Machine Learning

0. Introduction

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Computer Science
University of Hildesheim, Germany

Syllabus

Fri. 26.10.	(1)	0. Introduction
		A. Supervised Learning: Linear Models & Fundamentals
Fri. 2.11.	(2)	A.1 Linear Regression
Fri. 9.11.	(3)	A.2 Linear Classification
Fri. 16.11.	(4)	A.3 Regularization
Fri. 23.11.	(5)	A.4 High-dimensional Data
		B. Supervised Learning: Nonlinear Models
Fri. 30.11.	(6)	B.1 Nearest-Neighbor Models
Fri. 7.12.	(7)	B.4 Support Vector Machines
Fri. 14.12.	(8)	B.3 Decision Trees
Fri. 21.12.	(9)	B.5 A First Look at Bayesian and Markov Networks
		— *Christmas Break* —
Fri. 11.1.	(10)	B.2 Neural Networks
		C. Unsupervised Learning
Fri. 18.1.	(11)	C.1 Clustering
Fri. 25.1.	(12)	C.2 Dimensionality Reduction
Fri. 1.2.	(13)	C.3 Frequent Pattern Mining
Fri. 8.2.	(14)	Q&A

Outline

Outline

1. What is Machine Learning?

2. A First View at Linear Regression

3. Machine Learning Problems

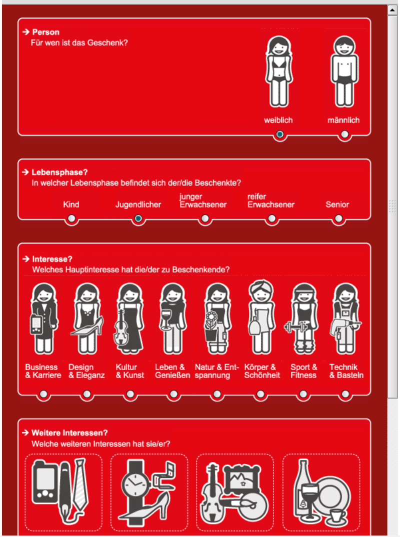
4. Lecture Overview

5. Organizational Stuff

What is Machine Learning?

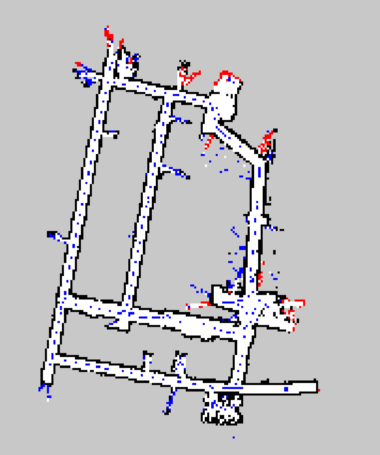
What is Machine Learning?

1. E-Commerce: predict what customers will buy.



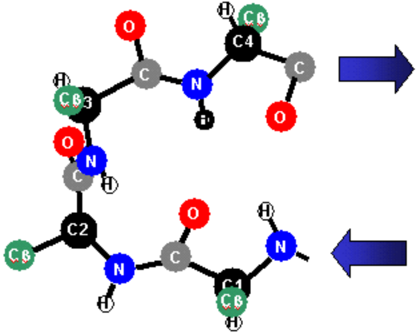
What is Machine Learning?

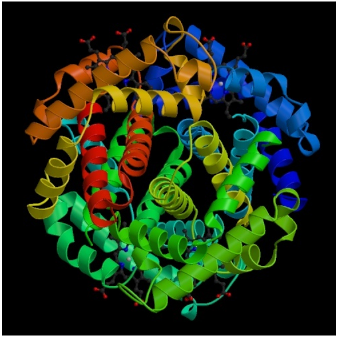
2. Robotics: Build a map of the environment based on sensor signals.



What is Machine Learning?

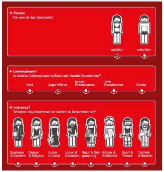
3. Bioinformatics: predict the 3d structure of a molecule based on its sequence.





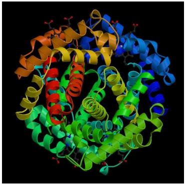
What is Machine Learning?

Information Systems



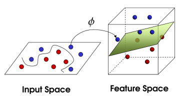
Robotics

Bioinformatics



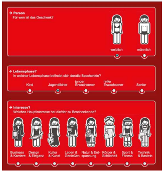
Many Further Applications!

MACHINE LEARNING



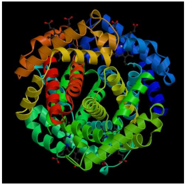
What is Machine Learning?

Information Systems



Robotics

Bioinformatics



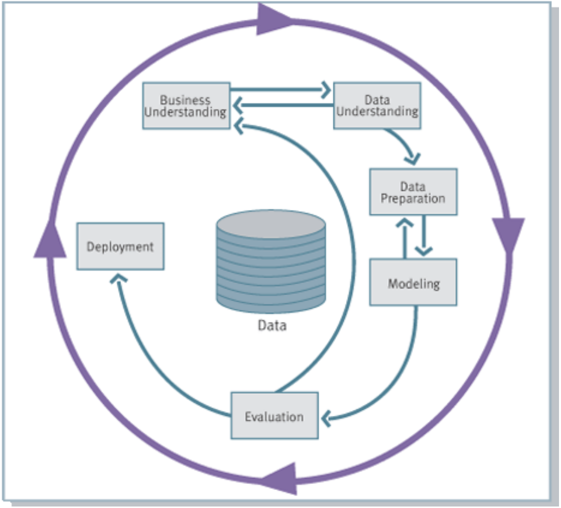
Many Further Applications!

MACHINE LEARNING

OPTIMIZATION

NUMERICS

Process models



Cross Industry Standard Process for Data Mining (CRISP-DM)

One area of research, many names (and aspects)

machine learning
historically, stresses learning logical or rule-based models (vs. probabilistic models).

data mining, **big data**
stresses the aspect of large datasets and complicated tasks.

knowledge discovery in databases (KDD)
stresses the embedding of machine learning tasks in applications, i.e., preprocessing & deployment.

data analysis historically, stresses multivariate regression and unsupervised tasks.

pattern recognition
name preferred by engineers, stresses cognitive applications such as image and speech analysis.

data science, **applied statistics**
stresses underlying statistical models, testing and methodical rigor.

predictive analytics, **business analytics**, **data analytics**
stresses business applications.

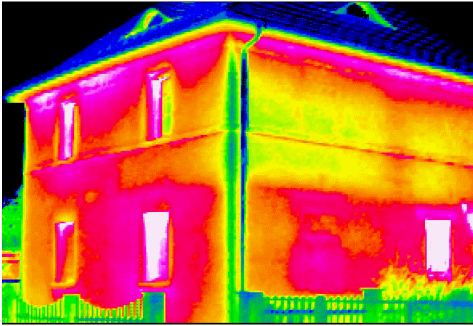
Outline

Example

How does gas consumption depend on external temperature?

Example data (Whiteside, 1960s): weekly measurements of

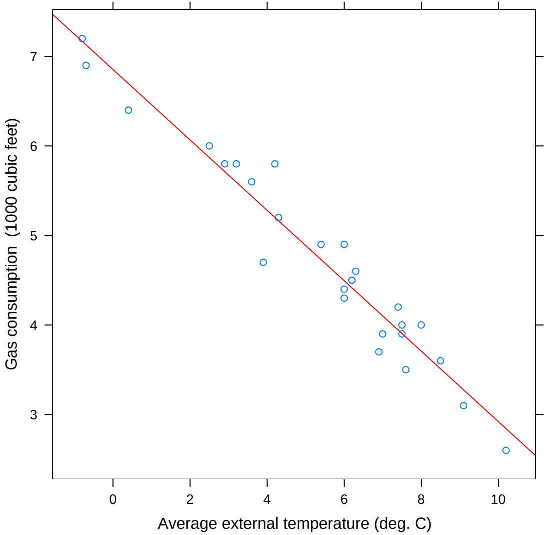
- average external temperature
- total gas consumption (in 1000 cubic feets)



How does gas consumption depend on external temperature?

How much gas is needed for a given temperature ?

Example


Gas consumption (1000 cubic feet)
7
6
5
4
3
0
2
4
6
8
10
Average external temperature (deg. C)

The Simple Linear Regression Problem (yet vague)

Given

- a set $\mathcal{D}^{\text{train}} := \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \subseteq \mathbb{R} \times \mathbb{R}$ called **training data**,

compute the line that describes the data generating process best.

The Simple Linear Model

For given predictor/input $x \in \mathbb{R}$, the **simple linear model** predicts/outputs

$$\hat{y}(x) := \hat{\beta}_0 + \hat{\beta}_1 x$$

with **parameters** $(\hat{\beta}_0, \hat{\beta}_1)$ called

$\hat{\beta}_0$ **intercept** / **bias** / **offset**

$\hat{\beta}_1$ **slope**

```
predict-simple-linreg(x ∈ ℝ, β̂₀, β̂₁ ∈ ℝ):
  ŷ := β̂₀ + β̂₁x
  return ŷ
```

When is a Model Good?

We still need to specify what "describes the data generating process best" means. — What are good predictions $\hat{y}(x)$?

Predictions are considered better the smaller the difference between

- an **observed** y_n (for predictors x_n) and
- a **predicted** $\hat{y}_n := \hat{y}(x_n)$

is on average, e.g., the smaller the **L2 loss** / **squared error**:

$$\ell(y_n, \hat{y}_n) := (y_n - \hat{y}_n)^2$$

Note: Other error measures such as absolute error $\ell(y_n, \hat{y}_n) = |y_n - \hat{y}_n|$ are also possible, but more difficult to handle.

When is a Model Good?

Pointwise losses are usually averaged over a dataset $\mathcal{D}$

$$\mathrm{err}(\hat{y};\mathcal{D}) := \frac{1}{N}\mathrm{RSS}(\hat{y};\mathcal{D}) = \frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}(x_n))^2$$

$$\text{or}\ \ \mathrm{err}(\hat{y};\mathcal{D}) := \mathrm{RSS}(\hat{y};\mathcal{D}) := \sum_{n=1}^{N}(y_n - \hat{y}(x_n))^2$$

called **residual sum of squares** (RSS) or generally **error/risk**.

Equivalently, often **Root Mean Square Error** (RMSE) is used:

$$\mathrm{err}(\hat{y};\mathcal{D}) := \mathrm{RMSE}(\hat{y};\mathcal{D}) := \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{y}(x_n))^2}$$

Note: RMSE has the same scale level / unit as the original target y, e.g., if y is measured in meters so is RMSE.

Generalization

We can trivially get a model with error zero on training data, e.g., by simply looking up the corresponding y_n for each x_n:

$$\hat{y}^{\text{lookup}}(x) := \begin{cases} y_n, & \text{if } x = x_n \\ 0, & \text{else} \end{cases}$$

with $\text{RSS}(\hat{y}^{\text{lookup}}, \mathcal{D}^{\text{train}}) = 0$ optimal

Models should not just reproduce the data, but **generalize**, i.e., predict well on fresh / unseen data (called **test data**).

The Simple Linear Regression Problem

Given

- a set $\mathcal{D}^{\text{train}} := \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \subseteq \mathbb{R} \times \mathbb{R}$ called **training data**,

compute the **parameters** $(\hat{\beta}_0, \hat{\beta}_1)$ of a linear **regression function**

$$\hat{y}(x) := \hat{\beta}_0 + \hat{\beta}_1 x$$

s.t. for a set $\mathcal{D}^{\text{test}} \subseteq \mathbb{R} \times \mathbb{R}$ called **test set** the **test error**

$$\text{err}(\hat{y}; \mathcal{D}^{\text{test}}) := \frac{1}{|D^{\text{test}}|} \sum_{(x,y) \in \mathcal{D}^{\text{test}}} (y - \hat{y}(x))^2$$

is minimal.

Note: $\mathcal{D}^{\text{test}}$ has (i) to be from the same data generating process and (ii) not to be available during training.

Least Squares Estimates

As $\mathcal{D}^{\text{test}}$ is not accessible during training, use $\mathcal{D}^{\text{train}}$ as **proxy** for $\mathcal{D}^{\text{test}}$:

- rationale: models predicting well on $\mathcal{D}^{\text{train}}$ should also predict well on $\mathcal{D}^{\text{test}}$ as both come from the same data generating process.

The parameters with minimal L2 loss for a dataset $\mathcal{D}^{\text{train}} := \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ are called **(ordinary) least squares estimates**:

$$
\begin{aligned}
(\hat{\beta}_0, \hat{\beta}_1) &:= \arg\min_{\hat{\beta}_0, \hat{\beta}_1} \text{RSS}(\hat{y}, \mathcal{D}^{\text{train}}) \\
&:= \arg\min_{\hat{\beta}_0, \hat{\beta}_1} \sum_{n=1}^{N} (y_n - \hat{y}(x_n))^2 \\
&= \arg\min_{\hat{\beta}_0, \hat{\beta}_1} \sum_{n=1}^{N} (y_n - (\hat{\beta}_0 + \hat{\beta}_1 x_n))^2
\end{aligned}
$$

Learning the Least Squares Estimates

The least squares estimates can be written in closed form:

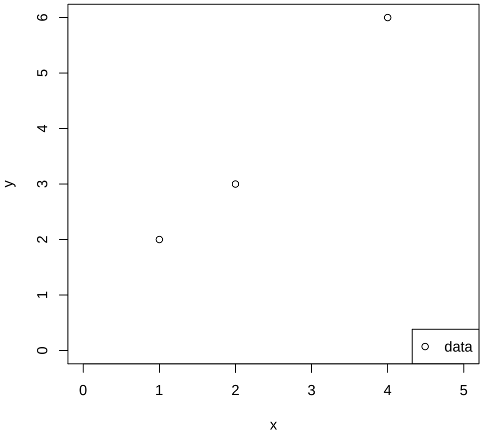
$$\hat{\beta}_1 = \frac{\sum_{n=1}^{N}(x_n - \bar{x})(y_n - \bar{y})}{\sum_{n=1}^{N}(x_n - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

learn−simple−linreg($\mathcal{D}^{\text{train}} := \{(x_1, y_1), \ldots, (x_N, y_N)\} \in \mathbb{R} \times \mathbb{R}$):
  $\bar{x} := \frac{1}{N}\sum_{n=1}^{N} x_n$
  $\bar{y} := \frac{1}{N}\sum_{n=1}^{N} y_n$
  $\hat{\beta}_1 := \frac{\sum_{n=1}^{N}(x_n - \bar{x})(y_n - \bar{y})}{\sum_{n=1}^{N}(x_n - \bar{x})^2}$
  $\hat{\beta}_0 := \bar{y} - \hat{\beta}_1 \bar{x}$
  return $(\hat{\beta}_0, \hat{\beta}_1)$

A Toy Example

Given the data $\mathcal{D} := \{(1,2), (2,3), (4,6)\}$, predict a value for $x = 3$.



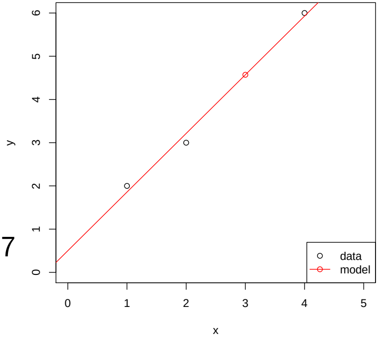
A Toy Example / Least Squares Estimates

Given the data $\mathcal{D} := \{(1,2),(2,3),(4,6)\}$, predict a value for $x = 3$.
Use a simple linear model.
$\bar{x} = 7/3$, $\bar{y} = 11/3$.

n	$x_n - \bar{x}$	$y_n - \bar{y}$	$(x_n - \bar{x})^2$	$(x_n - \bar{x}) \cdot (y_n - \bar{y})$
1	$-4/3$	$-5/3$	$16/9$	$20/9$
2	$-1/3$	$-2/3$	$1/9$	$2/9$
3	$5/3$	$7/3$	$25/9$	$35/9$
$\sum$			$42/9$	$57/9$

$$\hat{\beta}_1 = \frac{\sum_{n=1}^{N}(x_n - \bar{x})(y_n - \bar{y})}{\sum_{n=1}^{N}(x_n - \bar{x})^2} = 57/42 = 1.357$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} = \frac{11}{3} - \frac{57}{42} \cdot \frac{7}{3} = \frac{63}{126} = 0.5$$

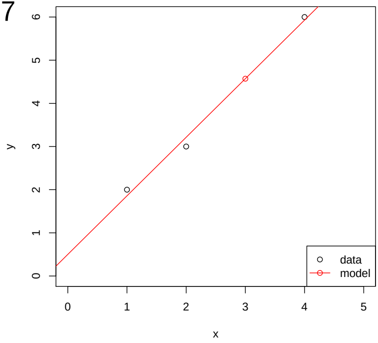


A Toy Example / Least Squares Estimates

Given the data $\mathcal{D} := \{(1,2),(2,3),(4,6)\}$, predict a value for $x = 3$.
Use a simple linear model.

$$\hat{\beta}_1 = \frac{\sum_{n=1}^N (x_n - \bar{x})(y_n - \bar{y})}{\sum_{n=1}^N (x_n - \bar{x})^2} = 57/42 = 1.357$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} = \frac{11}{3} - \frac{57}{42} \cdot \frac{7}{3} = \frac{63}{126} = 0.5$$



RSS:

n	y_n	$\hat{y}_n$	$(y_n - \hat{y}_n)^2$
1	2	1.857	0.020
2	3	3.214	0.046
3	6	5.929	0.005
$\sum$			0.071

$\hat{y}(3) = 4.571$

Outline

Regression

Real regression problems are more complex than simple linear regression in many aspects:

- There is more than one predictor.
- The target may depend non-linearly on the predictors.

Examples:

- predict sales figures.
- predict rating for a customer review.
- ...

Classification

Example: classifying iris plants (Anderson 1935).

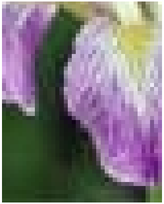
150 iris plants (50 of each species):

- species: setosa, versicolor, virginica
- length and width of sepals (in cm)
- length and width of petals (in cm)

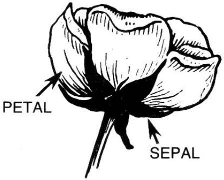
Given the lengths and widths of sepals and petals of an instance, which iris species does it belong to?

iris setosa

iris versicolor



iris virginica

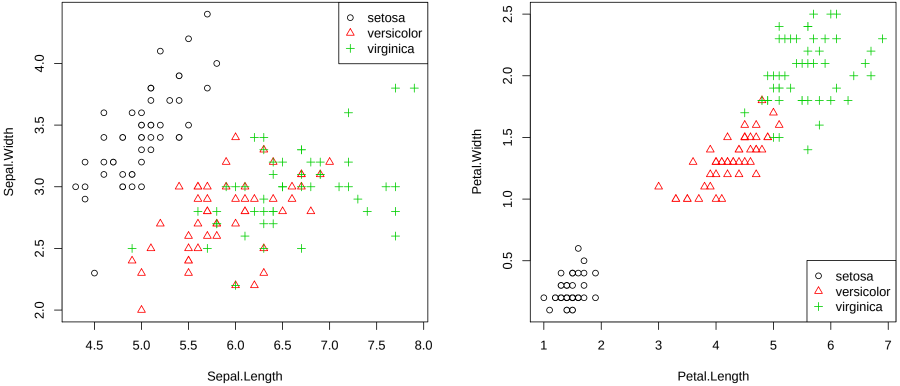


[source: iris species database, http://www.badbear.com/signa]

Classification

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.10	3.50	1.40	0.20	setosa
2	4.90	3.00	1.40	0.20	setosa
3	4.70	3.20	1.30	0.20	setosa
4	4.60	3.10	1.50	0.20	setosa
⋮	⋮	⋮	⋮	⋮	⋮
51	7.00	3.20	4.70	1.40	versicolor
52	6.40	3.20	4.50	1.50	versicolor
53	6.90	3.10	4.90	1.50	versicolor
⋮	⋮	⋮	⋮	⋮	⋮
101	6.30	3.30	6.00	2.50	virginica
102	5.80	2.70	5.10	1.90	virginica
103	7.10	3.00	5.90	2.10	virginica
⋮	⋮	⋮	⋮	⋮	⋮
150	5.90	3.00	5.10	1.80	virginica

Classification



Classification

Example: classifying email (lingspam corpus)

Subject: query: melcuk (melchuk)

does anybody know a working email (or other) address for igor melcuk (melchuk) ?

legitimate email ("ham")

Subject: '

hello ! come see our naughty little city made especially for adults http://208.26.207.98/freeweek/enter.html once you get here, you won't want to leave !

spam

How to classify email messages as spam or ham?

Classification

Subject: query: melcuk (melchuk)

does anybody know a working email (or other) address for igor melcuk (melchuk) ?

$\Rightarrow$

$$\begin{pmatrix} \text{a} & 1 \\ \text{address} & 1 \\ \text{anybody} & 1 \\ \text{does} & 1 \\ \text{email} & 1 \\ \text{for} & 1 \\ \text{igor} & 1 \\ \text{know} & 1 \\ \text{melcuk} & 2 \\ \text{melchuk} & 2 \\ \text{or} & 1 \\ \text{other} & 1 \\ \text{query} & 1 \\ \text{working} & 1 \end{pmatrix}$$

Classification

lingspam corpus:

- email messages from a linguistics mailing list.
- 2414 ham messages.
- 481 spam messages.
- 54742 different words.
- an example for an early, but very small spam corpus.

Classification

All words that occur at least in each second spam or ham message on average (counting multiplicities):

	!	your	will	we	all	mail	from	do	our	email
spam	14.18	7.45	4.36	3.42	2.88	2.77	2.69	2.66	2.46	2.24
ham	0.38	0.46	1.93	0.94	0.83	0.79	1.60	0.57	0.30	0.39

	out	report	order	as	free	language	university
spam	2.19	2.14	2.09	2.07	2.04	0.04	0.05
ham	0.34	0.05	0.27	2.38	0.97	2.67	2.61

example rule:

if freq("!") $\geq$ 7 and freq("language")$=$0 and freq("university")$=$0 then spam, else ham

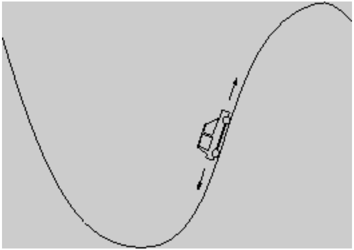
Should we better normalize for message length?

Reinforcement Learning

A class of learning problems where

- the correct / optimal action never is shown,
- but only positive or negative feedback for an action actually taken is given.

Example: steering the mountain car.



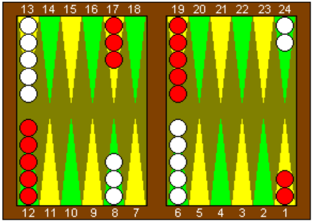
Observed are

- x-position of the car,
- velocity of the car

Possible actions are

- accelerate left,
- accelerate right,

Reinforcement Learning / TD-Gammon



Program	Hidden Units	Training Games	Opponents	Results
TD-Gam 0.0	40	300,000	Other Programs	Tied for Best
TD-Gam 1.0	80	300,000	Robertie, Magriel, ...	−13 pts / 51 games
TD-Gam 2.0	40	800,000	Var. Grandmasters	−7 pts / 38 games
TD-Gam 2.1	80	1,500,000	Robertie	−1 pts / 40 games
TD-Gam 3.0	80	1,500,000	Kazaros	+6 pts / 20 games

[source: ?]

See also Google's AlphaGo Zero [?] for Go!

Cluster Analysis

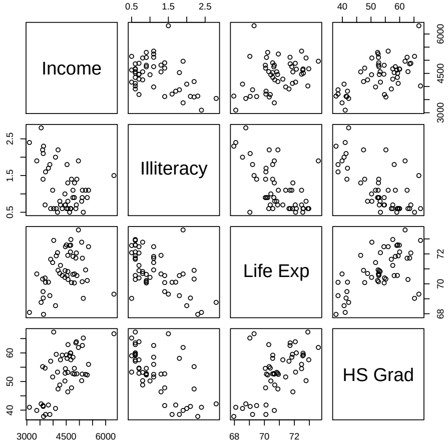
Finding groups of similar objects.

Example: sociographic data of the 50 US states in 1977.

state dataset:

- income (per capita, 1974),
- illiteracy (percent of population, 1970),
- life expectancy (in years, 1969–71),
- percent high-school graduates (1970).

(and some others not used here).



Fundamental Machine Learning Problems

1. Density Estimation
2. Regression
3. Classification

Supervised Learning

4. Optimal Control

Reinforcement Learning

5. Clustering
6. Dimensionality Reduction
7. Association Analysis

Unsupervised Learning

Supervised learning: correct decision is observed (**ground truth**).
Unsupervised learning: correct decision never is observed.

Outline

Syllabus

Fri. 26.10.	(1)	0. Introduction
		A. Supervised Learning: Linear Models & Fundamentals
Fri. 2.11.	(2)	A.1 Linear Regression
Fri. 9.11.	(3)	A.2 Linear Classification
Fri. 16.11.	(4)	A.3 Regularization
Fri. 23.11.	(5)	A.4 High-dimensional Data
		B. Supervised Learning: Nonlinear Models
Fri. 30.11.	(6)	B.1 Nearest-Neighbor Models
Fri. 7.12.	(7)	B.4 Support Vector Machines
Fri. 14.12.	(8)	B.3 Decision Trees
Fri. 21.12.	(9)	B.5 A First Look at Bayesian and Markov Networks
		— *Christmas Break* —
Fri. 11.1.	(10)	B.2 Neural Networks
		C. Unsupervised Learning
Fri. 18.1.	(11)	C.1 Clustering
Fri. 25.1.	(12)	C.2 Dimensionality Reduction
Fri. 1.2.	(13)	C.3 Frequent Pattern Mining
Fri. 8.2.	(14)	Q&A

Outline

Tutorial groups

No	Day & Time	Room	Box	Tutor	E-Mail
1	Mo. 8:00-10:00	B026	61	Randolf	scholz@uni-hildesheim.de
2	Mo. 8:00-10:00	C213	62	Riccardo	lucato@uni-hildesheim.de
3	We. 14:00-16:00	C213	63	Qi Yang	qiyang@uni-hildesheim.de
4	We. 8:00-10:00	C213	64	Migena	manami@uni-hildesheim.de
5	We. 8:00-10:00	B026	65	Nikhil	cherian@uni-hildesheim.de

- General information can be found at
  - LSF: https://lsf.uni-hildesheim.de
  - LearnWeb: https://www.uni-hildesheim.de/learnweb2018
- Lecture ID: 3101, Tutorial ID: 3102
- Exam: 25.02.2019 10:00-12:00 (register until 11.01.2019)

Tutorial organization

- Sheets will be uploaded on Tuesday 11:59 am at LearnWeb
- Deadline: Wednesday 11:59 pm if not stated otherwise
- Drop your printed or clearly handwritten results in the boxes (61-65) at Samelsonplatz
- Each week there are 2 exercises each 10 points
- You can earn bonus points for the exam by doing the exercises
- Team submission up to 3 people are allowed and encouraged
  - Each team member should be able to present the results on the board.
  - no bonus points can be granted for group submissions
- Plagiarism is illegal and usually leads to expulsion from the program.
  - about plagiarism see `https://en.wikipedia.org/wiki/Plagiarism`

Exam and Credit Points

- There will be a written exam at end of term (2h, 4 problems).
- The course gives 6 ECTS (2+2 SWS).
- The course can be used in
  - Angewandte Informatik BSc. / Informatik 5 (mandatory)
  - Angewandte Informatik MSc. / Informatik / Masch. Lernen (elective)
    - for students who did not have it in their Bachelors already
  - Data Analytics MSc. / Machine Learning (mandatory)
  - IMIT BSc. / Informatik 5 (mandatory)
  - IMIT MSc. / Informatik / Maschinelles Lernen (elective)
    - for students who did not have it in their Bachelors already
  - Wirtschaftsinformatik BSc. / Vertiefung Maschinelles Lernen (elective)
  - Wirtschaftsinformatik MSc. / Business Intelligence / Maschinelles Lernen (elective)
- This course is a pre-requisite for most courses at ISMLL.

Some Books

- Gareth James, Daniela Witten, Trevor Hastie, R. Tibshirani (2013): *An Introduction to Statistical Learning with Applications in R*, Springer.

- Kevin P. Murphy (2012): *Machine Learning, A Probabilistic Approach*, MIT Press.

- Trevor Hastie, Robert Tibshirani, Jerome Friedman (^{2}2009): *The Elements of Statistical Learning*, Springer.

  Also available online as PDF at `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`

- Christopher M. Bishop (2007): *Pattern Recognition and Machine Learning*, Springer.

- Richard O. Duda, Peter E. Hart, David G. Stork (^{2}2001): *Pattern Classification*, Springer.

Some First Machine Learning Software

- numpy (http://www.numpy.org) — vectors, matrices, arrays,
  pandas (https://pandas.pydata.org) — tables
  matplotlib (https://matplotlib.org) — plots
  - elementary building blocks for ML in Python
- scikit-learn (http://scikit-learn.org/)
  - Python based ML algorithms
- R (http://www.r-project.org)
  - statistical programming language in its own
- Weka (http://www.cs.waikato.ac.nz/~ml/)
  - Java based ML algorithms and GUI

Public data sets:

- UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/)
- UCI Knowledge Discovery in Databases Archive (http://kdd.ics.uci.edu/)

Further Readings

- For a general introduction: [**?**, chapter 1&2], [**?**, chapter 1], [**?**, chapter 1&2].
- For linear regression: [**?**, chapter 3], [**?**, chapter 7], [**?**, chapter 3].

References

Simple Linear Regression / Least Squares Estimates / Proof (p. 18):

$$\mathrm{RSS} = \sum_{i=1}^{n} (y_i - (\hat\beta_0 + \hat\beta_1 x_i))^2$$

$$\frac{\partial\, \mathrm{RSS}}{\partial \hat\beta_0} = \sum_{i=1}^{n} 2(y_i - (\hat\beta_0 + \hat\beta_1 x_i))(-1) \overset{!}{=} 0$$

$$\implies \quad n\hat\beta_0 = \sum_{i=1}^{n} (y_i - \hat\beta_1 x_i)$$

Simple Linear Regression / Least Squares Estimates / Proof

Proof (ctd.):

$$
\begin{aligned}
\text{RSS} &= \sum_{i=1}^{n} (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2 \\
&= \sum_{i=1}^{n} (y_i - (\bar{y} - \hat{\beta}_1 \bar{x}) - \hat{\beta}_1 x_i)^2 \\
&= \sum_{i=1}^{n} (y_i - \bar{y} - \hat{\beta}_1 (x_i - \bar{x}))^2 \\
\frac{\partial \,\text{RSS}}{\partial \hat{\beta}_1} &= \sum_{i=1}^{n} 2(y_i - \bar{y} - \hat{\beta}_1 (x_i - \bar{x}))(-1)(x_i - \bar{x}) \stackrel{!}{=} 0 \\
\implies \quad \hat{\beta}_1 &= \frac{\sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}
\end{aligned}
$$