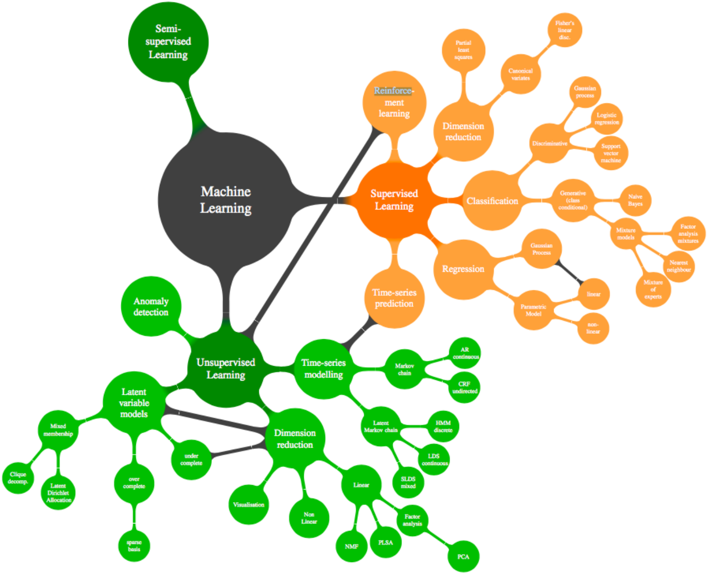
Overview[1]

- Artificial intelligence is the field where computers solve problems.
- It is easy for a computer to solve tasks which can be described formally (Chess, Tic-Tac-Toe). The challenge is to solve a tasks which are hard to describe formally (but are easy for humans: walk, drive a car, speak, recognize people ...)
- The solution is to allow computers to learn from experience and to understand the world by a hierarchy of concepts, each concept defined in terms of its relation to simpler concepts.
- A fixed knowledge-base would be somehow limiting such that we are interested in such attempts where the systems acquire their own knowledge, which we call **Machine Learning**.

[1] This introduction follows closely Goodfellow et.al. (2016).

- First examples of machine learning are **logistic regression** or **naive Bayes** $\rightarrow$ standard statistical procedures (Cesarean delivery / Recognition of Spam, more examples to follow)
- Problems become simpler with a nice representation. Of course it would be nice if the system itself could find such a representation, which we call **representation learning**.
- An example is the so-called **auto-encoder**. This is a combination of an encoder and a decoder. The encoder converts the input to a certain representation and the decoder converts it back again, such that the result has nice properties.
- Speech for example might be influenced by many factors of variation (age, sex, origin, ...) and it needs nearly human understanding to disentangle the variation from the content we are interested in.
- **Deep Learning** solves this problem by introducing hierarchical representations.

- This leads to the following hiearchy:
- AI $\rightarrow$ machine learning $\rightarrow$ representation learning $\rightarrow$ deep learning.



Source: Barber (2012).

Examples of Machine Learning

Some of the most prominent examples:

- LeCun et.al.[2] recognition of handwritten digits. The MNIST Database[3] provides 60.000 samples for testing algorithms.
- The Viola & Jones face recognition,[4]. This path-breaking work proposed a procedure to combine existing tools with machine-learning algorithms. One key is the use of approx. 5000 learning pictures to train the routine. We will revisit this procedure shortly.

---

[2] Yann LeCun u. a. (1998). „Gradient-based learning applied to document recognition". In: **Proceedings of the IEEE** 86.11, S. 2278–2324.

[3] `http://yann.lecun.com/exdb/mnist/`

[4] Paul Viola und Michael Jones (2001). „Robust Real-time Object Detection". In: **International Journal of Computer Vision**. Bd. 4. 34–47.

- Speech recognition has long been a difficult problem for computers (first works date to the 50's) and only recently been solved with high computer power. It may seem surprising, that mathematical tools are at the core of these solutions. Let us quote Hinton et.al.[5]

> *Most current speech recognition systems use hidden Markov models (HMMs) to deal with the temporal variability of speech and Gaussian mixture models (GMMs) to deter- mine how well each state of each HMM fits a frame or a short window of frames of coefficients that repre- sents the acoustic input. (...) Deep neural networks (DNNs) that have many hidden layers and are trained using new methods have been shown to outperform GMMs on a vari- ety of speech recognition benchmarks, sometimes by a large margin*

  So, one of our tasks will be to develop a little bit of mathematical tools which we will need later. Most notably, some of the mathematical parts can be replaced by deep learning, which will be of high interest to us.

[5] Geoffrey Hinton u. a. (2012). „Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups". In: **IEEE Signal Processing Magazine** 29.6, S. 82–97.

Types of machine learning:

- Supervised learning: The data consists of datapoints and associated labels, i.e. we start from the dataset

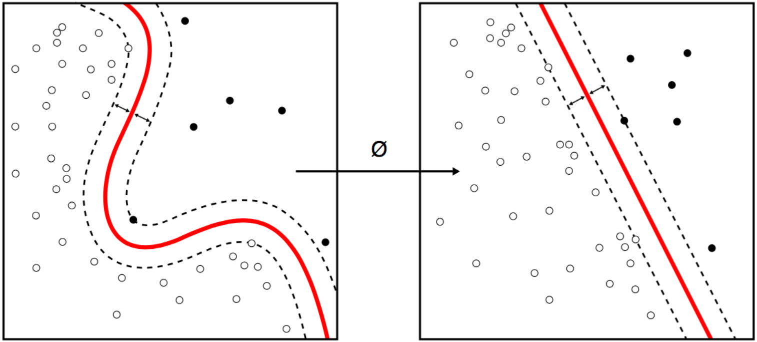
$$(x_i, y_i)_{i \in I}.$$

  We give some examples:
- **Image recognition** (face recognition) where the images come with labels, i.e. cats / dogs or the person to which the image is associated to.
- **Spam filter** the training set contains emails together with the label spam / no spam.
- **Speech recognition** here sample speech files comes together with the content of the sentences. It is clear, that some sort of grammar understanding helps to break up the sentences into smaller pieces, i.e. words.

- Unsupervised learning: In this case the data just comes at it is, i.e.

$$(x_i)_{i \in I}$$

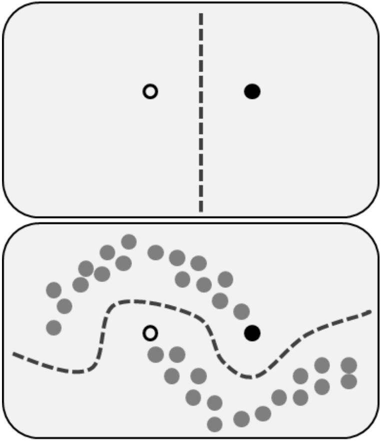
and one goal would be to identify a certain structure from the data itself. In this sense the machine learning algorithm shall itself find a characteristics which divides the data into suitable subsets.



Picture by: Alisneaky, svg version by User:Zirguezi - Own work, Some examples

- Analysis of genomic data
- Density estimation
- Clustering
- Principal component analysis

- Semi-supervised learning: only a few data are labelled and many are unlabelled.
- Labelling typically is quite expensive and the additional use of unlabelled data might improve the performance. However, some assumptions need to be made, such that this procedure works through.



Picture by: Techerin - Own work,

Reinforcement learning

is quite different from the above examples.

- First: time matters, the problem depends on time ! Observations accumulate over time.
- There is no supervisor but a reward signal measuring the quality of the decision.
- The approach utilizes a probabilistic framework: Markov decision processes.
- Examples are: drive a car, optimally manage a portfolio ...

- In a nutshell, we proceed iteratively through time.
- At time t, we *observe* X_t, get a reward $U(X_t)$ and are able to make a decision D_t which influences the state at time $t+1$, X_{t+1}.
- A *policy* describes the decision given the state. It can be stochastic or deterministic.
- While initially the environment is unknown, the system gathers information through its interactions with the environment and improves its policy.

A quite related area is Statistical Learning. This new area of statistics is quite related to machine learning and we will study a number of relevant problems.

Definition

A computer program learns from experience E with respect to tasks T, if its performance P improves with experience E.

This quite vague definition allows us to develop some intuition about the situation.

- **Experience** is given by an increasing sequence of observations, for example $X_1, X_2, \dots, X_t$ could represent the information at time t. This is typically decoded in a **filtration**: a filtration is an increasing sequence of sub-σ-fields $(\mathscr{F}_t)_{t \in \mathscr{T}}$.
- The performance is often measures in terms of an **utility function**. For example the utility at time t could be given by $U(X_t)$ with an function U. U could of course depend on more variables. One could also look for the accumulated utility

$$\sum_{t=1}^{T} U(X_t).$$

One very simple learning algorithm is linear regression, a classical statistical concept. Here it arises as an example of **supervised learning**.

Example (Linear Regression)

Suppose we oberseve pairs $(x_i, y_i)_{i=1,\dots,n}$ and want to predict y on basis of x. **Linear** regression requires

$$\hat{y}(x) = \beta x$$

with some weight $\beta \in \mathbb{R}$. We specify a loss function[6]

$$\text{RSS}(\beta) := \sum_{i=1}^{n} (y_i - \hat{y}(x_i))^2$$

and minimize over β.

One could choose $-$MSE as utility function. So how does the system **learn**?

[6]Given by the Residual Sum of Squares here.

The system learns by maximizing the utility, i.e. minimizing the MSE for each n. And additional data will lead to a better prediciton. We will later see that this is in a certain sense indeed optimal.

We use the first-oder condition to derive the solution letting $\boldsymbol{x} = (x_1, \dots, x_n)$ and similar for $\boldsymbol{y}$,

$$
\begin{aligned}
& 0 = \partial_\beta (\boldsymbol{y} - \beta \boldsymbol{x})^2 = \partial_\beta (\boldsymbol{y}^2 - 2\boldsymbol{y}^\top \beta \boldsymbol{x} + \beta^2 \boldsymbol{x}^\top \boldsymbol{x}) \\
\Leftrightarrow \beta \quad & 0 = -2\boldsymbol{x}^\top \boldsymbol{y} + 2\beta \boldsymbol{x}^\top \boldsymbol{x}
\end{aligned}
$$

such that we obtain

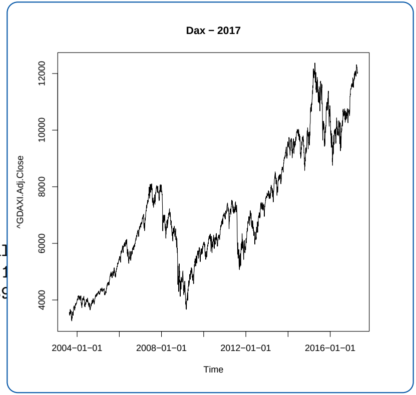
$$
\hat{\beta} = (\boldsymbol{x}^\top \boldsymbol{x})^{-1} \boldsymbol{x}^\top \boldsymbol{y}.
$$

Note that typically one considers affine functions of x without mentioning, i.e. one looks at functions $y = \alpha + \beta x$. This can simply be achieved with the linear approach by augmenting $\boldsymbol{x}$ by an additional entry 1.

- Of course many generalizations are possible:
- To higher dimensions: consider data vectors $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, $i = 1, \dots, n$,
- To nonlinear functions: include $x_i^1, \dots, x_i^p$ into the covariates
- and many more.

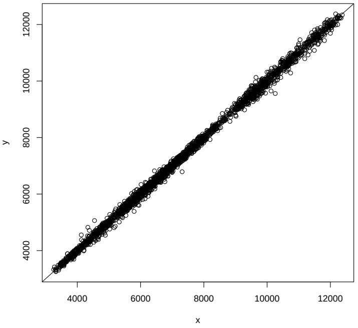
Let us consider a linear regression in R.

```
library (fImport)
stockdata <- yahooSeries(c("^GDAXI"),nDaysBack=5000)[,c("^GDAXI.Adj.Close")]
plot(stockdata)

N=length(stockdata)
# prepare for linear regression
x = stockdata[1:N-1]
y = stockdata[2:N]

plot(x,y)

Regression = lm (y~x)
summary (Regression)
abline (Regression)

# Coefficients:
#              Estimate Std. Error  t val
# (Intercept) 6.0077820  5.1533966    1
# x           0.9994964  0.0006941 1439
```



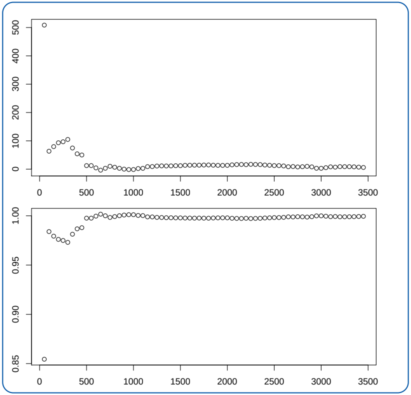
Let us consider a linear regression in R.

```
library (fImport)
stockdata <- yahooSeries(c("^GDAXI"),nDaysBack=5000)[,c("^GDAXI.Adj.Close")]
plot(stockdata)

N=length(stockdata)
# prepare for linear regression
x = stockdata[1:N-1]
y = stockdata[2:N]

plot(x,y)

Regression = lm (y~x)
summary (Regression)
abline (Regression)

# Coefficients:
#             Estimate Std. Error  t val
# (Intercept) 6.0077820  5.1533966    1
# x           0.9994964  0.0006941 1439
```



Now we consider the learning effect:

```
n=round(N/50)+1
ab = array(rep(0,2*n),dim = c(n,2))
j = 50; i=1
while (j < N-1) {
  Regression = lm(y[1:j]~x[1:j])
  ab [i,] = Regression$coefficients
  i=i+1; j=j+50  }
i=i-1

par ( mfrow = c(2,1),mar=c(2,2.1,1,1))
plot((1:i)*50,ab[1:i,1])
plot((1:i)*50,ab[1:i,2])
```



Could we improve this ? Suggestions ?

What is the difference to Statistics ?

In a statistical approach we start with a **parametric model**:

$$Y_i = \alpha + \beta x_i + \varepsilon_i, \quad i = 1, \ldots, n$$

and assume that $\varepsilon_1, \ldots, \varepsilon_n$ have a certain structure (for example, i.i.d. and $\mathcal{N}(0, \sigma^2)$). The one can derive (see, e.g. Czado & Schmidt (2011)) **optimal estimators** for α and β. One can also relax the assumptions and gets weaker results.

So what ? What are the advantages of the statistical approach ?

One particular outcome is that we are able to provide **confidence intervals**, **predictive intervals** and **test** hypothesises.