Machine Learning

– winter term 2016/17 –

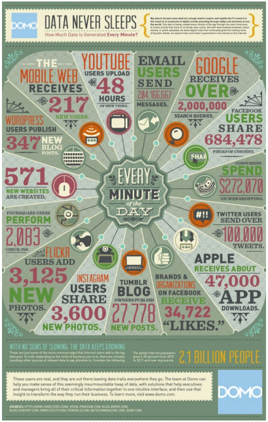
Chapter 01: Introduction

Prof. Adrian Ulges
Masters "Computer Science"
DCSM Department
University of Applied Sciences RheinMain

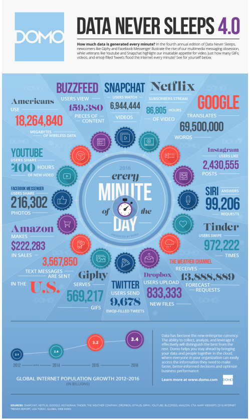
Outline

Machine Learning (“ML”) and Big Data images from [16]

(June 2012)



(June 2016)

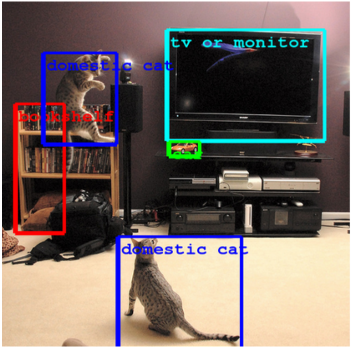


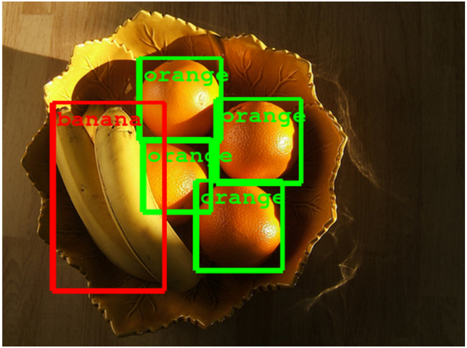
- There is more and more **unstructured** data around *(web 2.0, mobile devices, IoT, self-driving cars, ...)*

Machine Learning: Some recent Achievements image from [17]

"Imagine a historian of science writing about computer vision in the year 2100. They will identify the years 2011 to 2015 (and probably a few years beyond) as a time of **huge breakthroughs***, driven by deep convolutional nets."*

(M. Nielsen, "Neural Networks and Deep Learning")

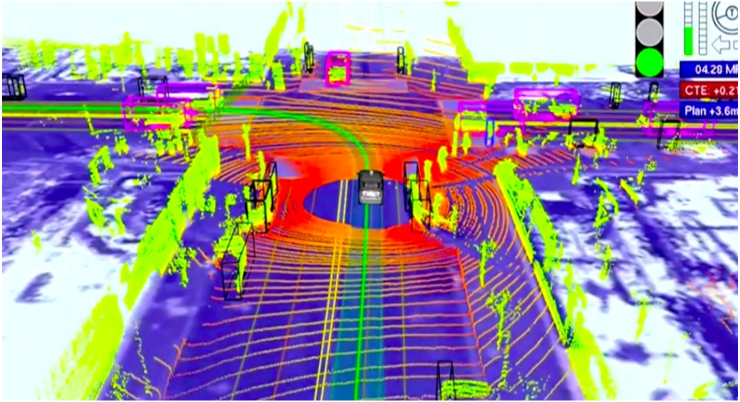




Machine Learning: Some recent Achievements image from [12]

"In the past decade, machine learning has given us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome."

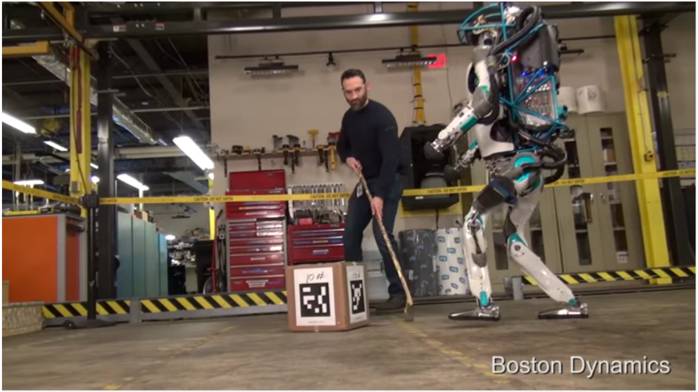
(Andrew Ng, Stanford University)



Machine Learning: Some recent Achievements image from [3]

"A survey carried out among AI experts recently shows they think machines will be as intelligent as humans by the year 2040."

(Nick Bostrom)



Atlas, The Next Generation

Machine Learning: Motivation

"If data had mass, the earth would be a black hole."

(Steven Marsland)

"As more data becomes available, more ambitious problems can be tackled. As a result, machine learning is **widely used in computer science** *and other fields. However, developing successful machine learning applications requires a substantial amount of* **'black art'** *that is hard to find in textbooks."*

(P. Domingos, *A few Useful Things to Know about Machine Learning*)

Example Applications

Have a look on kaggle.com

- Flight Quest
  optimize flight routes based on wheather and traffic
- TFI Restaurant Revenue Prediction
  predict annual sales of restaurants to open
- Job Recommendation Challenge
  predict which jobs users will apply to
- Whale Detection Challenge
  detect whale calls from audio, prevent collision with ship traffic
- Discovering trolling in user comments
- ...

There's also the 'classics'

- OCR, handwriting recognition, object recognition
- search engines, recommender systems, targeted advertising
- natural language processing, spam filtering

Outline

An ML Sample Application images from [4] [13]

- A computer system is to make a non-trivial decision
- Example: **spam filtering**
- Why not **hard-code** the decision logic?

Problems

- high effort to grasp problem's **complexity**
- easy to code *something*, difficult to reach the **optimal** program
- **feasibility checking**: What accuracy can be reached by a decision?
- code is extremely difficult to **maintain**
- keeping track of **data changes** (e.g., when spammers change strategies) is almost impossible
- there is no way to take **user feedback** into account

Machine Learning: Definition

"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** *from* **example inputs** *and using that to make* **predictions or decisions***, rather than following strictly static program instructions."*

(en.wikipedia.org)

"The field of study that gives computers the ability to learn **without being explicitly programmed.***"*

(Arthur Samuel (1959))

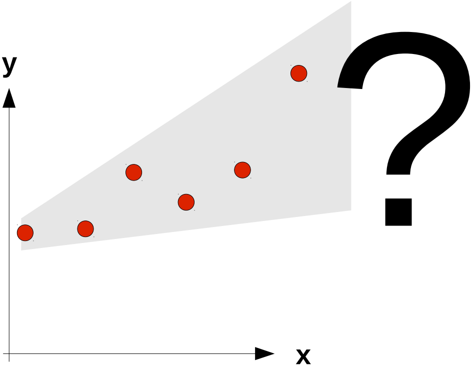
"A computer program is said to **learn** *from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with Experience E."*

(Tom Mitchell (1998))

Remark

These definitions are entirely **non-operational**!

Machine Learning's Hello World



- **Goal**: predict a person's weight in the future
- **Given**: a set of 2D samples $(x_1, y_1), \ldots, (x_n, y_n)$
  (x is time, y is person's weight)
- **Approach (*linear regression*)**: fit a line to the points and use this line for prediction
- Does this qualify as **machine learning**?

Machine Learning's Hello World

Linear Regression

- We define a line as a function

$$\mathcal{M}_\theta(x) = a \cdot x + b$$

- We measure the quality of a particular line $\theta = (a, b)$ using an error function E:

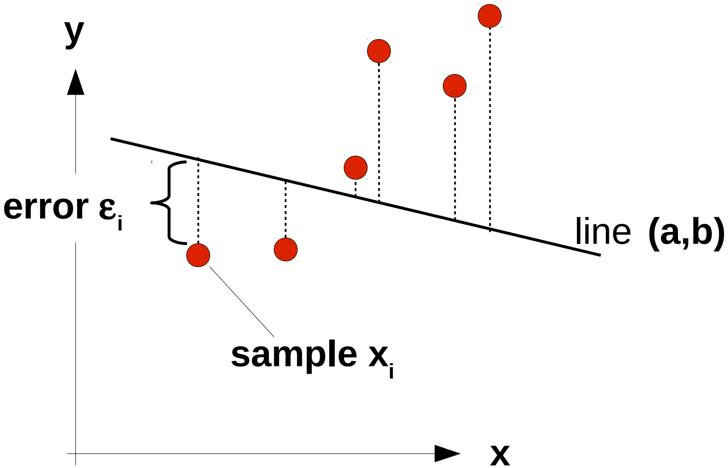
$$E(\theta) = \sum_{i=1}^{n} \overbrace{\Big(\underbrace{a \cdot x_i + b}_{\mathcal{M}(x_i)} - y_i \Big)^2}^{\text{error } \epsilon_i^2}$$

- The **best line** is the one that **minimizes** E:

$$(a^*, b^*) = \arg\min_{\theta \in \mathbb{R}^2} \; E(\theta)$$

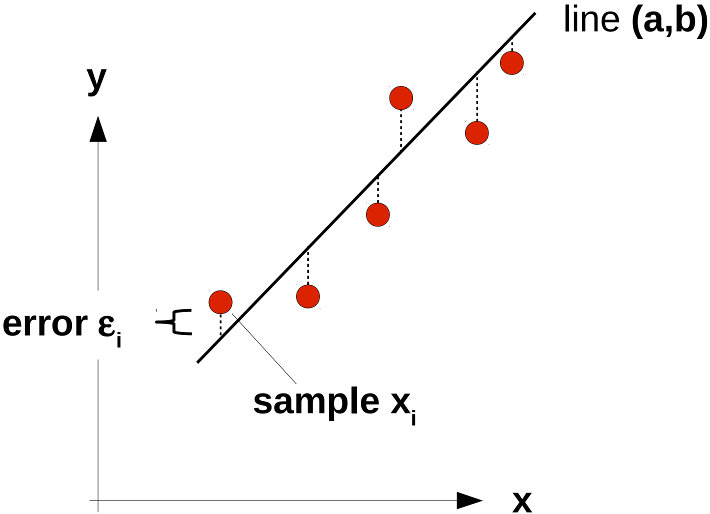
Machine Learning's Hello World

a bad line: The errors $\epsilon_1^2, ..., \epsilon_n^2$ are high $\rightarrow$ E is high



Machine Learning's Hello World

a good line: The errors $\epsilon_1^2, \ldots, \epsilon_n^2$ are low $\rightarrow$ E is low



Machine Learning's Hello World

We minimize E by settings its gradient to zero, followed by a bit of algebra:

$$\partial E/\partial b = \partial\Big(\sum_i (a \cdot x_i + b - y_i)^2\Big)/\partial b = 0$$

$$2 \cdot \sum_i (a \cdot x_i + b - y_i) = 0$$

$$n \cdot b = \sum_i y_i - a \cdot \sum_i x_i$$

$$b = \bar{y} - a\bar{x}$$

$$\partial E/\partial a = \partial\Big(\sum_i (a \cdot x_i + b - y_i)^2\Big)/\partial a = 0$$

$$2 \cdot \sum_i (a \cdot x_i + b - y_i) \cdot x_i = 0$$

$$\sum_i (a \cdot x_i + (\bar{y} - a\bar{x}) - y_i) \cdot x_i = 0$$

$$a\Big(\sum_i x_i^2 - \bar{x}\sum_i x_i\Big) = \sum_i y_i x_i - \bar{y}\sum_i x_i \qquad // \text{ :n}$$

$$a = s_{xy}/s_x^2$$

Machine Learning Terminology

- The problem above is a **regression problem**:
  It is about predicting a real-valued variable y.
- We call the points $(x_1, y_1), ..., (x_n, y_n)$ the **training samples**.
- We call our line function $\mathcal{M}(x) = a \cdot x + b$ the **model**.
- We call the process of estimating the model parameters (here, a and b) **training**.
- A typical training strategy is to formulate an error criterion (here, E) and **optimize** this criterion.
- In our example, we were able to pin down the solution by hand (we say: there is an *analytical* solution). In practice, optimization can be much trickier. Not all functions are easy to optimize.
- Therefore, learning is often done by **local search**.

Machine Learning Terminology

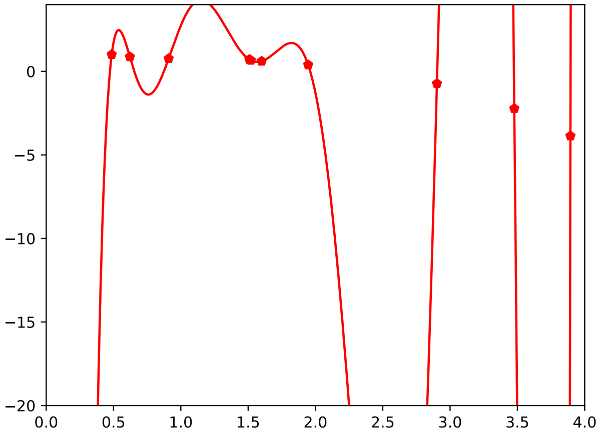
- Obviously, picking the **right model** for the right data is tricky. It is the core problem in machine learning, actually.
- **Example**: Would those be better models?

$$\mathcal{M}_{a,b,c,d}(x) = a + b \cdot sin(c \cdot x + d)$$
$$\mathcal{M}_{a,b,c,d,e}(x) = a + b \cdot sin(c \cdot x + d) + e \cdot x$$
$$\mathcal{M}_{a_0,a_1,\ldots,a_{100}}(x) = \sum_{i=0}^{100} a_i \cdot x^i$$

Machine Learning Terminology

Example: Runge's Phenomenon

- Fitting an 8-degree polynomial to 9 points



Machine Learning Terminology

"The real value of a scientific explanation lies not in its ability to explain (what one has already seen), but in predicting events that have yet to (be seen)"

(Blumer et al. 1987)

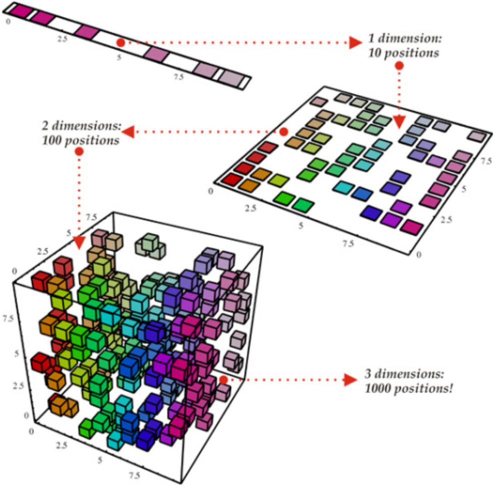
"With four parameters I can fit an elephant and with five I can make him wiggle his trunk."

(John von Neumann)

- Our polynomial model works well on the training data but **generalizes poorly** to novel data. The model **overfits**.
- We call $\theta = (a, b)$ the model's **parameters**.
  In practical models there may be thousands of parameters.
- Overfitting is usually more severe...
  - ... the more parameters a model has.
  - ... the fewer training samples we have.

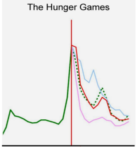
Machine Learning: The Curse of Dimensionality image from [10]

- Overfitting is also more severe the **more dimensions** we have.
- **Reason**: As the number of dimensions increases, we require *more and more data* to populate our input space!

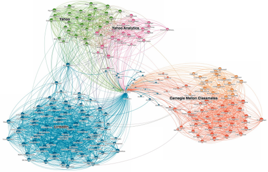


Machine Learning: Challenges images from [14] [9] [2] [1]

Regression



Clustering / Segmentation

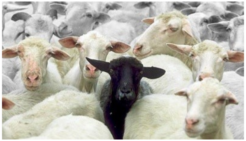


Recommendation

Classification

Data Reduction

Anomaly Detection



Machine Learning: Small Example image from [15]

	A	B	C	D	E	F	G	H	I	J	K	L
1	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7925		S
5	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
7	6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
8	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
9	8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21075		S
10	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333		S
11	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708		C
12	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
13	12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S
14	13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.05		S
15	14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31275		S
16	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	350406	7.8542		S
17	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	248706	16		S
18	17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29125		Q
19	18	1	2	Williams, Mr. Charles Eugene								
20	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Van								
21	20	1	3	Masselmani, Mrs. Fatima								
22	21	0	2	Fynney, Mr. Joseph J								
23	22	1	2	Beesley, Mr. Lawrence								
24	23	1	3	McGowan, Miss. Anna "Annie"								
25	24	1	1	Sloper, Mr. William Thompson								
26	25	0	3	Palsson, Miss. Torborg Danira								
27	26	1	3	Asplund, Mrs. Carl Oscar (Selma Augusta Emi								
28	27	0	3	Emir, Mr. Farred Chehab								
29	28	0	1	Fortune, Mr. Charles Alexander								
30	29	1	3	O'Dwyer, Miss. Ellen "Nellie"								
31	30	0	3	Todoroff, Mr. Lalio								
32	31	0	1	Uruchurtu, Don. Manuel E								
33	32	1	1	Spencer, Mrs. William Augustus (Marie Eugen								
34	33	1	3	Glynn, Miss. Mary Agatha								
35	34	0	2	Wheadon, Mr. Edward H								
36	35	0	1	Meyer, Mr. Edgar Joseph								
37	36	0	1	Holverson, Mr. Alexander Oskar								
38	37	1	3	Mamee, Mr. Hanna								
39	38	0	3	Cann, Mr. Ernest Charles								
40	39	0	3	Vander Planke, Miss. Augusta Maria								
41	40	1	3	Nicola-Yarred, Miss. Jamila								
42	41	0	3	Ahlin, Mrs. Johan (Johanna Persdotter Larsso								
43	42	0	2	Turpin, Mrs. William John Robert (Dorothy Ann								
44	43	0	3	Kraeff, Mr. Theodor								

Some more ML Terminology

- ML's goal is to make predictions about real-world **objects**
  - **SPAM filtering**: objects = e-mails
  - **route planning**: objects = routes to drive
- Our goal is to make a prediction regarding an object.
  We call this prediction a **label** or **target**
  - **SPAM filtering**: label $\in \{spam/ham\}$
  - **route planning**: label $\in \mathbb{R}_0^+$ ($=$ *time to destination*)
- We describe each object by a number of **features**
  - **Titanic**: features = gender, ticket price, passenger class, ...
- The collection of the features describing an object
  is called a **feature vector** (and usually denoted with $\mathbf{x}$).
- Features can be **categorial** or **numerical**.
- In case of numerical features, the object can be interpreted
  as a **point** in a (high-dimensional) space!

ML: Feature Engineering / Feature Extraction

Often, we **prepare** the input data before applying ML

1. Features may be **missing**, i.e. **x** is *incomplete*.
   Approach: estimate missing values (*imputation*)
2. **Categorical** features may have to be transformed into numerical ones, typically by introducing **dummy variables** *(one-hot encoding)*

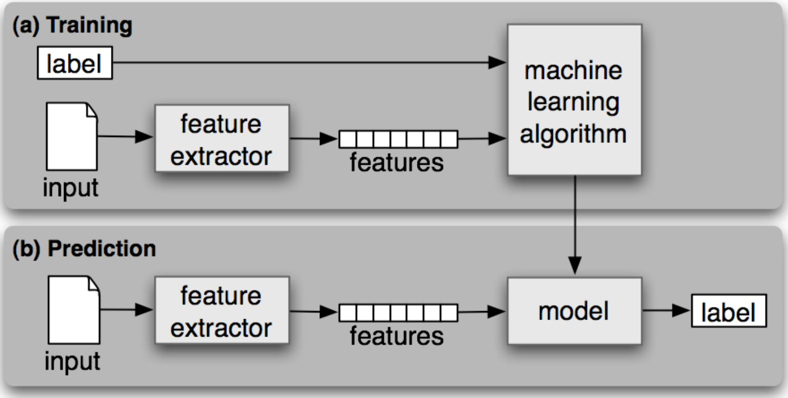
	PS	color
Prof. Ulges' car	70	white
Prof. Ulges' wives' car	690	red

		dummy variables	
PS	is_green	ist_silver	ist_red
73	0	0	0
690	0	0	1

3. Often, we discard **outliers**
4. Often, we try to pre-select **'important'** features.

A Basic ML System Pipeline image from [?]

1. We **train** the system in an off-line phase, obtaining a **model**
2. We **apply** the system in an on-line phase



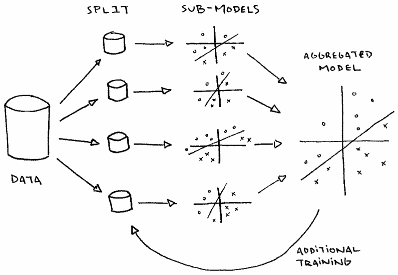
Types of Machine Learning

- **classification vs. regression**
  in classification, the labels $y_1, ..., y_n$ are categorical.
  In regression, they are real-valued.
- **supervised learning**
  learning from samples $\mathbf{x}_1, ..., \mathbf{x}_n$ with labels $y_1, ..., y_n$
- **unsupervised learning**
  learning only from samples $\mathbf{x}_1, ..., \mathbf{x}_n$, no labels
- **semi-supervised learning**
  learning from samples $\mathbf{x}_1, ..., \mathbf{x}_n$, some with labels
- **active learning**
  the system can pick which samples to label
- **ensemble learning**
  ... is about combining learners for a more robust decision
- **reinforcement learning**
  ... is about learning from feedback instead of labels
- ...

Types of Machine Learning (cont'd) image from [5]

Online Learning

- Perspective so far: **batch learning** *(train off-line, apply on-line)*
- What if data is too large for memory? Does the model allow a **sharding** of training data? *example: linear regression*



- What if data come in dynamic streams? ($\rightarrow$ **data stream mining** / online learning)

Outline

ML vs. Statistics image from [11]

"Statistics is the science of learning from data. Machine Learning is the science of learning from data. These fields are identical in intent, although they differ in their history, conventions, emphasis and culture."

("The rise of the machines" – Larry Wasserman, CMU)

ML vs. Statistics

- The term "machine learning" has a negativ overtone in statistics *("Why do Statisticians hate us?")*
- Both fields address the same problems with similar methods

Machine learning	Statistics
network, graphs	model
weights	parameters
learning	fitting
generalization	test set performance
supervised learning	regression/classification
unsupervised learning	density estimation, clustering
large grant = $1,000,000	large grant = $50,000
nice place to have a meeting: Snowbird, Utah, French Alps	nice place to have a meeting: Las Vegas in August

ML: Relation to other Fields (cont'd)

ML vs. Artificial Intelligence (AI)

- Machine Learning is a subfield of **subsymbolic** AI
- The other part of AI – **symbolic** AI – is about logic, discrete search and rule-based inference.

ML vs. Optimization

- Machine learning techniques usually employ optimization
- "Machine learning = optimization + **generalization**"

ML vs. Data Mining

- Data mining focuses on **exploratory data analysis** *(by a human expert)*, machine learning on **automatic inference**.
- Data mining experts use machine learning, and machine learning experts use data mining.

Outline

1. Motivation

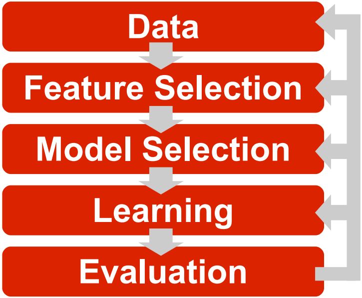
2. Machine Learning Basics

3. Machine Learning vs X

4. Benchmarking ML Systems

Why is Benchmarking so important?

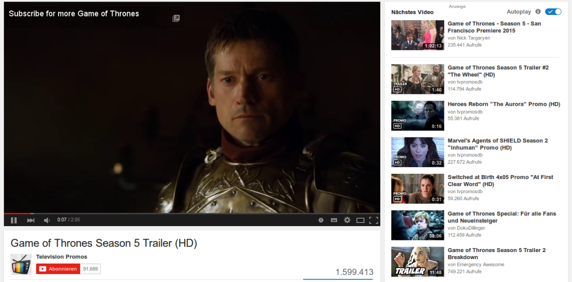
- **Machine Learning** = iterative re-design of...
  - data
  - features
  - models
  - parameters
- **Key Driver**: **Evaluation / Benchmarking**



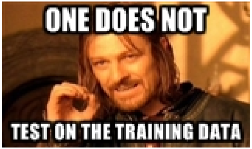
Benchmarking ML Systems: Motivation image from [7]

Example: YouTube recommender

- We add a **“user history” feature** to the recommender.
- For example, the feature could merge some videos from the user’s history into the recommendations.
- YouTube redirects some volume of **traffic** to the new system.
- YouTube benchmarks the new system against the old one.



Benchmarking: Ground Rule image from [6]



> *"The most common mistake among machine learning beginners is to test on the training data and have the illusion of success."*

(P. Domingos, *A few Useful Things to Know about Machine Learning*)

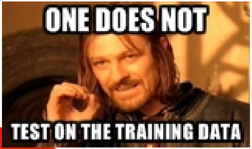
Why?

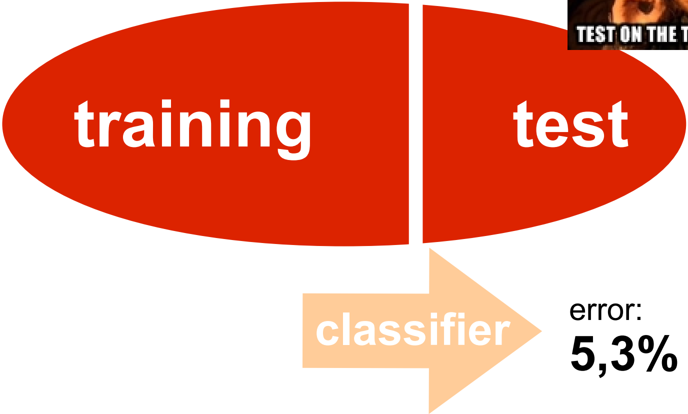
- All classifiers are prone to **overfitting**
- Achieving perfect accuracy on the training samples is quite simple *(by simply memorizing them)*.
- It is the generalization to **new data** that matters!
- When building classifiers, <u>always set some test data aside!</u>

Adhering to the Ground Rule is trickier than you think!

- **Example**: neural network training
- **Example**: standard datasets in research

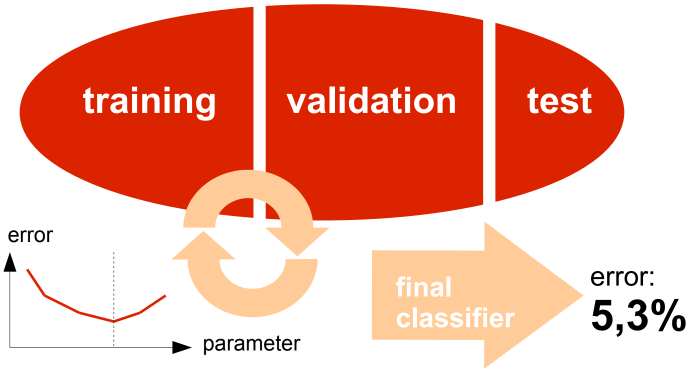
Machine Learning: Benchmarking





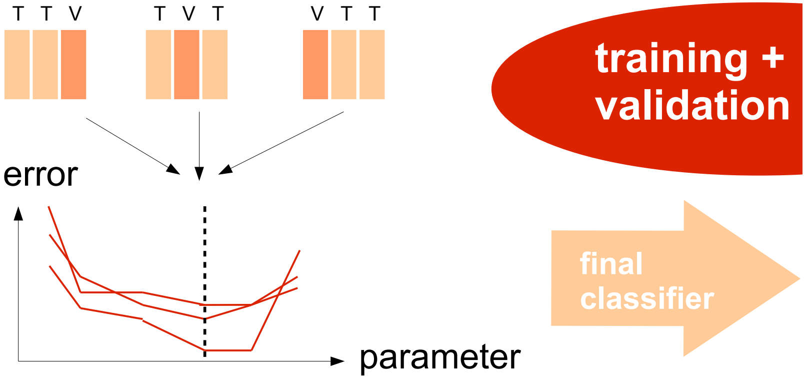
- We separate our dataset into **training and testing data**
- **Tip 1**: choose 'just enough' test data, use the rest for training
- **Tip 2**: use a roughly balanced class distribution

Machine Learning: Benchmarking



- Some of a classifiers' parameters are usually **learned**, other **(free)** parameters are set **manually**
- Typical approach: **grid search**
- **Example**: decision tree (later today) $\rightarrow$ *parameter* $=$ *depth*
- **Approach**: train $\rightarrow$ **validate** $\rightarrow$ test

Machine Learning: Benchmarking



- If there is **small training data**, we apply **cross-validation**: Split the data into subsets ("*folds*"), train/validate multiple times, and average the results
- Extreme case: **leave-one-out validation**

Accuracy Measures: Loss

- In the following, we focus on benchmarking **classification systems**
- Basic idea: Count the samples for which the classification system chooses the correct/wrong class.
- But: Some misclassifications are **more expensive** than others!
- **Example: spam filtering**
  - misclassifying ham as spam: really bad
  - misclassifying spam as ham: not so bad
- We model the cost of specific misclassifications via a **loss function L** (assuming possible classes $1, ..., C'$):

$$L : \{1, ..., C'\} \times \{1, ..., C'\} \rightarrow \mathbb{R}$$

- The loss function is often provided by a **business expert**.

Accuracy Measures: Loss

The loss function is usually modeled as a $C' \times C'$ matrix:

$$\boxed{L(i,j) := \text{cost for misclassifying a sample from class } i \text{ as class } j}$$

$$\begin{array}{cc} & \text{picked class } j \\ \text{true class } i & \begin{pmatrix} L(1,1) & L(1,2) & \dots & L(1,C') \\ L(2,1) & L(2,2) & \dots & L(2,C') \\ \dots & & & \\ L(C',1) & L(C',2) & \dots & L(C',C') \end{pmatrix} \end{array} \qquad \begin{array}{c} \begin{pmatrix} 0 & 1 & \dots & 1 \\ 1 & 0 & \dots & 1 \\ \dots & & & \\ 1 & 1 & \dots & 0 \end{pmatrix} \\ \text{zero-one-loss} \end{array}$$

Remark

- One standard loss function is **zero-one loss** (right): any error costs 1

Average Loss, Error Rate, Accuracy

- Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be a set of **test samples** with **labels** $y_1, \ldots, y_n \in \{1, \ldots, C'\}$, and $\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n) \in \{1, \ldots, C'\}$ the **predictions** of our classifier.
- We measure the classifier's **average loss**:

$$\frac{1}{n} \sum_{i=1}^{n} L(y_i, \phi(\mathbf{x}_i))$$

- In case of **zero-one-loss**, this is equivalent to the **error rate**:

$$\frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\phi(\mathbf{x}_i) \neq y_i}$$

- The error rate's counterpart is the (classification) **accuracy**:

$$\frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\phi(\mathbf{x}_i) = y_i}$$

- Obviously: **accuracy** $= \mathbf{1} -$ **error rate**

The Confusion Matrix image from [8]

- Sometimes, it is interesting to check which classes the classifier confuses most often.
- Therefore, we define the **confusion matrix** $C \in \mathbb{R}^{C' \times C'}$ with

$$C_{jk} := \#\{\, i \in \{1, \dots, n\} \mid y_i = j \wedge \phi(\mathbf{x}_i) = k\}$$

Example

		Estimated Emotion							
		Anger	Boredom	Disgust	Fear	Happiness	Sadness	Neutral	*Emotion Recog. Rate*
True Emotion	Anger	**19**	0	2	0	**3**	0	0	79.2%
	Boredom	1	**8**	1	1	0	1	**7**	42.1%
	Disgust	0	1	**6**	0	1	0	3	54.5%
	Fear	1	3	2	**7**	2	0	1	43.8%
	Happiness	**3**	0	3	2	**5**	0	2	33.3%
	Sadness	0	0	0	0	0	**14**	0	100.0%
	Neutral	0	**5**	1	0	0	0	**13**	68.4%
	HMM Recog. Rate	79.2%	47.1%	40.0%	70.0%	45.5%	93.3%	50.0%	

True/False Positives/Negatives

- **Binary classifiers** are frequent in practice! We can think of them as **detectors** *(information retrieval, spam filtering, medical expert systems, face detection, fraud detection, ...)*
- Often, binary classification problems are highly imbalanced, with very **few positive** samples vs. **many negative** ones.
- For binary classifiers, the terms in the confusion matrix have specific names.

$$C = \begin{pmatrix} TP & FN \\ FP & TN \end{pmatrix}$$

- TP = true positives
- TN = true negatives
- FP = false positives
- FN = false negatives

True/False Positive/Negative Rate

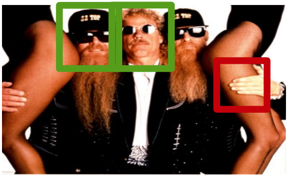
Common Quality Measures

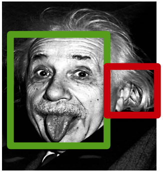
- error rate $:= \frac{FP+FN}{FP+FN+TP+TN}$
- false positive rate (FPR) $:= \frac{FP}{FP+TN}$
- false negative rate (FNR) $:= \frac{FN}{TP+FN}$
- true positive rate (TPR) $:= \frac{TP}{TP+FN} \left(= 1 - \text{FNR}, \text{aka sensitivity}\right)$
- true neg. rate (TNR) $:= \frac{TN}{FP+TN} \left(= 1 - \text{FPR}, \text{aka specificity}\right)$

Quality Measures from Information Retrieval

- recall $:= \frac{TP}{TP+FN} \quad \left(= \text{sensitivity}\right)$
- precision $:= \frac{TP}{TP+FP}$

TPs+FPs+FNs+TNs: Do-it-Yourself



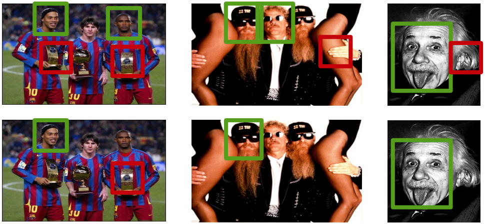


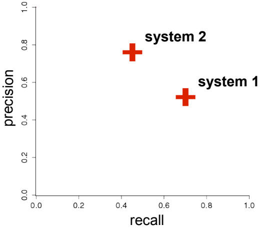
- false positive rate =
- false negative rate =

- recall =
- precision =

- error rate =

The F-Measure





Which of the two classifiers is better?

- We introduce the **F-measure** as a global quality indicator.
- Given recall r and precision p, we define:

$$F(r, p) := 2 \cdot \frac{p \cdot r}{p + r}$$

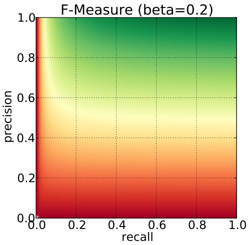
- The F-measure is the **harmonic mean** of precision and recall.

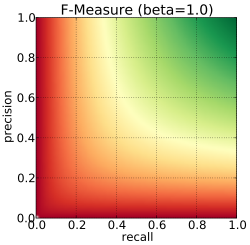
The F-Measure

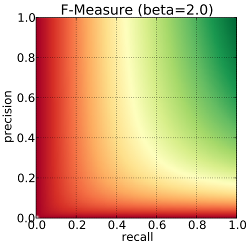
- We can extend the F-measure with a parameter β to emphasize precision or recall:

$$F_\beta(r, p) := \left(1 + \beta^2\right) \cdot \frac{p \cdot r}{\beta^2 \cdot p + r}$$

- The higher β, the **more important is recall** ($\beta = 1 \rightarrow$ recall and precision are equally important)







Score-based Benchmarking

- Often, classifiers compute an internal **score** for each sample, $\mathbf{s}(\mathbf{x})$, and threshold these scores to obtain their final decision:

$$\phi(\mathbf{x}) = 1 \quad \Leftrightarrow \quad s(\mathbf{x}) \geq T$$

- The **lower** the threshold T, the **more positives** the classifier generates.
- How do we benchmark the classifier **independent** of T?
- **Goal**: Measure how well the **scores discriminate** positive samples from negative ones.

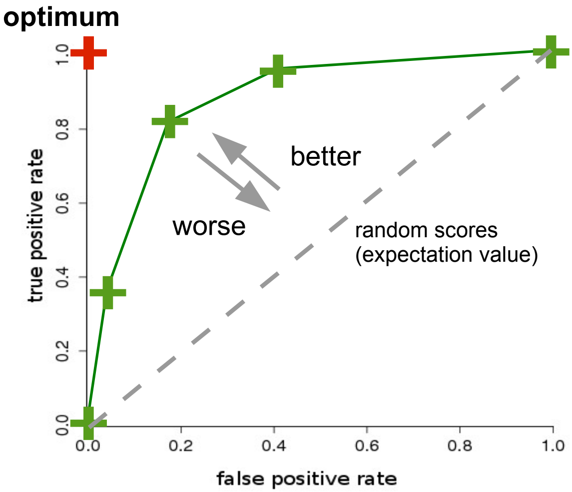
Approach

- We **vary the threshold** T and measure the FPR and FNR for each value of T.
- We obtain a curve, the **Receiver-Operator-Characteristic (ROC)** curve.

T	FPR	TPR
1.0	0.00	0.00
0.9	0.05	0.38
...	...	...
0.1	0.41	0.96
0.0	1.00	1.00

The ROC-Curve: Example

T	FPR	TPR
1.0	0.00	0.00
0.9	0.05	0.38
…	…	…
0.1	0.41	0.96
0.0	1.00	1.00



The ROC-Curve: Equal Error

We observe: When **lowering the threshold T**, the ...

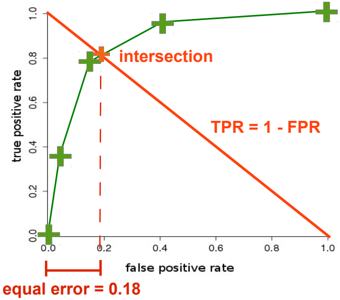
- ... false-positive-rate (FPR) increases from 0 to 1
- ... false-negative-rate (FNR) decreases from 1 to 0.

We derive a single quality measure from the ROC curve:
The **Equal Error Rate (EER)**.

- For some value T^* holds:

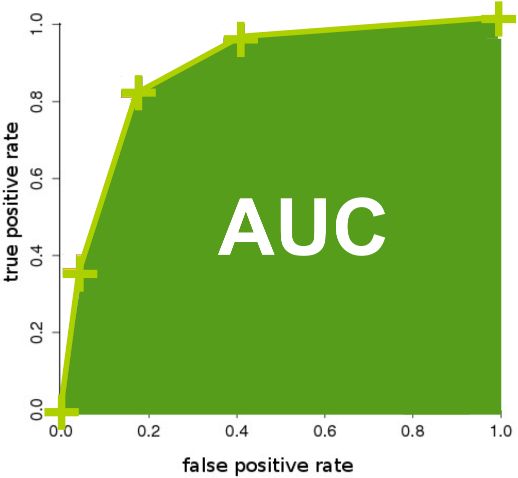
$$FNR \approx FPR.$$

- This error rate is the EER.
- Graphically, we obtain the EER by intersecting the ROC curve with $TPR = 1 - FPR$.



The ROC-Curve: AUC

We derive another quality measure from the ROC curve:
The **Area Under Curve (AUC)**



RPC Curves

- For retrieval/recommender systems, we often derive a **curve similar** to the ROC curve, but based on **precision and recall**.
- Varying the threshold T corresponds to traversing the **ranked result list** from top to bottom.
- Whenever we encounter a **new positive**, we add a new point with coordinates $(recall, precision)$ to the curve.

Example

1 0 0 1 1 0 0 0 1 0 0 0 0

top result

Whenever reaching a new positive, we add a point to the curve

last result

recall	precision
0.25	1.00
0.50	0.50
0.75	0.60
1.00	0.44

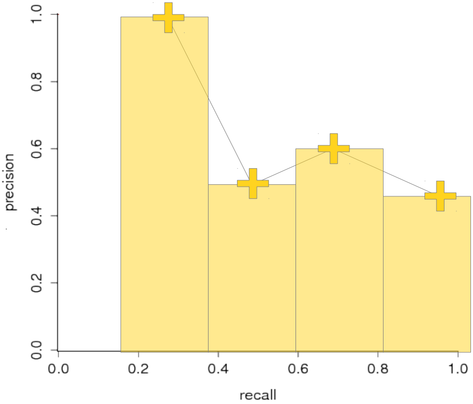
RPC Curves

The resulting graph is called
the **Recall-Precision-Curve (RPC)**.

recall	precision
0.25	1.00
0.50	0.50
0.75	0.60
1.00	0.44

RPC Curves → Average Precision

We derive another indicator from the RPC curve:
The **Average Precision**.



References I

[1] Animali i animaletti: owca nie tylko czarna.
http://studiaparlaama.pl/animali-i-animaletti-owca-nie-tylko-czarna/ (retrieved: Oct 2016).

[2] Axay Pithava: Perceptual Mapping Techniques.
http://www.slideshare.net/AxayPithava/perceptual-mapping-techniques (retrieved: Oct 2016).

[3] Boston Dynamics: Atlas, the Next Generation.
https://www.youtube.com/watch?v=rVlhMGQgDkY (retrieved: Oct 2016).

[4] Brizzle born and Bread.
https://www.flickr.com/photos/brizzlebornandbred/5292576151/ (retrieved: Oct 2016).

[5] Conjecture: Scalable Machine Learning in Hadoop with Scalding.
https://codeascraft.com/2014/06/18/conjecture-scalable-machine-learning-in-hadoop-with-scalding/ (retrieved: Oct 2016).

[6] The fellowship of the ring.

[7] Game of Thrones Season 5 (Trailer HD).
https://www.youtube.com/watch?v=A0pLbTXPHng (retrieved: Oct 2016).

[8] Game of Thrones Season 5 (Trailer HD).
http://webee.technion.ac.il/ (retrieved: Oct 2016).

[9] Google opens Cloud Vision API beta, world + dog asked to try it.
http://www.theregister.co.uk/2016/02/19/google_opens_cloud_vision_api_beta/ (retrieved: Oct 2016).

[10] Haifeng Li's (Blog): There is no big data in machine learning.
https://haifengl.wordpress.com/2016/02/29/there-is-no-big-data-in-machine-learning/ (retrieved: Oct 2016).

References II

[11] Machine Learning in R: Supervised Classification.
http://www.rmanchester.org/presentations/2016/02/ManchesterR_-_Machine_learning_in_r_-_Tom_Liptrot_-_20160229.pdf (retrieved: Oct 2016).

[12] Quartz: Tesla thinks it can prevent the next car crash by seeing the world with radar instead of cameras.
qz.com (retrieved: Oct 2016).

[13] Spam (Monty Python).
https://en.wikipedia.org/wiki/Spam_(Monty_Python) (retrieved: Oct 2016).

[14] The Value of a Professional Network?
https://www.linkedin.com/pulse/value-professional-network-daniel-tunkelang (retrieved: Oct 2016).

[15] 'Untergang der Titanic' Illustration von Willy Stöwer für die Zeitschrift Die Gartenlaube.
https://de.wikipedia.org/wiki/RMS_Titanic (retrieved: Oct 2016).

[16] Data never Sleeps X.0.
www.domo.com/blog/ (retrieved: Aug 2016).

[17] Google Research Blog: Building a deeper understanding of images.
https://research.googleblog.com/2014/09/building-deeper-understanding-of-images.html (retrieved: Aug 2016).