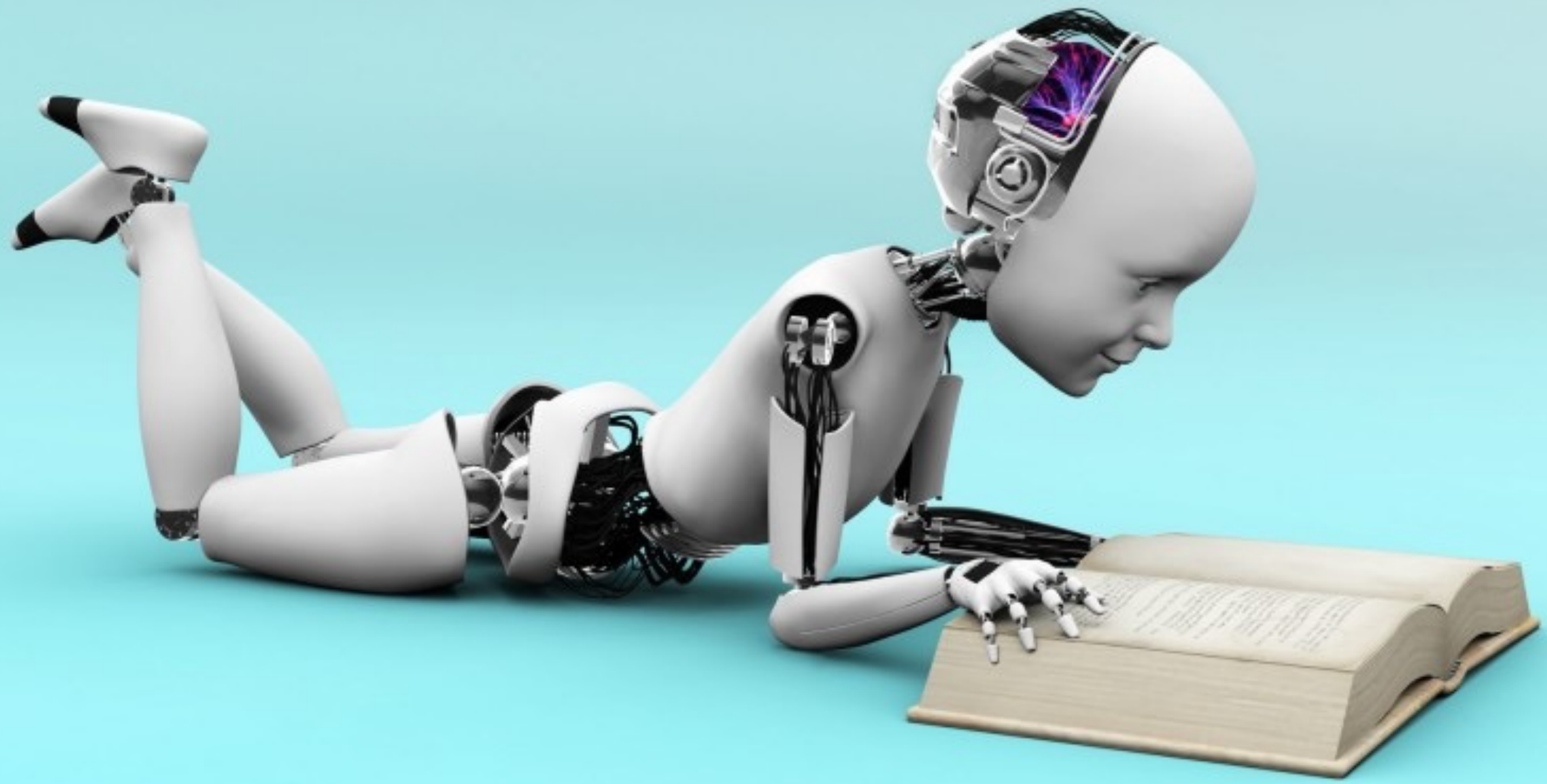
Crash Course on Machine Learning

August 22–24, 2016



simula
education

simula

Instructors

Valeriya Naumova

Senior Research Scientist

valeriya@simula.no

Interests: applied mathematics, learning theory, inverse problems

Arnaud Gotlieb

Chief Research Scientist

arnaud@simula.no

Interests: software engineering & testing

Lab Instructors

Timo Klock

PhD Student,
Biomedical Computing
Department

timo@simula.no

Carlo Leva

Senior Research Engineer,
Software Engineering
Department

carlo@simula.no

Sigurd Lekve

Summer Intern,
NTNU Student

sigurd.lekve@gmail.com

Organisers

Are Magnus Bruaset

Head of SSRI,
Section Director
Computing and Software

Elin Backe Christophersen

SSRI Advisor

Teodora Tufa

Executive Officer

Syllabus

Mon 22	**Morning**	Introduction to ML. Local Methods. Model Selection. Regularisation I.
	Afternoon	Machine Learning in Practice. Lab Work.
Tue 23	**Morning**	Regularisation II. Dimensionality Reduction, Variable Selection.
	Afternoon	Machine Learning Applications and Trends. Lab Work.
Wed 24	**Morning**	Basics in Logic. Programming in Logic. Decision Tree Induction.
	Afternoon	Version Space Search. Lab Work.
Thu 25	**Morning**	Open Problems Session and Discussion.
Sept 16 - May 17		Seminar and Mini-tutorial Series by Invited Guests (TBA).

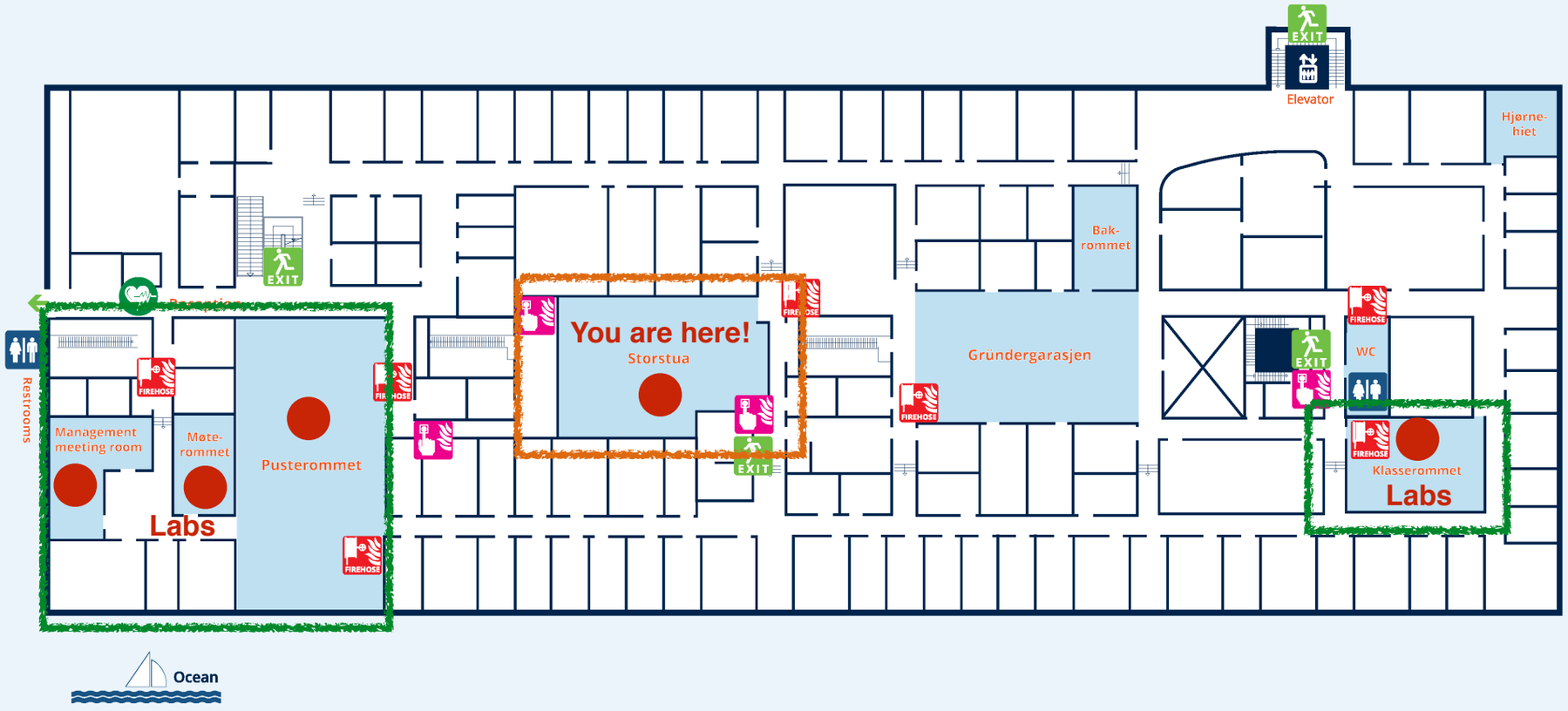
An introduction to *essential Machine Learning*:

- Concepts
- Algorithms

11:00-11:30 - Break
13:00-14:00 - Lunch
18:00-20:00 (Monday) - Social event

Practical Information

Important information in case of an emergency



Important information in case of an emergency

Below you will find the central emergency contact numbers to the fire brigade, police and ambulance, as well as the link to the official Norwegian website providing information to the general public before, during and after a crisis. In addition, you will find information on how any emergency event affecting Simula's premises, employees, students or guests should be notified internally.

Fire:	110
Police:	112
Ambulance:	113

Crisis information from the authorities: www.kriseinfo.no

Security at Technopolis (24h): 67 82 70 00

Emergency notification internally at Simula

If an emergency occurs that involves Simula's employees, students, guests or premises, after having alerted the appropriate emergency contact number, you should report directly to either of the contact persons below:

Contact	Role	Phone number
• Aslak Tveito	Managing Director	(+47) 906 87 348
• Kyrre Lekve	Deputy Managing Director	(+47) 934 24 311
• Marianne M. Sundet	Director of Administration	(+47) 900 18 483
• Halvard Moe	Technical Director	(+47) 928 10 236
• Maria Benterud	Administrative Manager	(+47) 954 95 616

simula

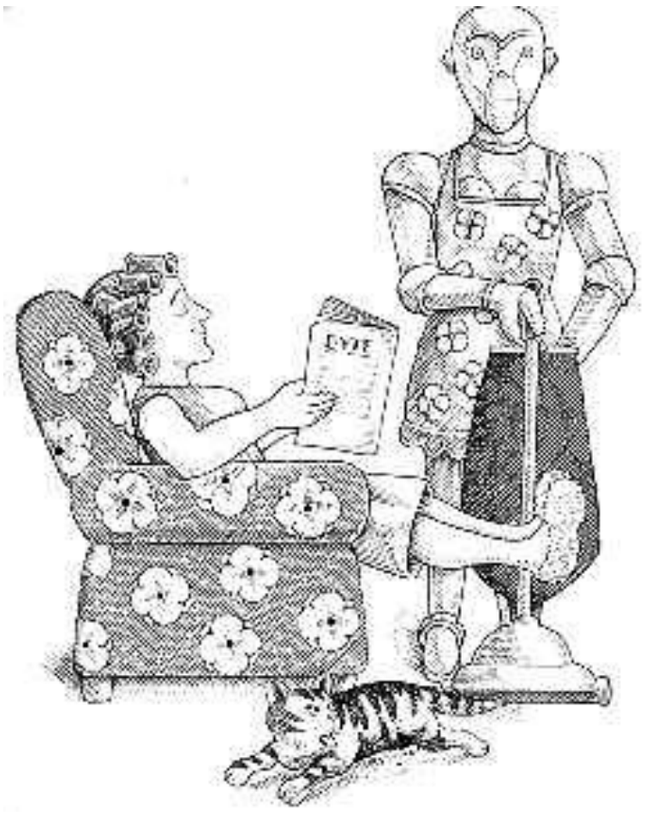
Course Material

Lectures and materials for lab sessions are available via

- Github vnaumova / ML_Course2016
- Dropbox

Software requirements for lab sessions

- Python with Matplotlib and Scipy packages (for Monday-Tuesday)
- SICStus Prolog 4.3 (for Wednesday)



Introduction to Machine Learning

Basic Concepts. Regularisation I

Crash Course on Machine Learning

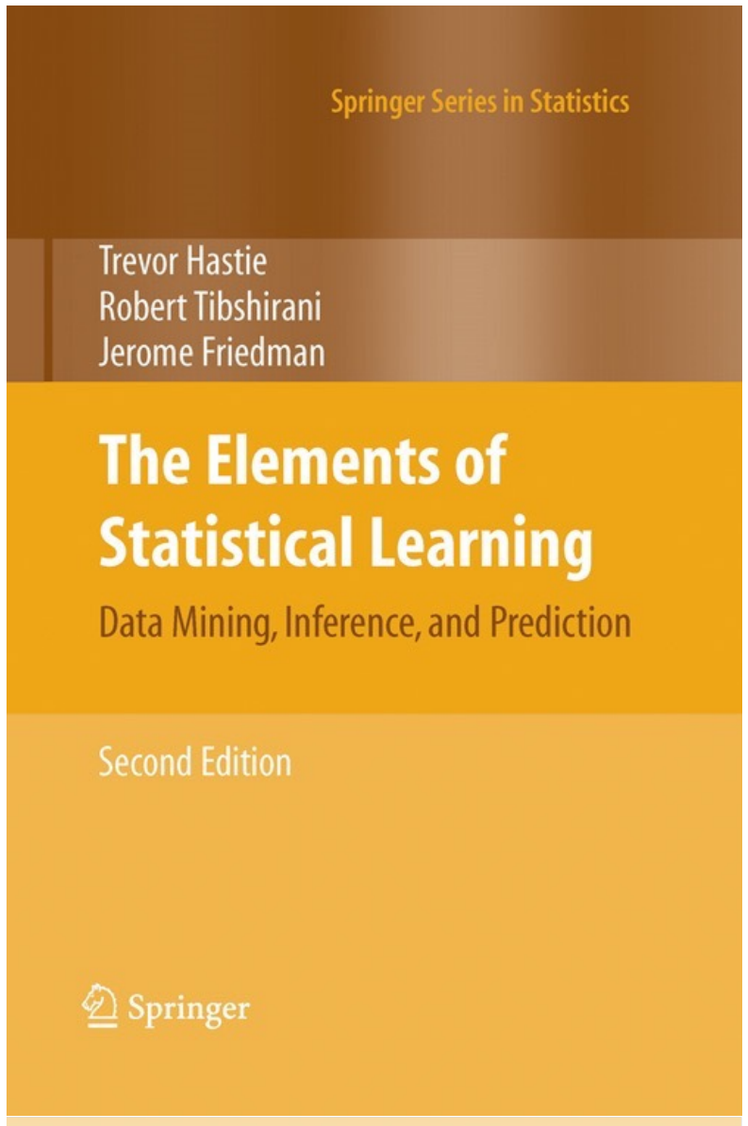
August 22-24, 2016

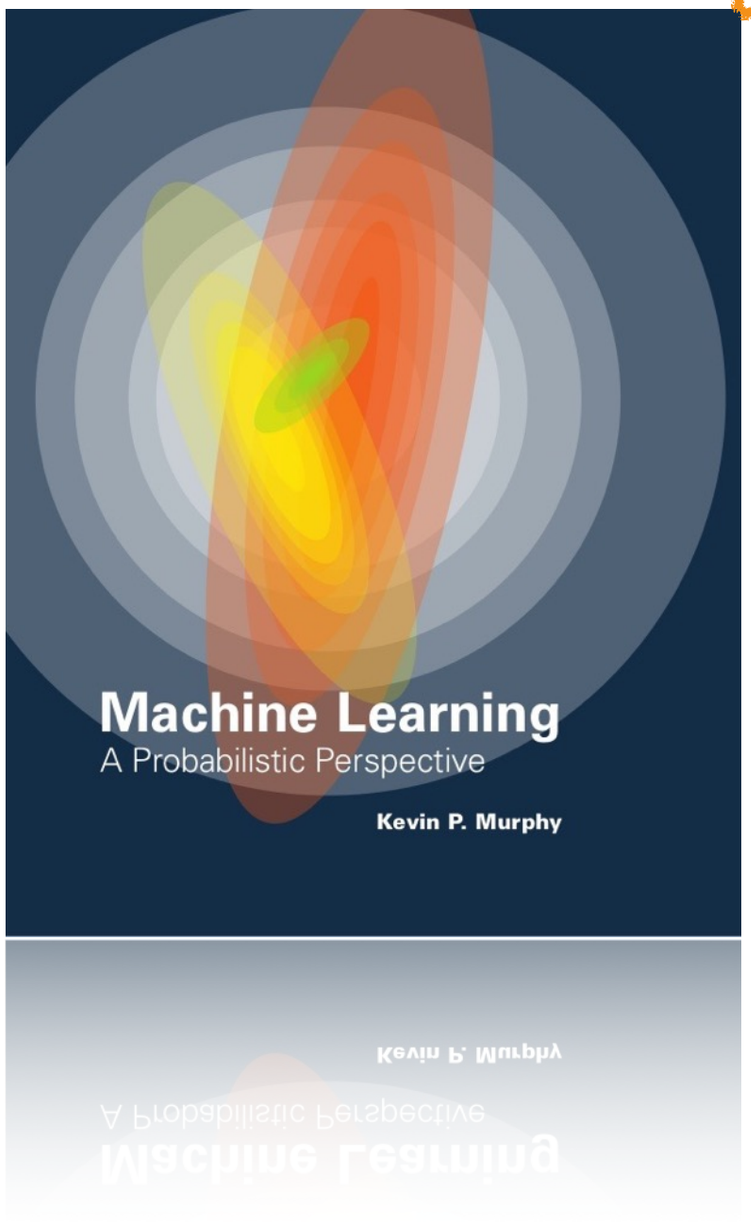
Valeriya Naumova and Arnaud Gotlieb

Simula Research Laboratory AS

Simula School of Research and Innovation (SSRI)

Literature





We are drowning in information and starving for knowledge.

Rutherford D. Roger, American Librarian

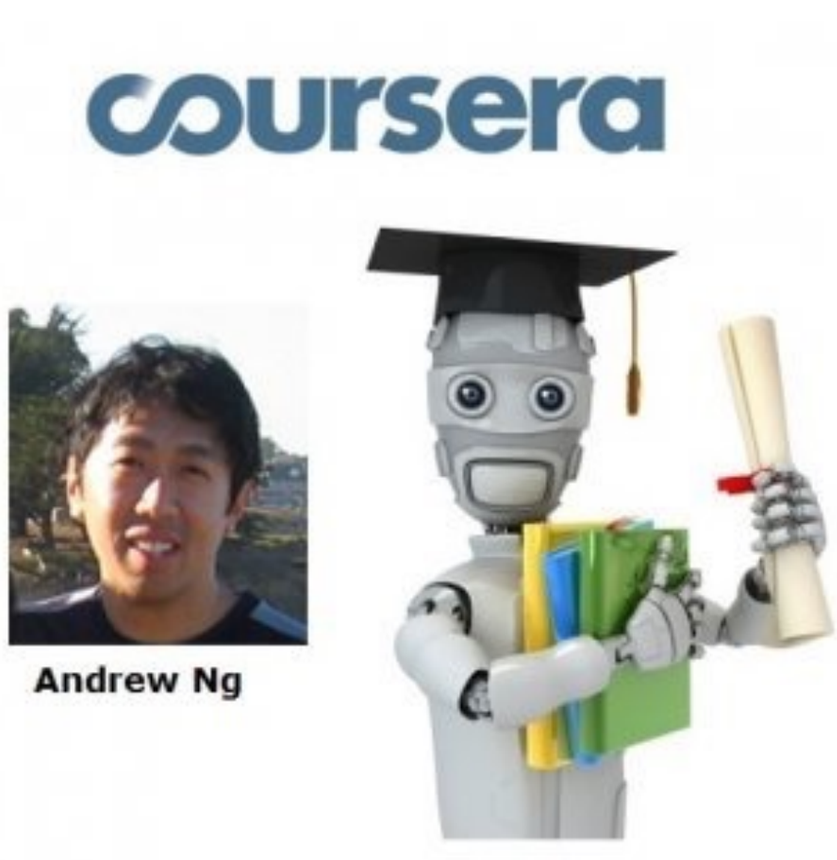
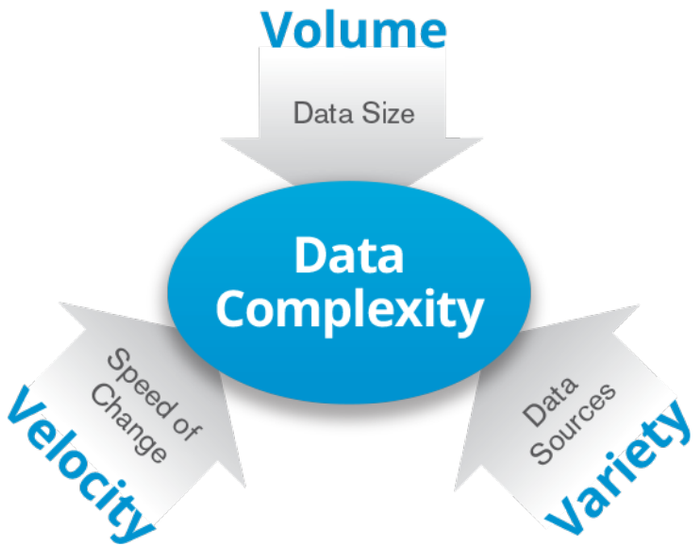
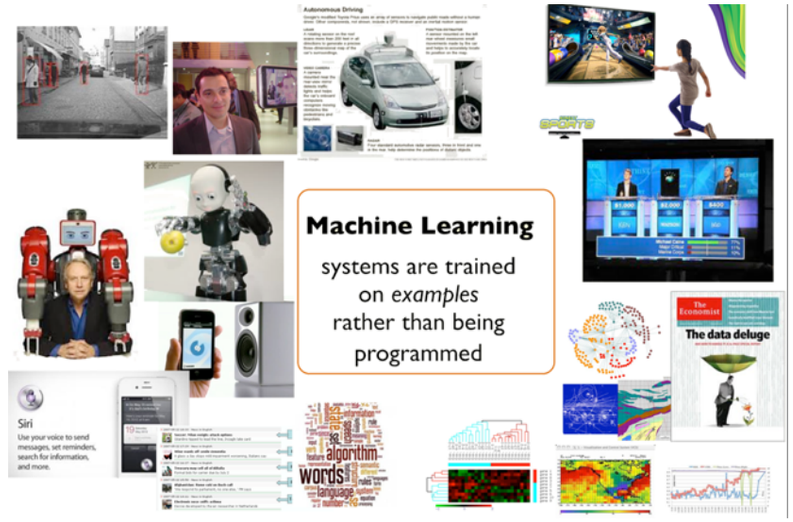


Table of Contents

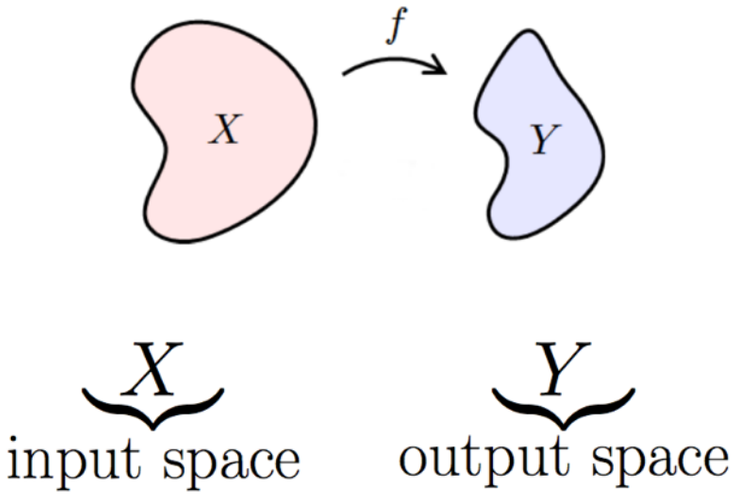
What is Big Data?



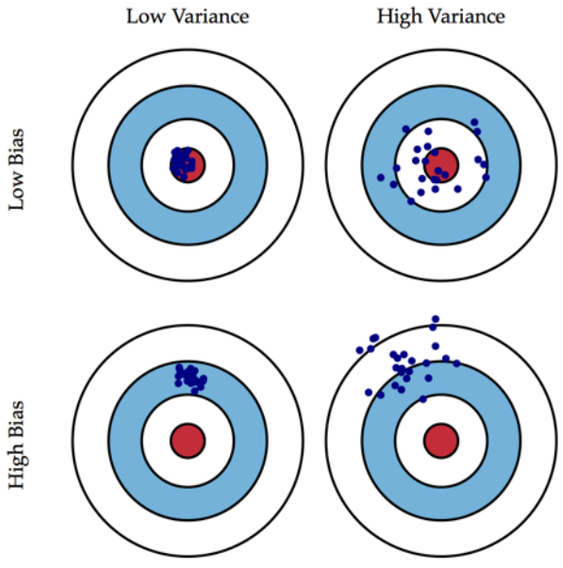
ML: History, Types, Applications



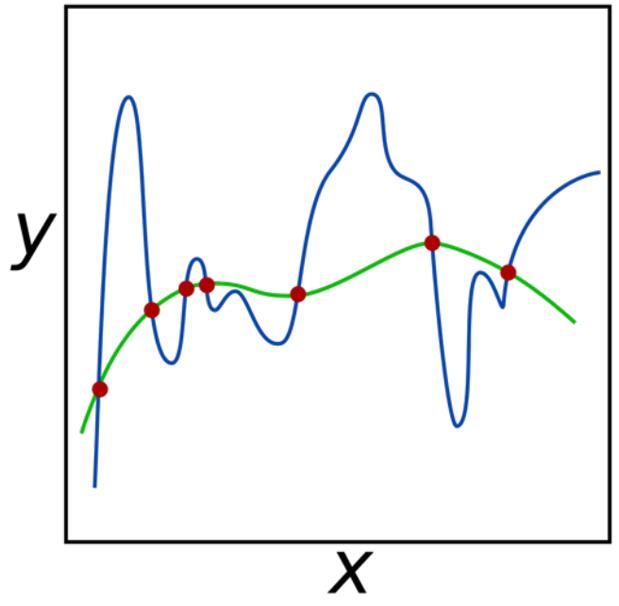
Basic Statistical Learning Theory



Local Methods & Bias-Variance



Regularisation I



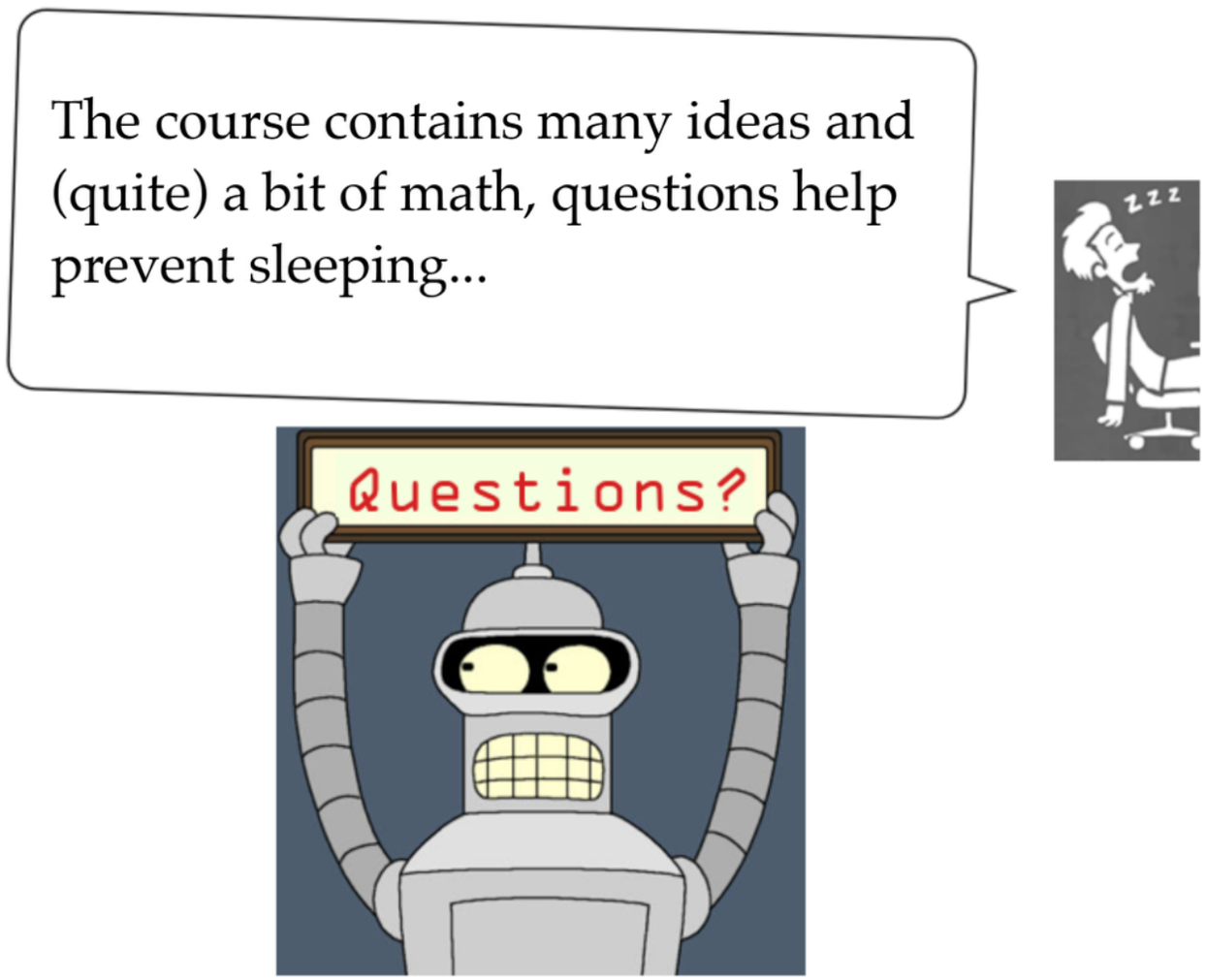
Colour Coding / Feedback

Really important stuff

Important stuff

Regular stuff

Let us know if you have comments, concerns, suggestions!



Data scientist ... 'sexiest job of the 21st Century'.

Harvard Business Review

In God we trust, all others bring data.

William Deming, Engineer & Statistician

The ~~four~~ five dimensions (V's) of Big Data

- A collection of large and complex data sets which are difficult to process using common database management tools or traditional data processing applications.

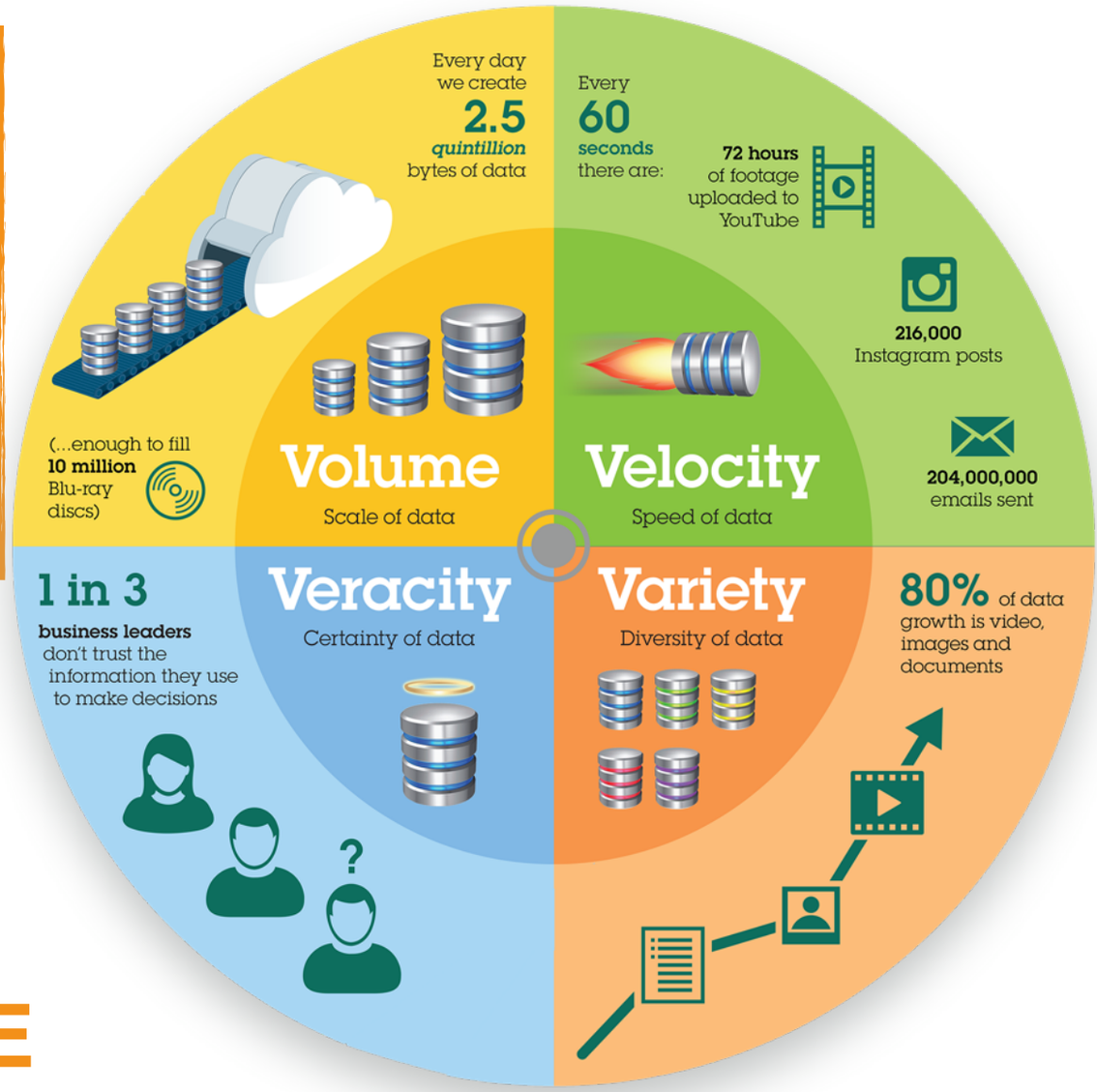
- "Big data refers to the tools, processes and procedures allowing an organisation to create, manipulate, and manage very large data sets and storage facilities".

according to zdnet.com

Big data is not just about size.

- Finds insights from complex, noisy, heterogeneous, longitudinal, and voluminous data.
- It aims to answer questions that were previously unanswered.

The challenges include capturing, storing, searching, sharing & analysing.



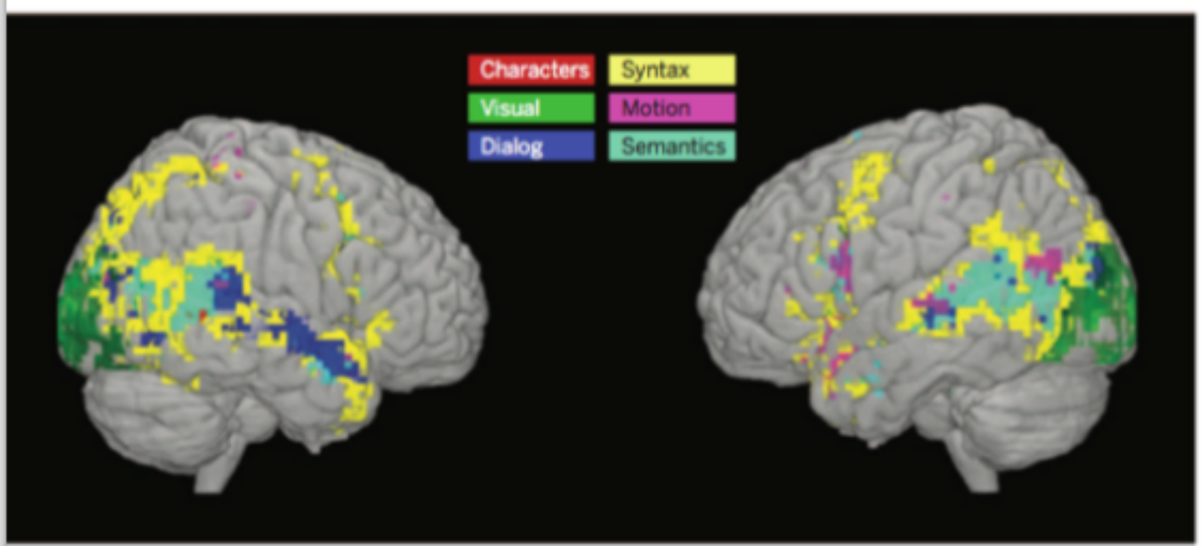
+ VALUE

Big Data Analysis - How?

→ machine / statistical learning

A breakthrough in machine learning would be worth ten Microsofts.

Bill Gates, Microsoft



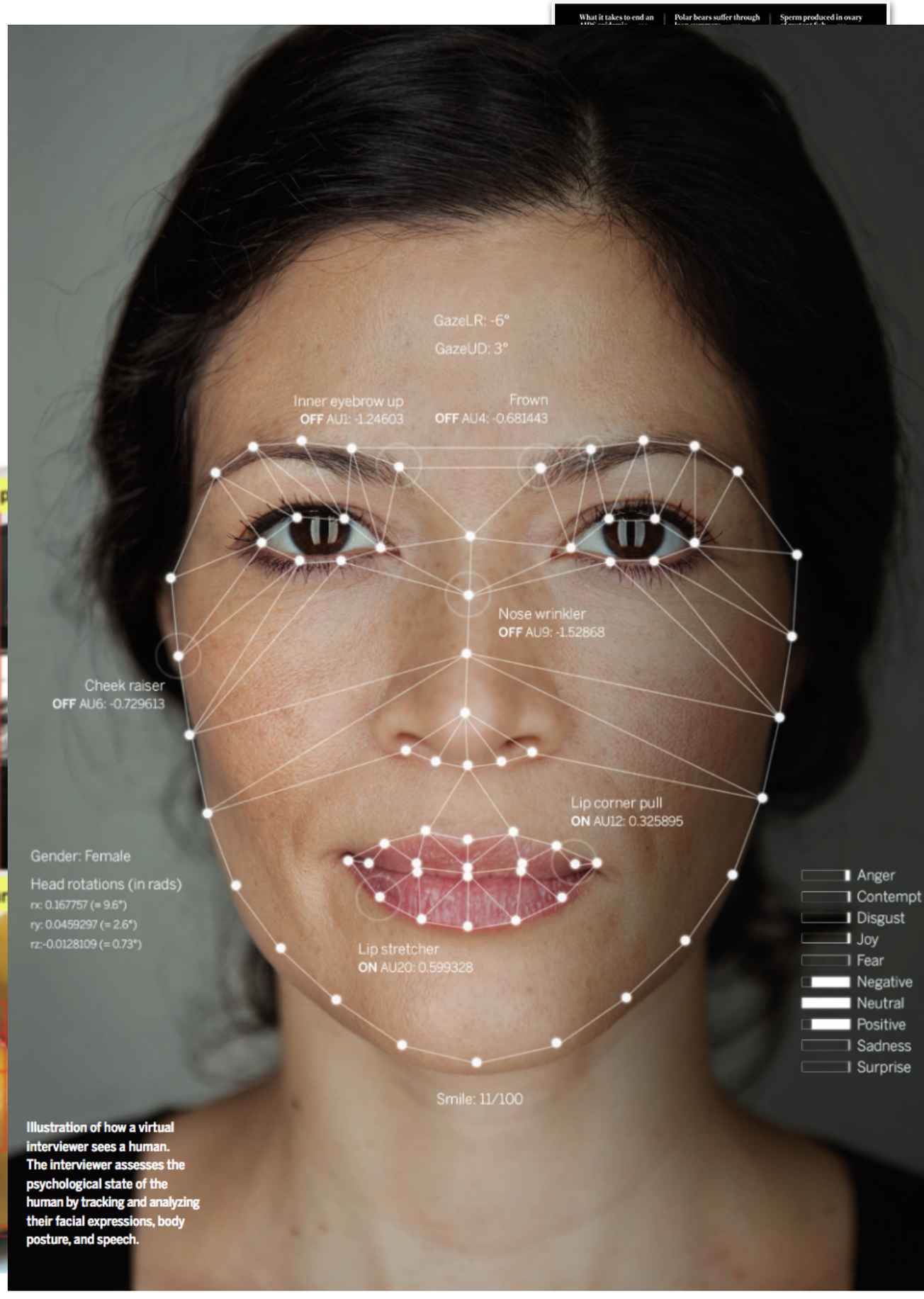


Illustration of how a virtual interviewer sees a human. The interviewer assesses the psychological state of the human by tracking and analyzing their facial expressions, body posture, and speech.

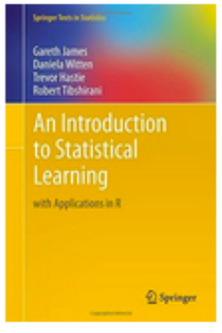
Science vol 349, 2015

Big Data Analysis : Amazon / Netflix

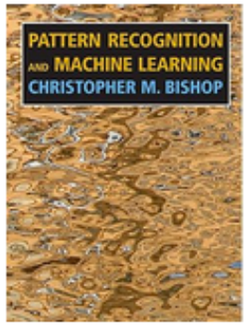
Netflix recommendations

Amazon Books

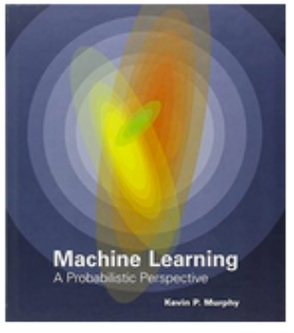
Customers Who Bought This Item Also Bought



An Introduction to Statistical Learning: with Applications in R…
› Gareth James
3
Gebundene Ausgabe
EUR 55,99 Prime



Pattern Recognition and Machine Learning (Information Science and…
Christopher Bishop
13
Gebundene Ausgabe
EUR 62,45 Prime

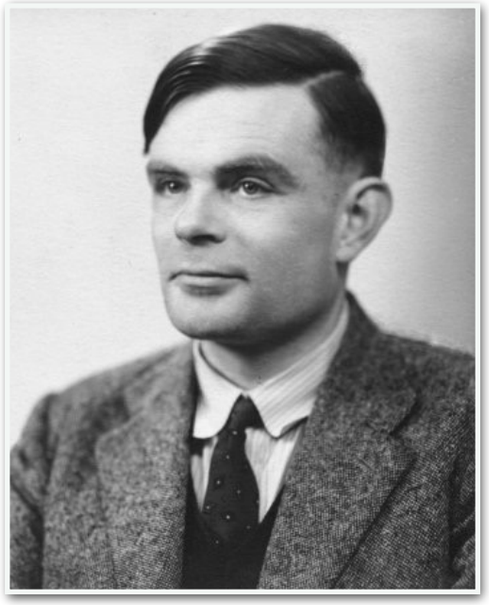


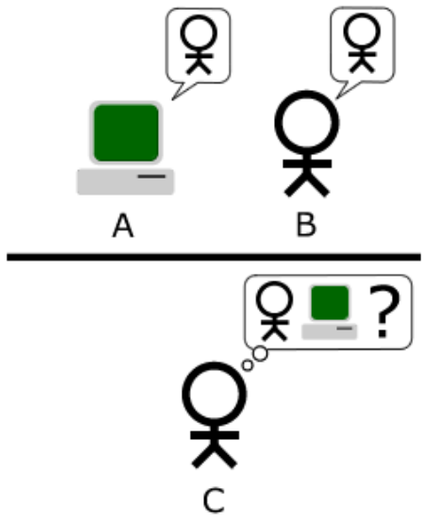
Machine Learning: A Probabilistic Perspective (Adaptive computation…
› Kevin P Murphy
1
Gebundene Ausgabe
EUR 79,53 Prime

Machine learning is the next Internet.

Tony Tether, DARPA

The Quest for AI. Birth of Dream





Intelligence: a Working Definition

- *abstract reasoning, knowledge acquisition, decision making.*
- *knowledge acquisition: memorisation vs learning.*

Alan Turing (1912–1954) proposed a test that calls for a panel of judges to review typed answers to any question that has been addressed to both a computer and a human. If the judges can make no distinctions between the two answers, the machine may be considered **intelligent.**

Ingredients of AI

- natural language processing;
- knowledge representation;
- automated reasoning;
- machine learning;
- computer vision;
- robotics to manipulate.

The Quest for AI. Birth of Dream

1950
Turing Test to measure machine's intelligence

1950
Computer chess by C.Shannon

1956
AI founded as an academic discipline

1955
Dartmouth College Summer AI Conference
(by C. Shannon, J. McCarthy, M. Minsky, N. Rochester)

1948
'If you will tell me precisely what it is that a machine cannot do, then I can always make a machine which will do just that!'
J. Neumann

1955
"Logic Theorist"
for math proofs
(by A. Newell and H. Simon)

New paradigm
"reasoning as search"

Computers communicate in natural languages

The Optimism by AI researchers

Paradigm
"Micro-worlds"

First problems:
- limited computer power;
- intractability;
- commonsense knowledge.

The revival of connectionism

$850 million
Fifth generation computer (Japan)

'Rise of expert systems'
1980
XCON saved
$40 million annually

Nouvelle approach to AI, based on robotics

AI became a more rigorous scientific discipline

Intelligent Agents

1997
Deep Blue chess machine (IBM) beats Garry Kasparov

2005
DARPA Grand Challenge

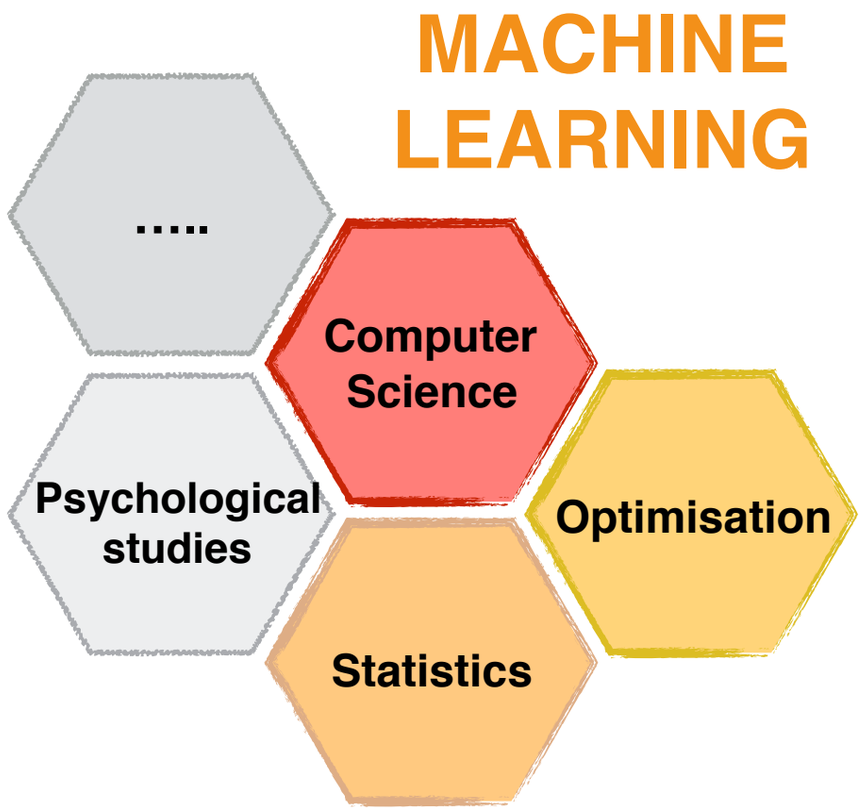
2009
Google self-driving car

2011
Apple's Siri, Google's Google Now…

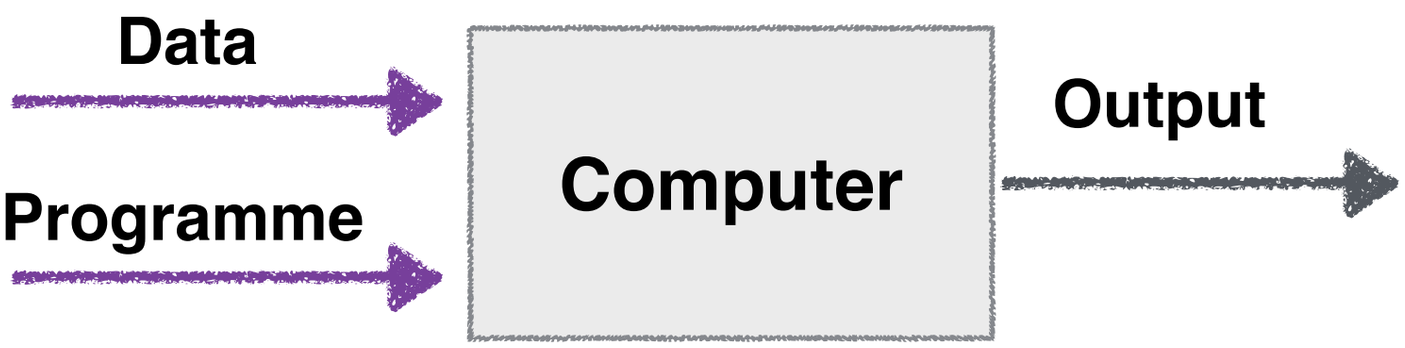
~1950 | **1952-1956** The birth of AI | **1956-1974** The golden years | **1974-1980** First AI Winter | **1980-1987** Boom | **1987-1993** Second AI Winter | **>1993**

So What is Machine Learning?

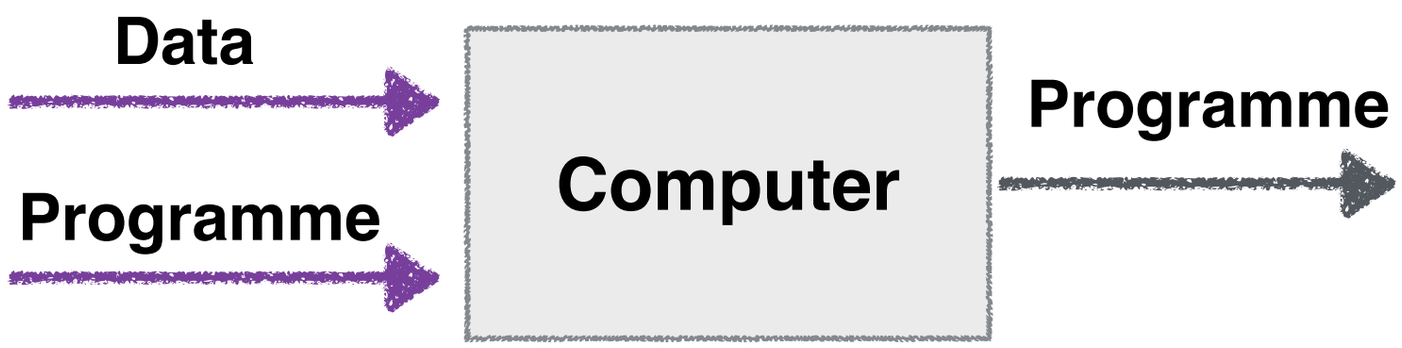
- Automating automation
- Getting computers to program themselves
- Writing software is the bottleneck
- Let the data do the work instead!



Traditional Programming



Machine Learning



Is Machine Learning Magic?

No, it is more like gardening…

- **Seeds** = Algorithms
- **Nutrients** = Data
- **Gardener** = You
- **Plants** = Programs

> Machine learning is not a magic; it can't get something from nothing. What it does it gets more from less.
>
> *Pedro Domingos, UC Washington*

Sample Applications: Prediction

Living area (feet^2)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

Sample Applications: Prediction

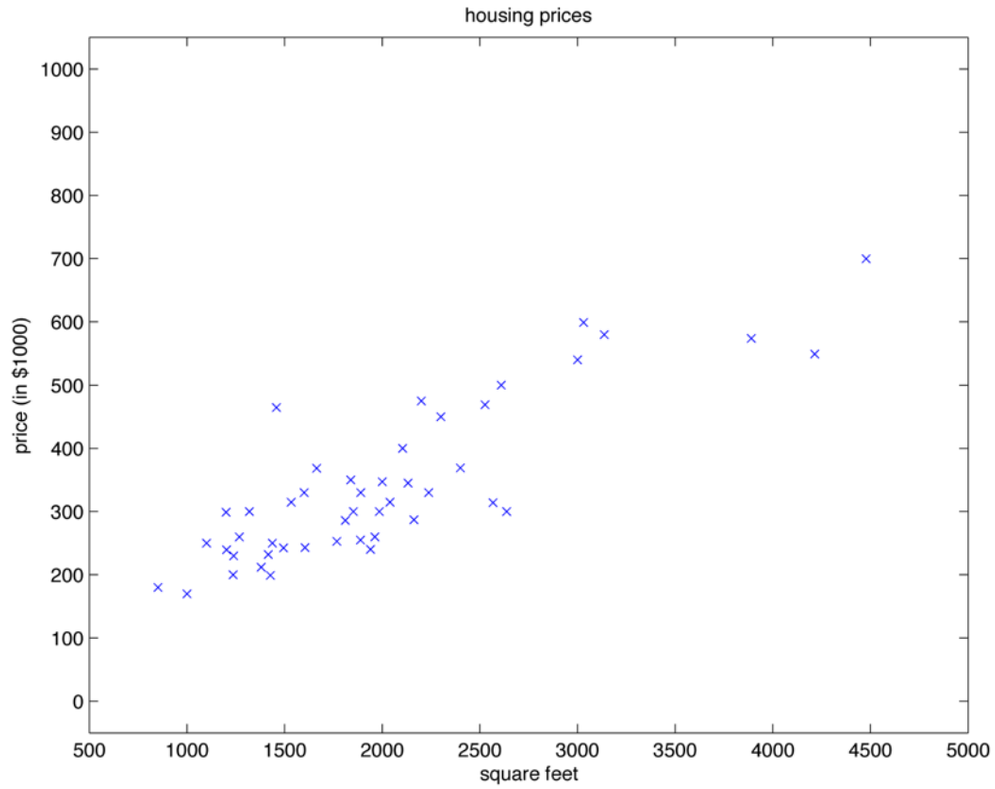
Living area (feet^2)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

$$S = \{(x_1, y_1), \ldots, (x_n, y_n)\}.$$

Sample Applications: Prediction

Living area (feet^2)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

$$S = \{(x_1, y_1), \ldots, (x_n, y_n)\}.$$

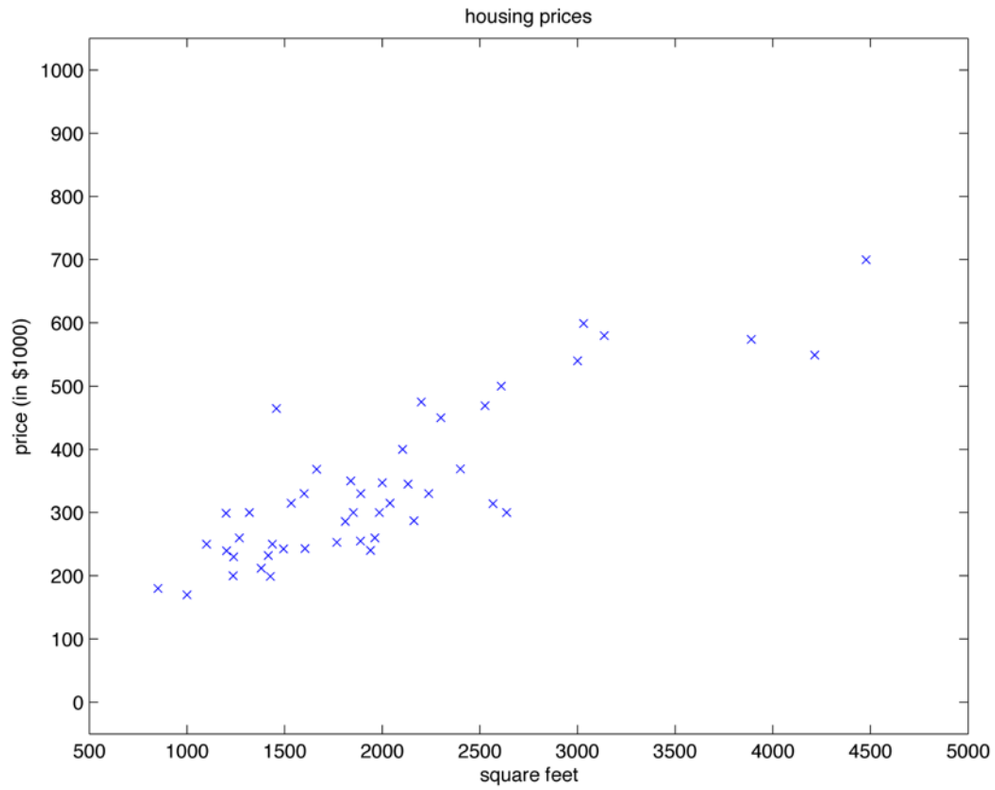


Sample Applications: Prediction

Living area (feet^2)	Price (1000$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮

$$S = \{(x_1, y_1), \ldots, (x_n, y_n)\}.$$

Living area (feet^2)	#bedrooms	Price (1000$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮



$$y_i = f(x_i) + \sigma\varepsilon_i, \;\; \sigma > 0$$

$$\text{e.g.} \quad f(x) = w^T x, \;\; \varepsilon_i \sim N(0,1)$$

Sample Applications: Spam Filtering

True loneliness is when you don't even receive spam

Ham

[simula]

Mail — More — 1–50 of 2,489

COMPOSE

Inbox (2)
Starred
Important
Sent Mail
Drafts (233)
Administration/Apa…
Administration/Boo…
Administration/Har…
Administration/Lan…
Administration/VN
Administration SIm…
Administration SIm…
Valeriya

Kristin McLeod	Hiking/cabin weekend August 20-21 - Hey gang Hermenegild, Johannes, and I were thinking of going for a weekend hiking/c	10:21
Airbnb	Reservation reminder - August 13, 2016 - Airbnb Pack your bags! It's almost time for your trip to \ Modify reservation	10:18
H2020 - IKT	**Don't miss this autumn's interesting events - H2020 ICT 10/2016** - H2020 ICT - National Contact Points - NCP Hvis du ik	**5 Aug**
Andy Edwards	Fwd: Your reservation: Aug 7 through 12 - VRBO.com #857936 - Google maps, what a tremendous piece of software. Forwa	4 Aug
Sigurd, me (2)	Report: Wednesday and Thursday (08.03 and 08.04) - Hi Sigurd, Thanks for the update and your hard work (so late!)! It look	4 Aug
Viviane, me (2)	Awesome Summer School Pool Party! - Hi Viviane, Thanks a lot for the invitation, which I gladly accept :) Looking forward to	4 Aug
Massimo, me (6)	Funny. - I fully agree :) I think I need at least one year to decide for myself, finish current work, see how it works with the grou	4 Aug
me, Kereta (6)	Projects/FRIPRO/VN Reimbursement - Dear Zeljko, Great, many thanks for the response and your wishes! I am very happy	4 Aug
Inger, me (2)	EU news item - Thanks Karoline for your kind email and changing the news! Best Valeriya On 04 Aug 2016, at 15:22, Inger H	4 Aug
Jan, me, Julio (7)	**[Reminder] CanPathPro - WP5 Skype meeting** - Dear all, this is a brief reminder regarding our regular Skype meeting. As	**4 Aug**
Lars, me (2)	Crash Course on Machine Learning - Dear Lars, Yes for sure! You are more than welcome to join. Best Valeriya Dr. Valeriya	4 Aug

Spam

[simula] in:spam

Mail — More — 1–12 of 12

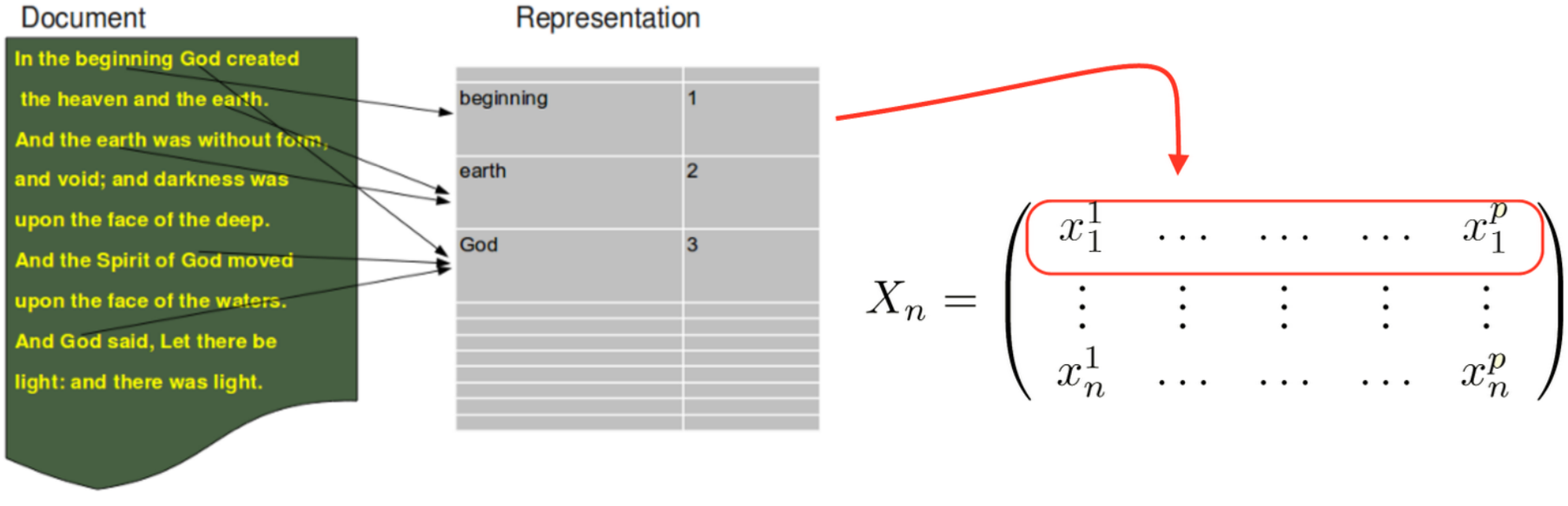
Delete all spam messages now (messages that have been in Spam more than 30 days will be automatically deleted)

COMPOSE

Inbox
Starred
Important
Sent Mail
Drafts (233)
Administration/Apa…
Administration/Boo…
Administration/Har…
Administration/Lan…
Administration/VN
Administration SIm…
Administration SIm…
Valeriya

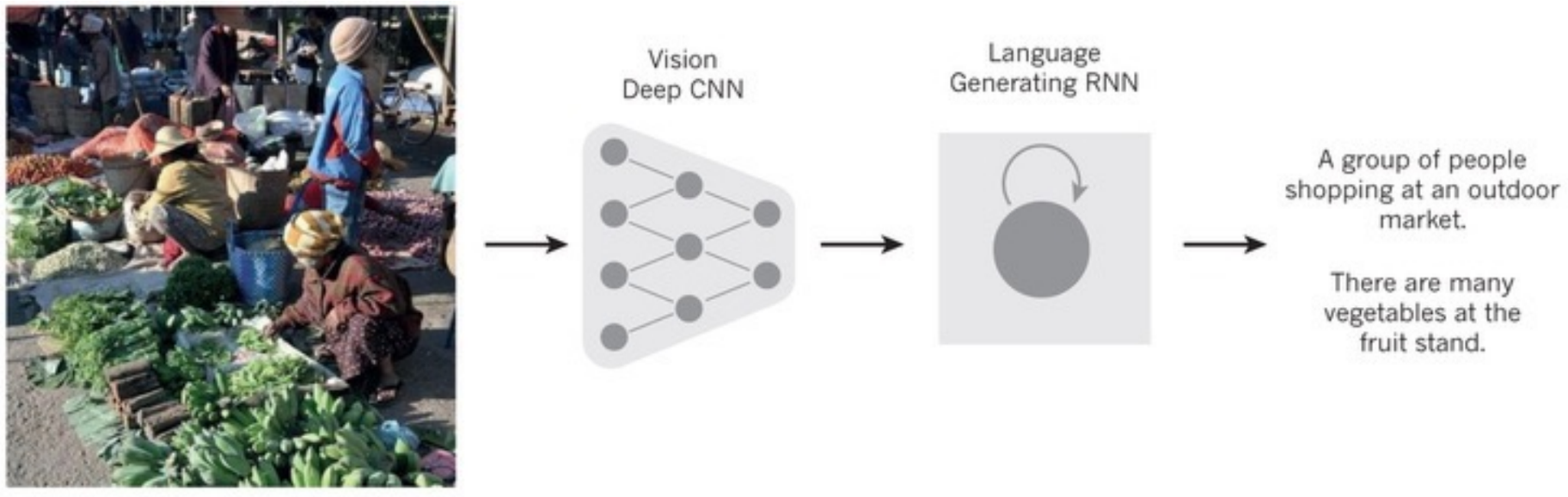
Markus Stahl	**Mehr als 448 Euro pro Tag verdienen? HEUTE noch möglich** - Sehr geehrter Interessent, wir möchten Sie mit dieser Em	**03:39**
Laura Schubert (4)	**RE: Kundendienst schuldet dir Geld** - Hallo valeriya@simula.no! Das Customer Service Team sendete mir deine Details ü	**5 Aug**
Dr. Matthias Reich	**Der unerwartete Geldsegen** - Guten Tag, Unverhofft kommt oft? Naja, in diesem Fall stimmt es zumindest. Klicken Sie hier	**5 Aug**
Google Trading Inc. (5)	**TAG 1 - Wettbewerb einladen for valeriya@simula.no!** - Hallo valeriya@simula.no, TAG 1 competition.jpeg Ich habe eine	**5 Aug**
Linnea	**Symposium 2016 - Invitation Letter from China** - Symposium 2016 - Invitation Letter from China Dear Professor/Researcl	**5 Aug**
Mr.Brian William.	**Attention.** - BG Group PLC Thames Valley Park, Reading, RG6 1PT, Berkshire, United Kingdom. Attention. I am Mr. Brian V	**5 Aug**
Sven Lindholm	**Recently posted academic job vacancies at Educaloxy** - Dear Colleague, We are pleased to present you our specialised	**5 Aug**
Google Benachrichtigung	**Wettbewerb einladen for valeriya@simula.no!** - IEMB16051 Hallo valeriya@simula.no, Ich habe eine Woche gewartet oh	**4 Aug**
DNB Bank ASA	**Varsling!** - Kjære kunde, Du har mottatt en intern melding. Klikk her Vennlig hilsen DNB ASA	4 Aug
giuseppelunardi	RE: valeriya	2 Aug

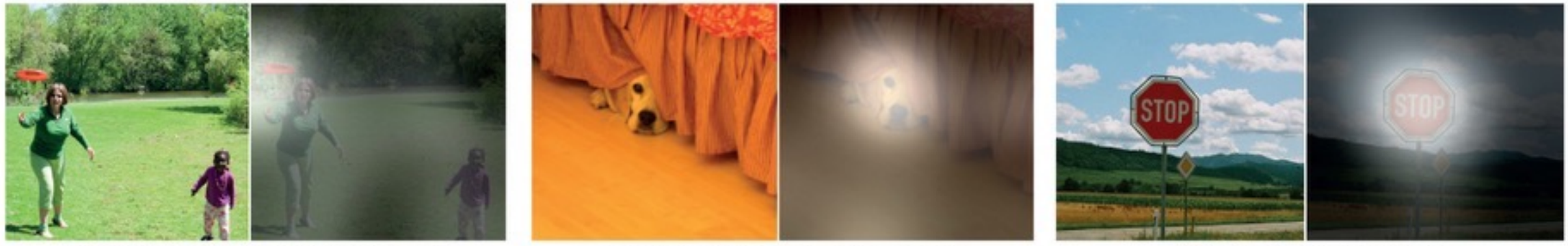
Sample Applications: Text Classification



Sample Applications: Object Recognition

Sample Applications: from Image to Text





A woman is throwing a **frisbee** in a park.

A **dog** is standing on a hardwood floor.

A **stop** sign is on a road with a mountain in the background

A little **girl** sitting on a bed with a teddy bear.

A group of **people** sitting on a boat in the water.

A giraffe standing in a forest with **trees** in the background.

Nature vol 521, 2015

Sample Applications

Web ranking

Social Networks

Computational Biology

Debugging

Finance

E-Commerce

Mobile technology

Robotics

Space exploration

Navigation

…[Your favourite area]

Tens of thousands of machine learning algorithms

Hundreds new every year

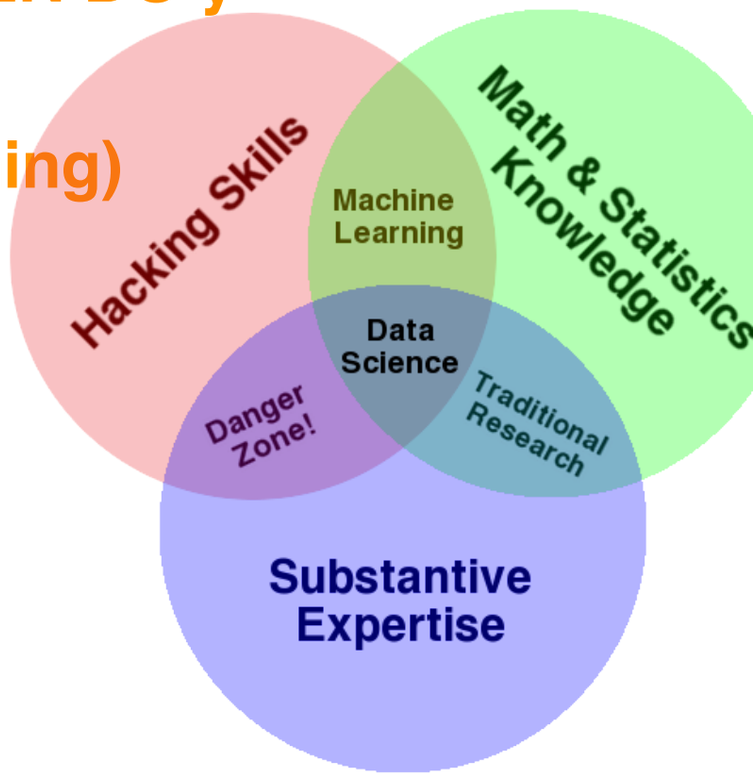
Programming with Data / Data Analysis

- Want adaptive robust and fault tolerant systems
- Rule-based implementation is (often)
  - difficult (for the programmer)
  - brittle (can miss many edge-cases)
  - becomes a nightmare to maintain explicitly
  - often doesn't work too well (e.g. OCR)

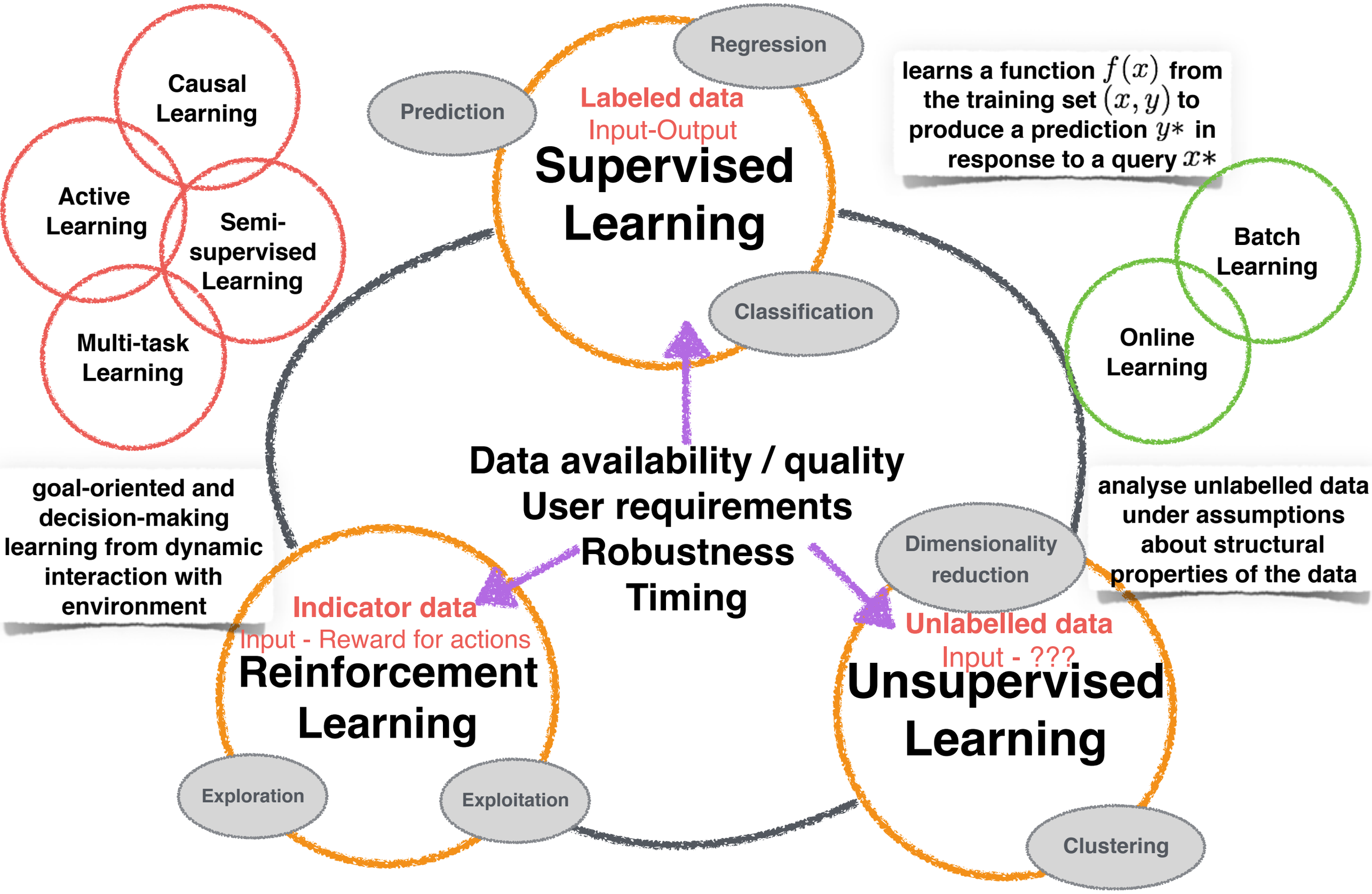
> We say that a program for performing a task has been acquired by learning if it has been acquired by any means ofter than explicit programming.
>
> *Valiant, 1984*

- Usually easy to obtain examples of what we want **IF x THEN DO y**
- Collect many pairs $\{(x_i, y_i)\}_{i=1}^{N}$
- Estimate function such that $f(x_i) = y_i$ **(supervised learning)**
- Detect patterns in data **(unsupervised learning)**

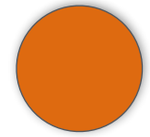
> . . . learning from examples, refers to systems that are trained instead of programmed with a set of examples, that is, a set of input/output pairs…
>
> *Poggio and Smale, 2003*



Three paradigms of learning

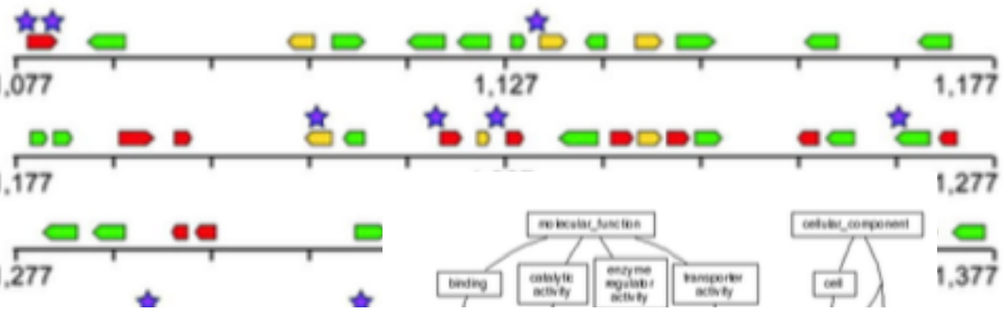


Problem Prototypes: Supervised Learning $y = f(x)$



We are given a set of input-output pairs (training set) $\{(x_i, y_i)\}_{i=1}^{n}$

- **Binary classification:** Given x find y in {-1,1}
- **Multi-category (Multi-class) classification:** Given x find y in {1,. . . ,K}
- **Regression:** Given x find y in $\mathbb{R}$ (or $\mathbb{R}^d$)
- **Sequence annotation:** Given sequence $x_1, \ldots, x_k$ find $y_1, \ldots, y_l$
- **Hierarchical categorisation (Ontology):**



Given x find a point in

- **Prediction:** Given x_t and $y_{t-1}, \ldots, y_1$ find

Forms of mapping f:

- decision trees / forests,
- logistic regression,
- support vector machine,
- neural networks,
- kernel machines,
- Bayesian classifiers.

given sequence

gene finding

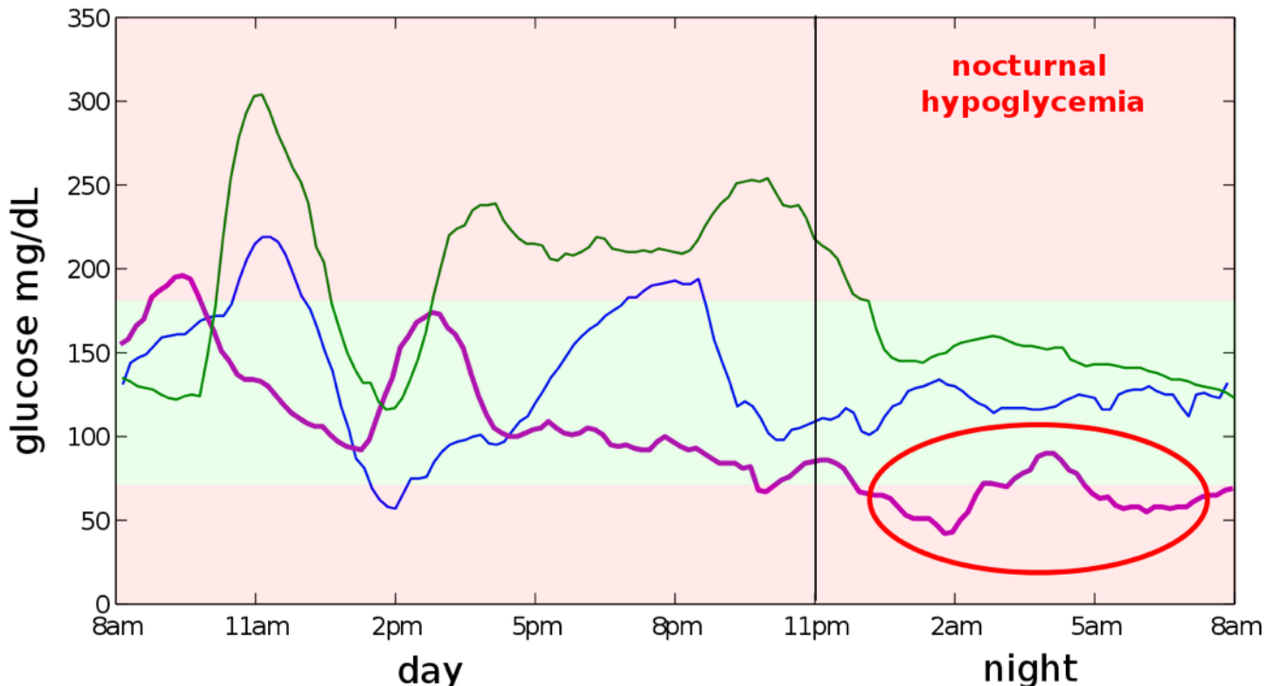
Map image to digit

speech recognition

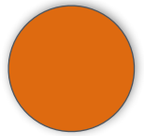
activity segmentation

Often with loss function

$$V(y, f(x))$$



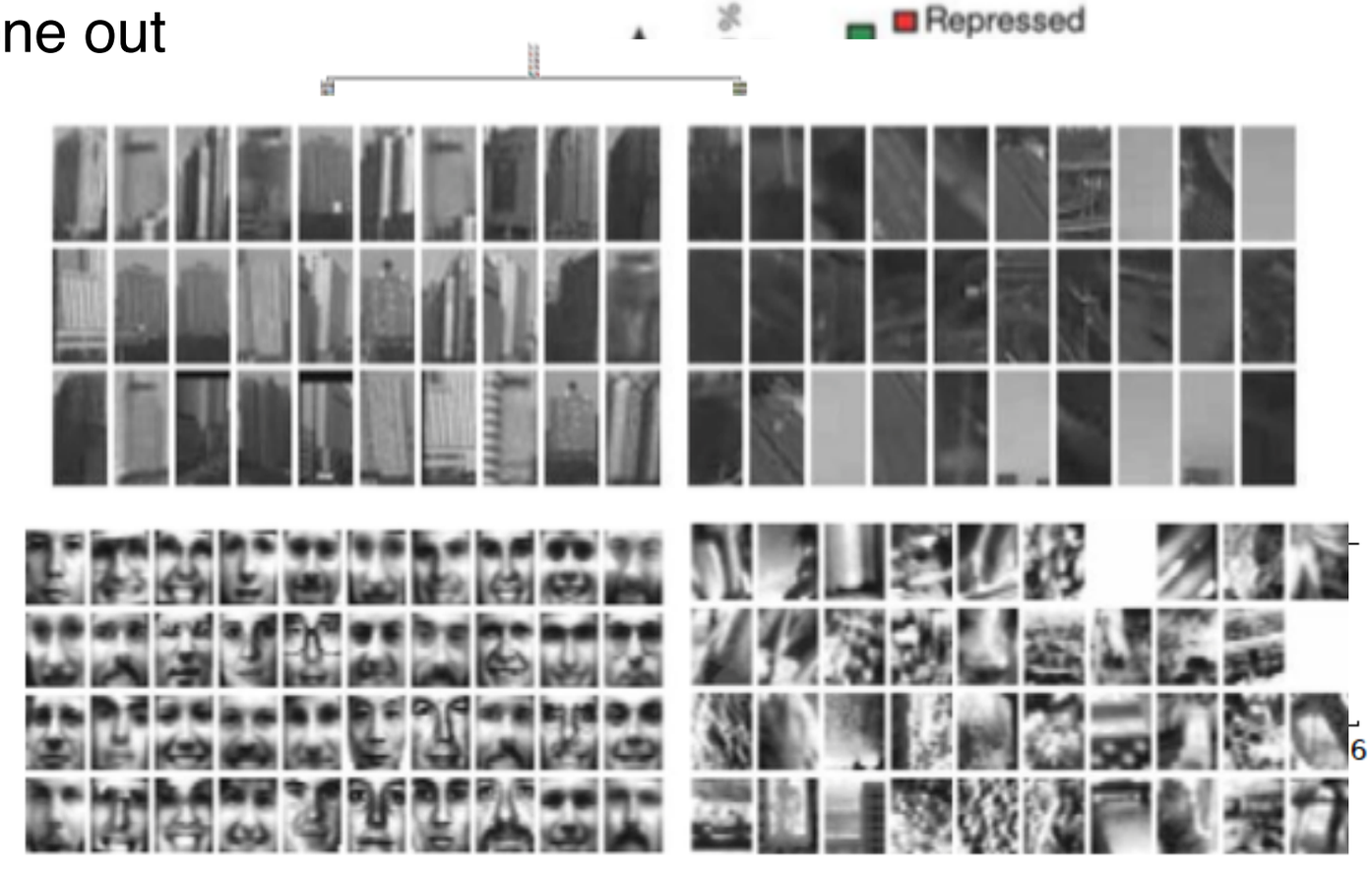
Problem Prototypes: Unsupervised Learning



Given data x , ask a good question…. about x or about model for x

- **Clustering:** Find a set of prototypes representing the data
- **Principal components:** Find a subspace representing the data
- **Sequence analysis:** Find a latent causal sequence for observations

  (Kalman Filter, Hidden Markov Model)
- **Hierarchical representation**
- **Dictionary learning:** Find (small) set of factors for observation
- **Novelty detection:** Find the odd one out



Problem Prototypes: Reinforcement Learning

Examples of desired behaviour are not available but it is possible to score examples of behaviour according to some performance measure

Sequential decision tasks:

- Take action
- Environment responds
- Observe stuff
- Update model
- Repeat

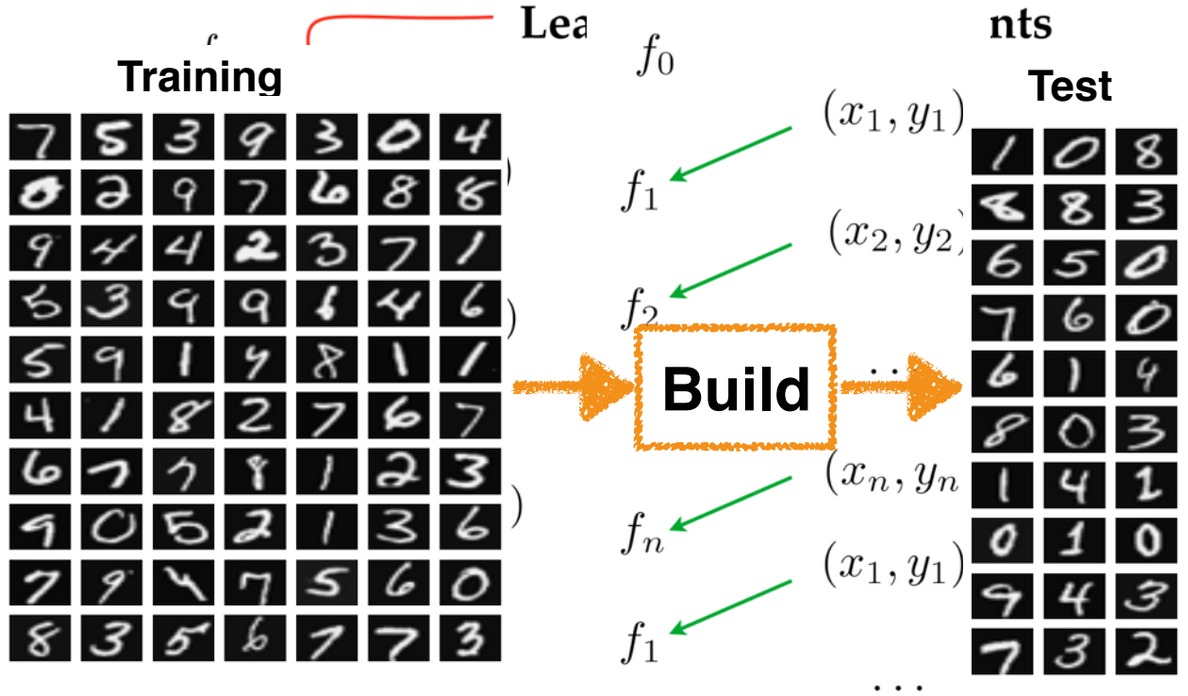
Bandits: Pick arm, get reward, pick new arm

- Choose an option
- See what happens (get reward)
- Update model
- Choose next option

Reinforcement learning combines search and long-term memory

Differences to Supervised learning?

Problem Prototypes: Other types



- **Induction:** Reasoning from observed training cases to general rules, which are then applied to the test cases
- **Transductive inference:** Reasoning from observed, specific (training) cases to specific (test) cases

- **Multi-task learning:** Learns a problem together with other related problems at the same time, using a shared representation.
- **Active learning:** Interactive querying of the user to obtain the desired outputs at anew data points
- **Online learning:** Analyse each training example as it is presented
- **Batch learning:** Collect training examples, analyse them, output an hypothesis

Key Issues in Machine Learning

- **What are good hypothesis spaces?** Which spaces have been useful in practical applications and why?
- **What algorithms can work with these spaces?** Are there general design principles for machine learning algorithms?
- **How can we optimise accuracy on future data points?** This is sometimes referred to “generalisation” ability or problem of overfitting.
- **How can we have confidence in the results?** *(the statistical question)*
- **Are some learning problems computationally intractable?**

  (the computational question)
- **How can we formulate application problems as machine learning problems?** *(the engineering question)*

Hypothesis: a function produced by a learning algorithm, which is believed to be close to the true function.

Hypotheses space: the space of all hypotheses that can be an output by a learning algorithm.

Part 0. Settings for Supervised Learning

The goal of **supervised learning** is to find an underlying input-output relation

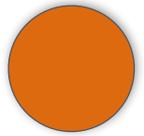
$$f(x_{new}) \sim y$$

given data.

The data, called **training set**, is a set of n input-output pairs,

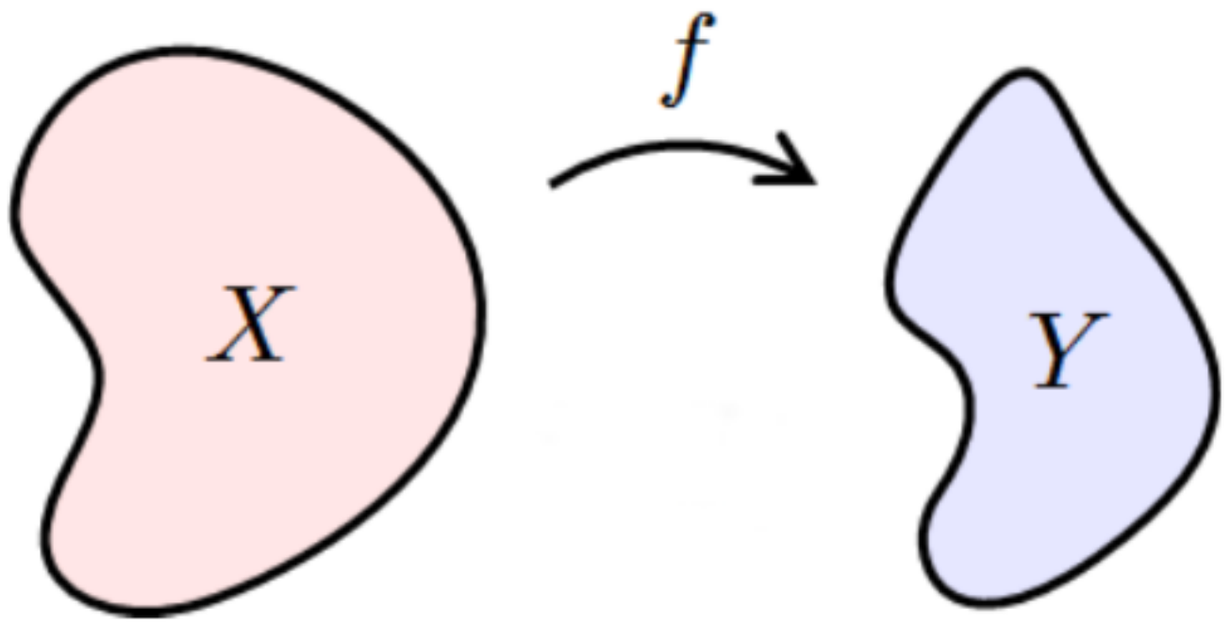
$$S = \{(x_1, y_1), \ldots, (x_n, y_n)\}.$$

We Need a Model to Learn



- We consider the approach to machine learning based on the *learning from examples* paradigm.
- **Goal:** Given the training set, learn a corresponding I/O relation.
- We have to postulate the existence of a model for the data.
- The model should take into account the possible *uncertainty* in the task *and* in the data.

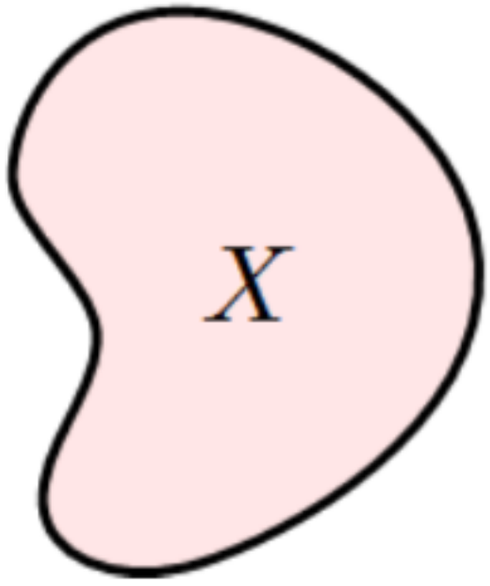
Data Space



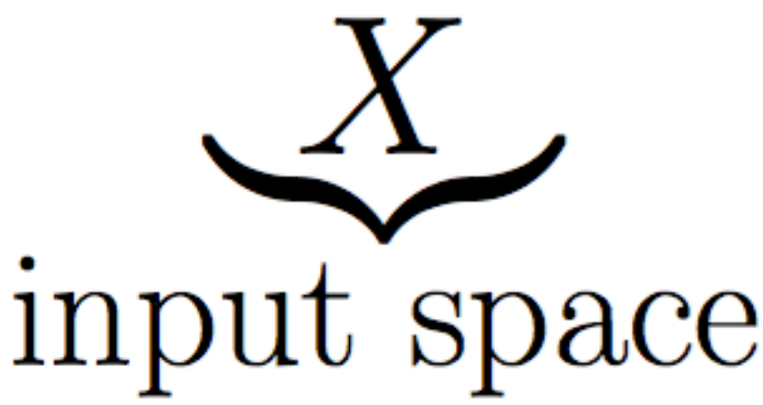
$$\underbrace{X}_{\text{input space}} \quad \mathbf{x} \quad \underbrace{Y}_{\text{output space}}$$

Data Space

Examples of Input Space

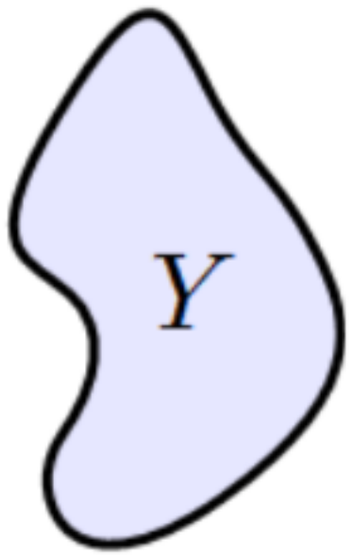


- linear spaces, e. g.
  - vectors,
  - functions,
  - matrices/operators

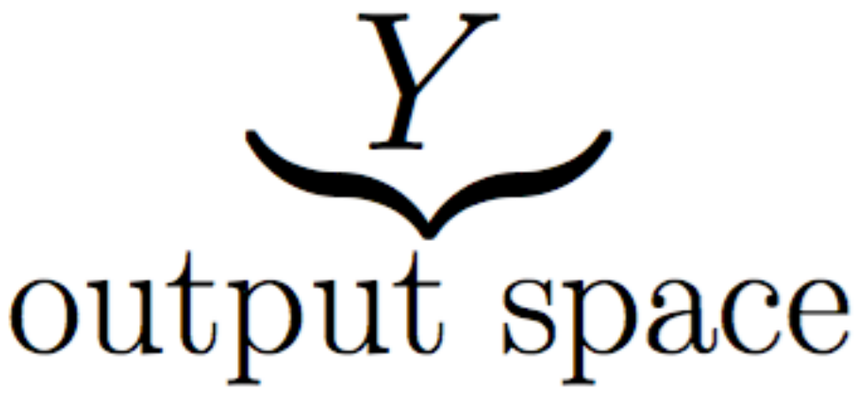


- “structured” spaces, e. g.
  - strings,
  - probability distributions,
  - graphs

Examples of Output Space

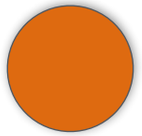


- linear spaces, e. g.
  - $Y = \mathbb{R}$, regression,
  - $Y = \{+1, -1\}$, classification
  - $Y = \mathbb{R}^T$, multi-task regression
  - $Y = \{1, \ldots, T\}$, multi-label classification



- “structured” spaces, e. g.
  - strings,
  - probability distributions,
  - graphs

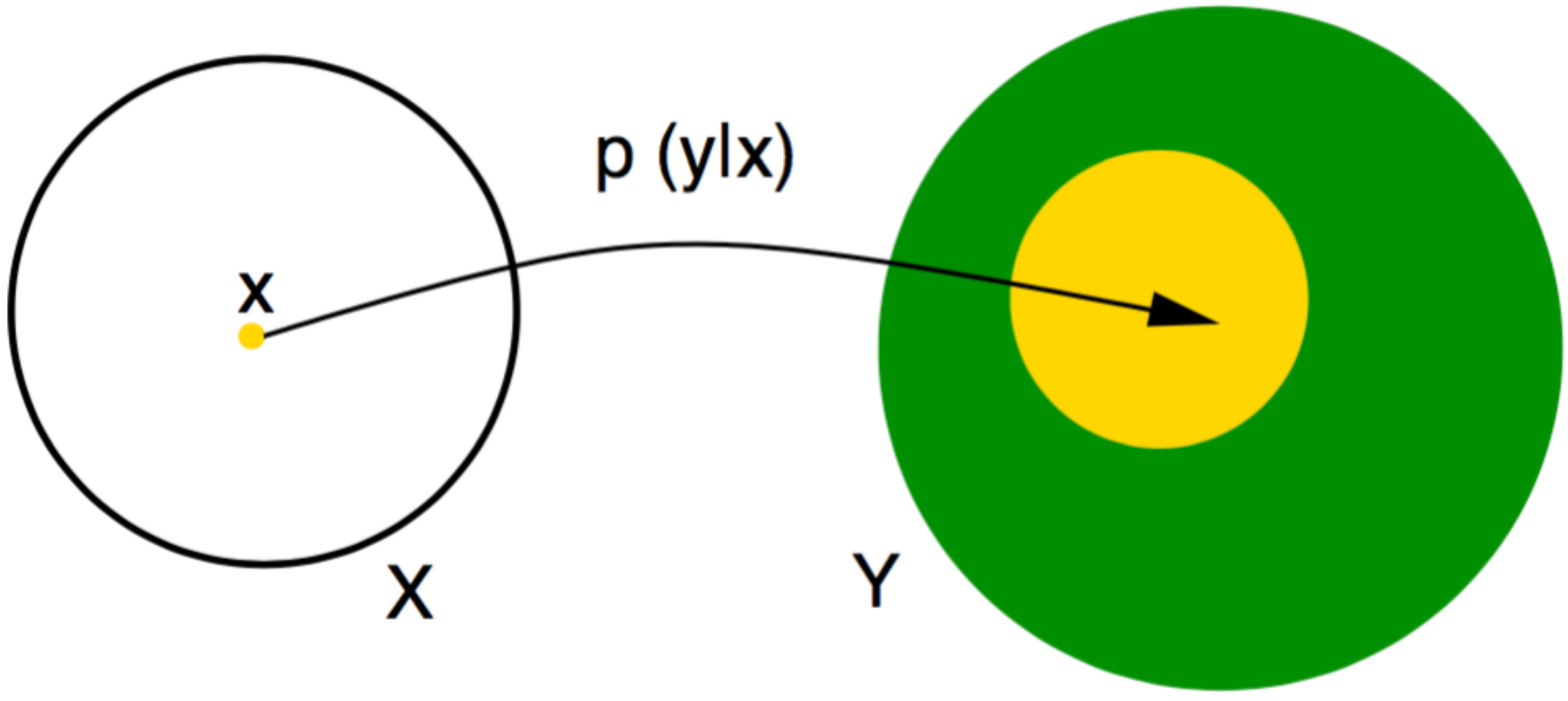
Modelling Uncertainty in the Data Space



- **Assumption**: there exists a fixed unknown distribution $\rho(x, y)$, according to which the data are identically and independently sampled
- The distribution ρ models different sources of uncertainty
- **Assumption**: ρ factorizes as $\rho(x, y) = \rho_X(x)\rho(y|x)$

Marginal and Conditional

$\rho(y|x)$ can be seen as a form of noise in the output



For each input x there is a distribution of possible outputs $\rho(y|x)$

The marginal distribution $\rho_X(x)$ models uncertainty in the sampling of the input samples

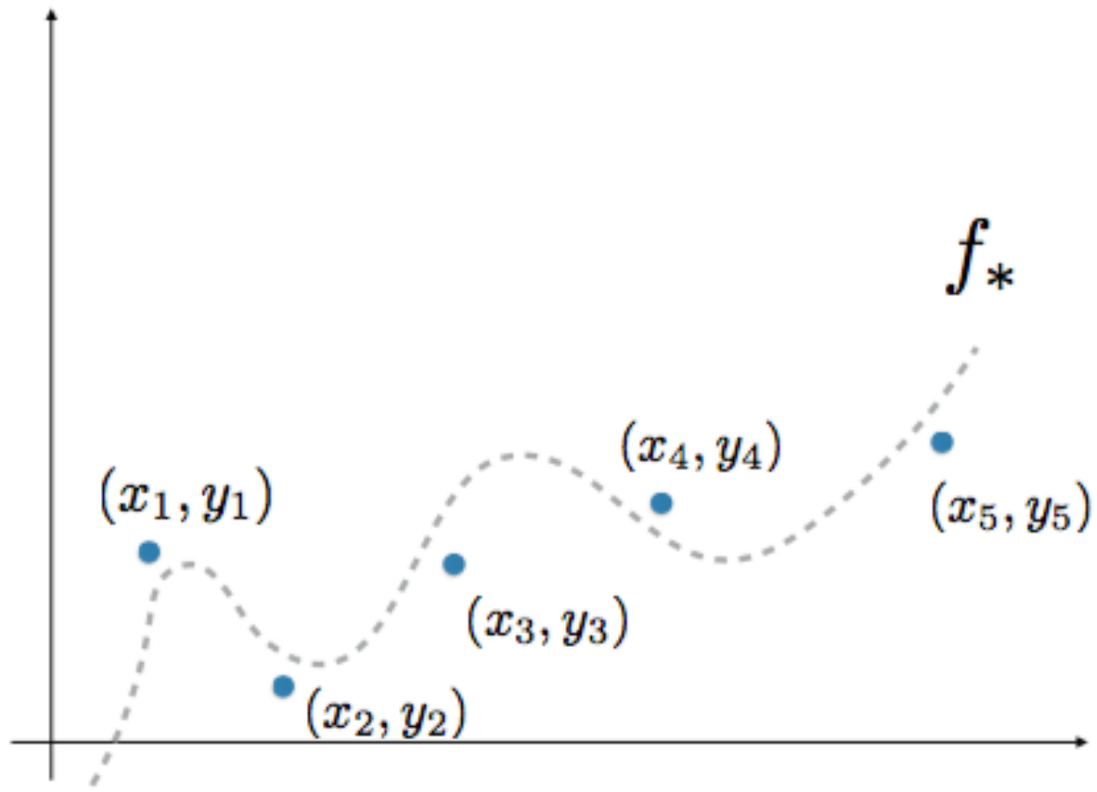
Data Models

In **regression**, the following model is often considered

$$y = f^*(x) + \epsilon$$

where

- $f^*(x)$ fixed unknown (regression) function
- ϵ random noise, e.g., standard Gaussian



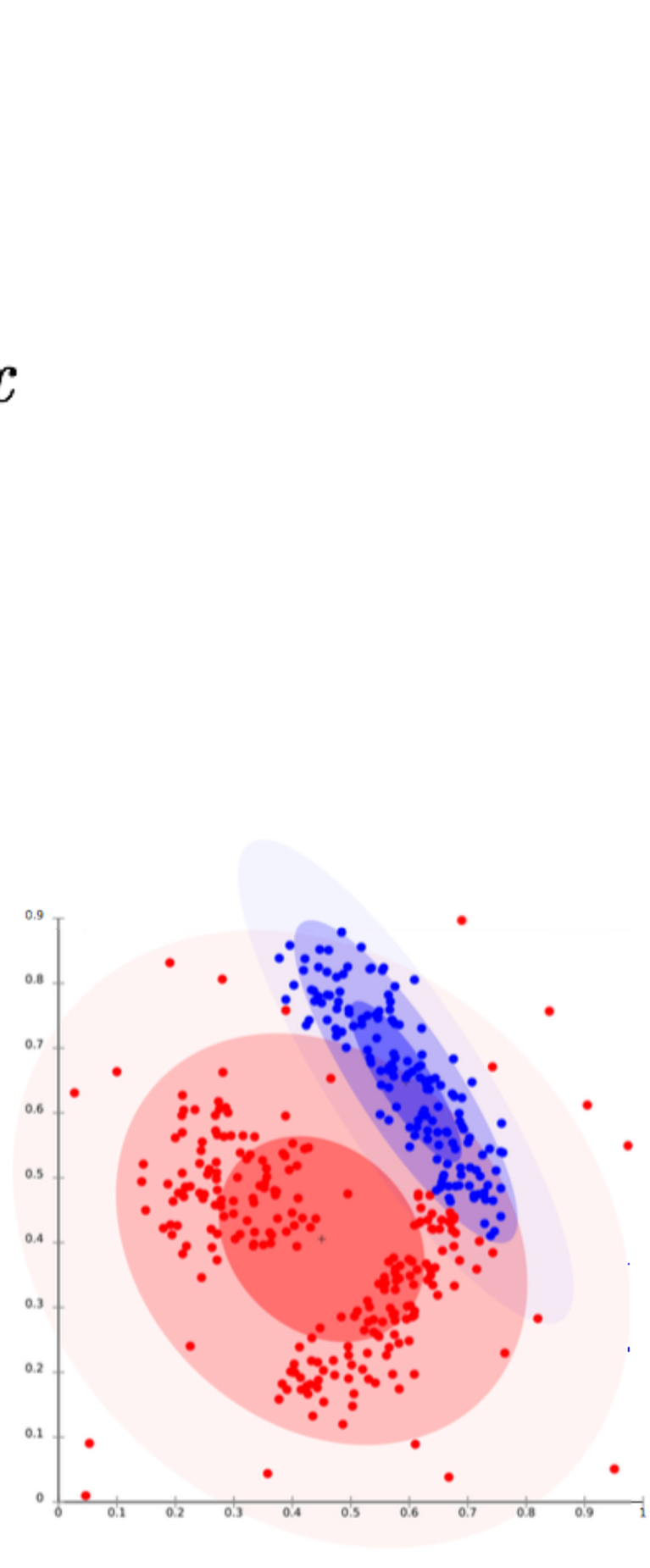
Data Models

In **classification**,

$$\rho(1|x) = 1 - \rho(-1|x) \quad \forall x$$

Noiseless classification

$$\rho(1|x) = \{1, 0\} \quad \forall x$$



Loss Function

Goal of learning: Estimate "best" I/O relation, not the whole $\rho(x, y)$

We need to fix a measurable loss function

$$V : Y \times Y \to [0, \infty)$$

It is a point-wise error measure: cost of when predicting $f(x)$ in place of *y*

Expected Risk and Target Function

The *expected loss* (or *expected risk*)

$$\mathcal{E}(f) = \mathbb{E}[V(y, f(x))] = \int \rho(x, y) V(y, f(x)) dx dy$$

can be seen as a measure of the error on past as well as future data.

Given the loss function and a distribution, the 'best' I/O relation is the *target function*

$$f^* : X \to Y$$

that minimises the expected risk.

Learning from Data. Learning Algorithms

- The target function cannot be computed, since the probability is unknown.
- The goal of learning is to find an estimator of the target function from data.

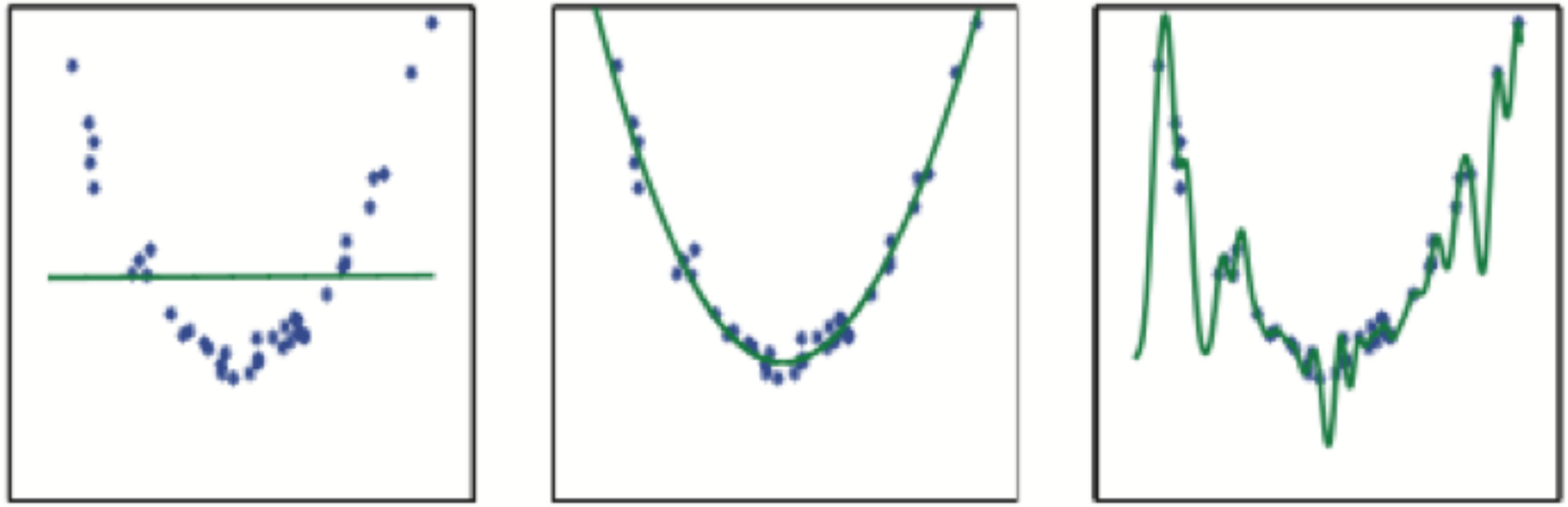
- A learning algorithm is a procedure that given a training set S computes an estimator f_S .
- An estimator should mimic the target function, in which case we say that it **generalises.**

Generalisation: Fitting and Stability

How to design a good algorithm?

Two concepts are key:

- **Fitting**: an estimator should *fit* data well.
- **Stability**: an estimator should be stable, it should not change much if data change slightly.



- We say that an algorithms **overfits**, if it fits the data while being unstable.
- We say that an algorithms **oversmooth**, if it is stable while disregarding the data.

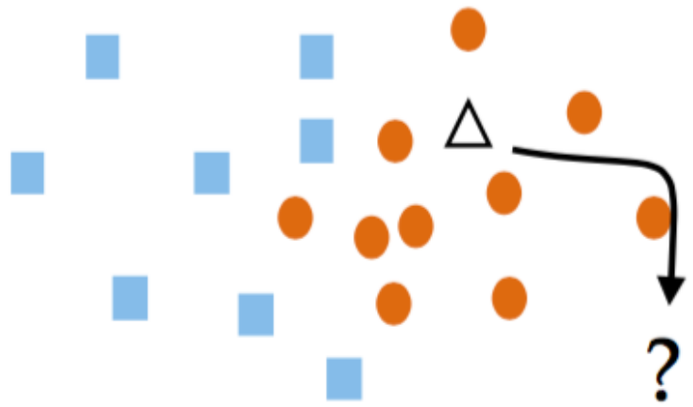
Part I. Local Methods / Bias-Variance

GOAL:

Discuss the fundamental concept of bias-variance trade-off starting from simple machine learning approaches to understand parameter tuning

Local Methods: Nearest Neighbour

Nearby points have similar labels



Given a training set

$$S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \text{ with } x_i \in \mathbb{R}^D, y_i \in \mathbb{R}.$$

and a new input $\bar{x}$, let

$$i' = \arg \min_{i=1,\ldots,n} \|\bar{x} - x_i\|^2$$

In general,

$$i' = \arg \min_{i=1,\ldots,n} d(\bar{x}, x_i)$$

and define the nearest neighbour (NN) estimator as

$$\hat{f}(\bar{x}) = y_{i'}$$

Can we do any better?

Computational cost: $O(nD)$

We compute n times the distance $\|\bar{x} - x_i\|$ which costs $O(D)$

Local Methods: K-Nearest Neighbours

A new point is the mean values of the K closest points

Consider

$$d_{\bar{x}} = (\|\bar{x} - x_i\|^2)_{i=1}^n$$

the array of distances of a new point $\bar{x}$ to the input points in the training set. Let

$$s_{\bar{x}}$$

be the above array sorted in increasing order and

$$I_{\bar{x}}$$

the corresponding vector of indices,

$$K_{\bar{x}} = \{I_{\bar{x}}^1, \ldots, I_{\bar{x}}^K\}$$

be the array of the first K entries of $I_{\bar{x}}$. The K-nearest neighbours estimator (KNN) is defined as,

$$\hat{f}(\bar{x}) = \sum_{i' \in K_{\bar{x}}} y_{i'}$$

Computational cost: $O(nD + n \log n)$

We compute the n distance for each point and order them

Local Methods. Remarks

Generalisation I: closer points should count more (*Parzen windows*)

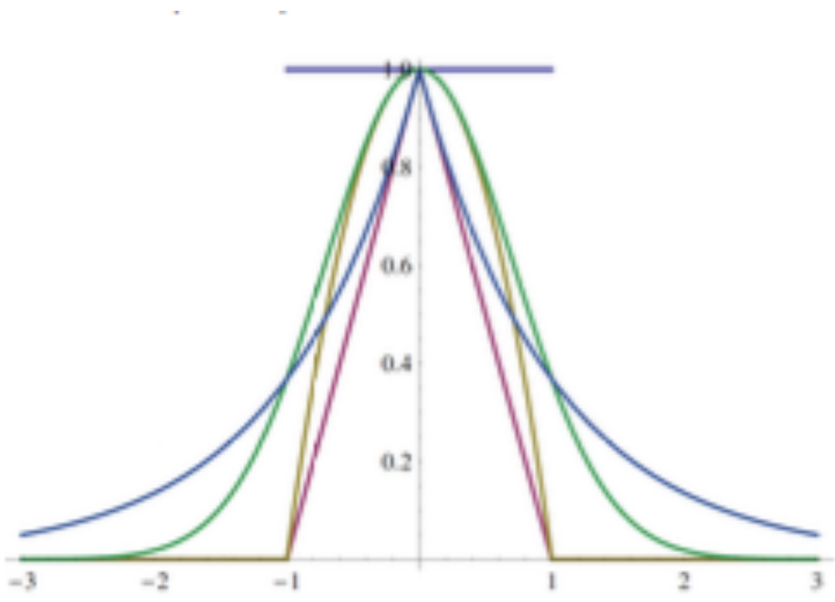
$$\hat{f}(\bar{x}) = \frac{\sum_{i=1}^{n} y_i k(\bar{x}, x_i)}{\sum_{i=1}^{n} k(\bar{x}, x_i)}$$

where k is a similarity function

$k(x, \bar{x}) \geq 0$ for all $x, \bar{x} \in \mathbb{R}^D$;

$k(x, \bar{x}) \to 1$ when $\|x - \bar{x}\| \to 0$;

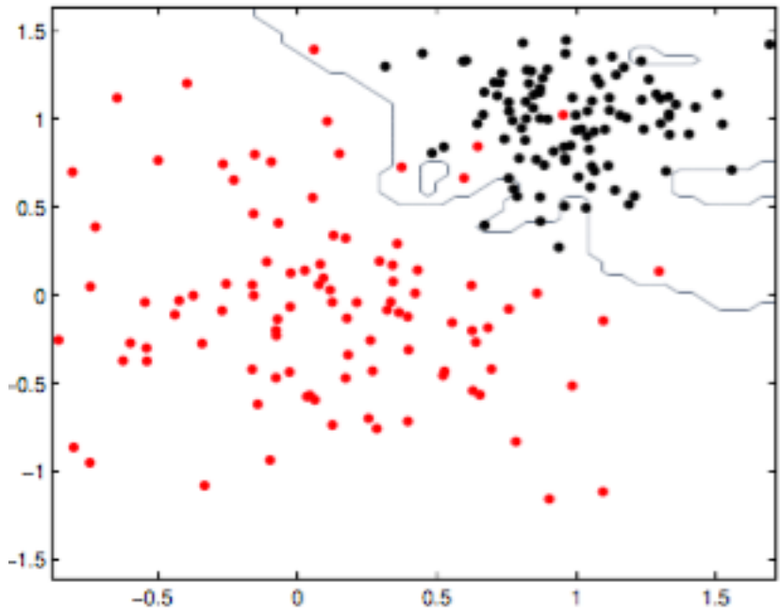
$k(x, \bar{x}) \to 0$ when $\|x - \bar{x}\| \to \infty$.



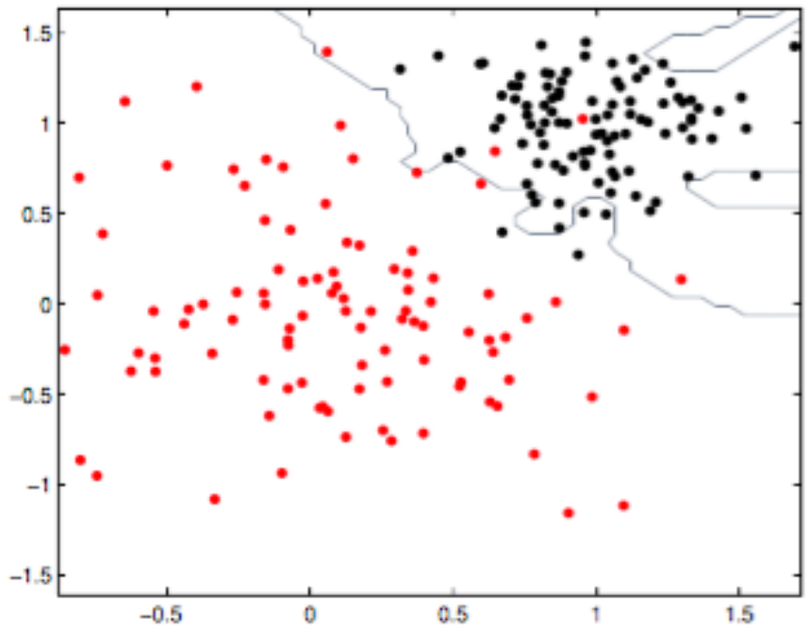
Generalisation II: other metric / similarities

$$X = \{0, 1\}^D \qquad d_H(x, \bar{x}) = \frac{1}{D} \sum_{j=1}^{D} \mathbf{1}_{[x^j \neq \bar{x}^j]}$$

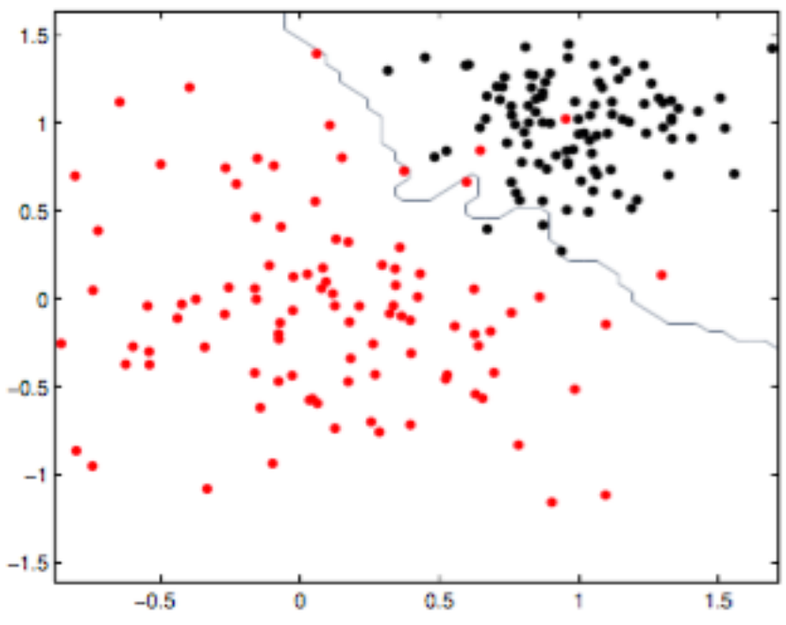
Local Methods



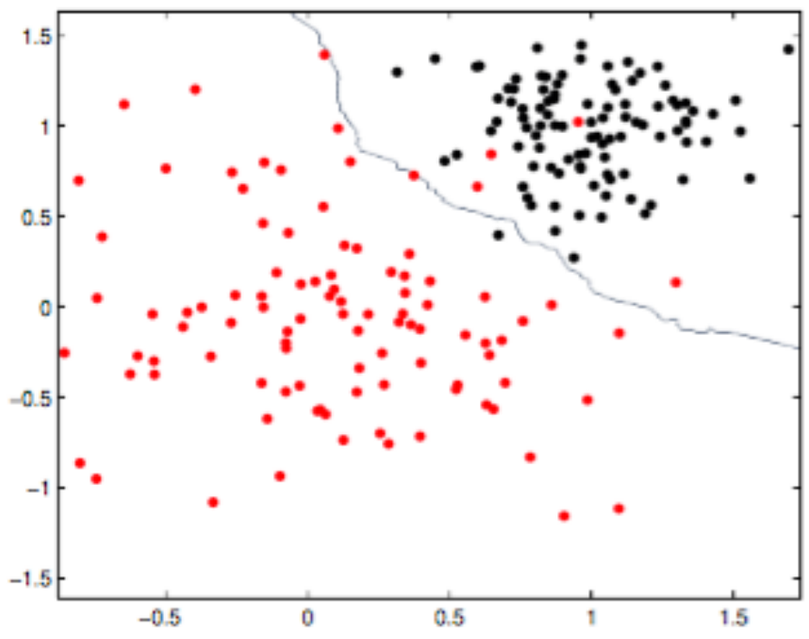
1-Nearest Neighbour



2-Nearest Neighbours



5-Nearest Neighbours



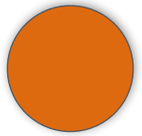
9-Nearest Neighbours

How can we choose it?

Is there an optimal value?

Can we compute it?

Model Selection Bias-Variance Trade-Off



Is there an optimal parameter?

Ideally we would like to choose K that minimises the expected error

$$\mathcal{E}_K = \mathbf{E}_S \mathbf{E}_{x,y}(y - \hat{f}_K(x))^2.$$

Optimal hyperparameter K* should minimise $\mathcal{E}_K$

$$K^* = \arg\min_{K \in \mathbb{K}} \mathcal{E}_K$$

Ideally! (In practice we don't have access to the distribution)

- We can still try to understand the above minimisation problem: **Does a solution exists? What does it depend on?**
- Yet, ultimately, we need something we can compute!

Next: Characterise corresponding minimisation problem to uncover one of the **most fundamental aspect of machine learning.**

Example: Regression

Regression model $y_i = f^*(x_i) + \delta_i$

$\mathbf{E}\delta = 0, \mathbf{E}\delta_i^2 = \sigma^2, i = 1, \ldots, n$

Fix x

Now $\mathcal{E}_K = \mathbf{E}_x \mathbf{E}_S \mathbf{E}_{y|x}(y - \hat{f}_K(x))^2 = \mathbf{E}_S \mathbf{E}_{y|x}(f^*(x) + \delta - \hat{f}_K(x))^2$
that is

$$\mathcal{E}_K(x) = \mathbf{E}_S(f^*(x) - \hat{f}_K(x))^2 + \sigma^2$$

...

Bias-Variance Trade-Off for K-NN

Define the noiseless K-NN (it is ideal!)

$$\mathbf{E}_{y|x}\hat{f}_K(x) = \frac{1}{K}\sum_{l \in K_x} f^*(x_l).$$

Bias-Variance Trade-Off for K-NN

Define the noiseless K-NN (it is ideal!)

$$\mathbf{E}_{y|x}\hat{f}_K(x) = \frac{1}{K}\sum_{l\in K_x} f^*(x_l).$$

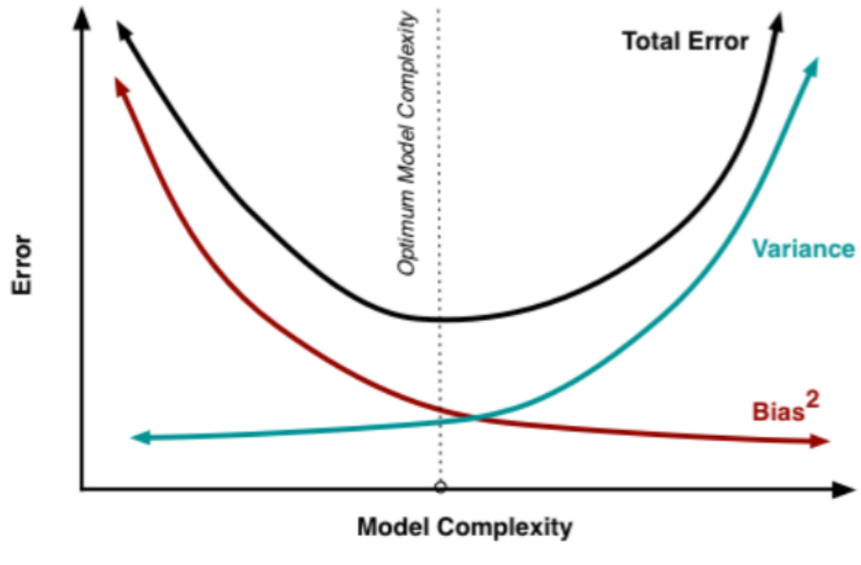
Consider

$$\mathcal{E}_K(x) = \mathbf{E}_S\mathbf{E}_{y|x}(f^*(x) - \hat{f}_K(x))^2 =$$
$$= \underbrace{(f^*(x) - \mathbf{E}_S\mathbf{E}_{y|x}\hat{f}_K(x))^2}_{bias} + \underbrace{\mathbf{E}_S\mathbf{E}_{y|x}(\mathbf{E}_{y|x}\hat{f}_K(x) - \hat{f}_K(x))^2}_{variance} + \sigma^2$$

$$(f^* - \frac{1}{K}\sum_{l\in K_x} f^*(x_l))^2 \qquad + \qquad \frac{\sigma^2}{K}$$

Is there an optimal value?

Yes



Can we compute it?

Model Selection Bias-Variance Trade-Off

Bias-Variance Trade-Off is theoretical but shows that:

- an optimal parameter exists and
- it depends on the noise and the unknown target function.

How to choose K in practice? **Cross Validation!!!**

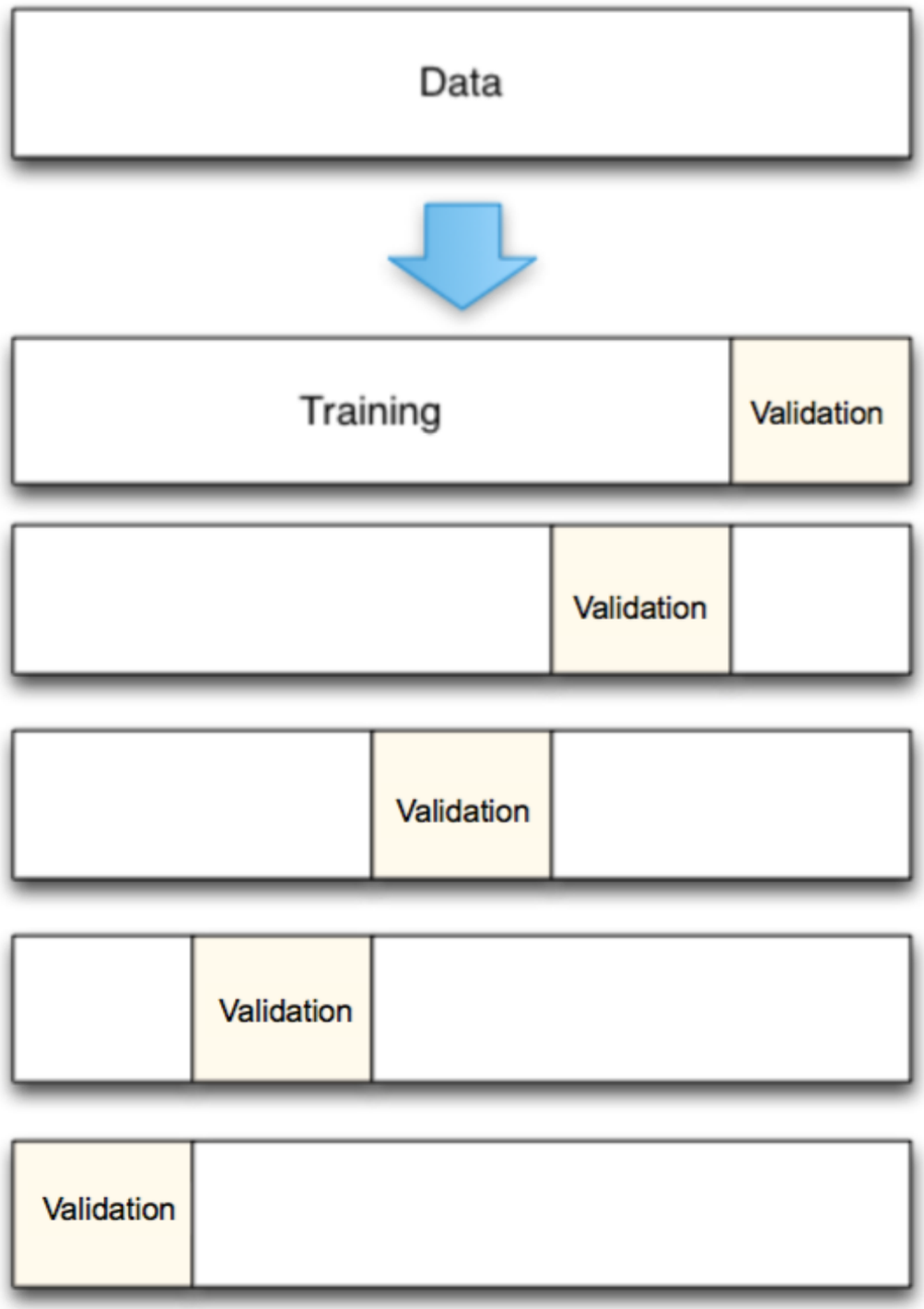
- **Idea:** train on some data and validate the parameter on new unseen data as proxy for the ideal case.

Cross-Validation Flavours

Hold-out

Cross-Validation Flavours

K-Fold



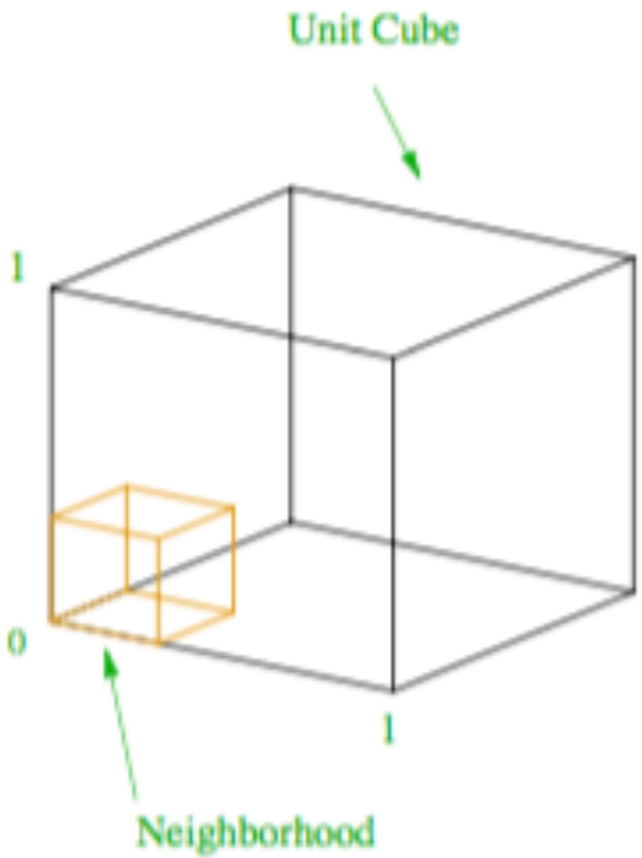
End of Part I

We made our first encounter with learning algorithms (local methods) and the problem of tuning their parameters (via bias-variance trade-off and cross-validation) to avoid overfitting and achieve generalisation.

Stability - Overfitting - Bias/Variance - Cross-Validation

End of story?

Local Methods in High Dimensions



What is the length of the edge of a cube containing 1% of the volume of a cube with edge 1?

GOAL:

Introduce the basic (global) regularisation methods and study their computational aspects

Going Global + Impose Smoothness

Regularisation. Expected Risk Minimisation

Of all the principles which can be proposed for that purpose, I think there is none more general, more exact, and more easy of application, that of which we made use in the preceding researches and which consists of rendering the **sum of square of the errors** a minimum.

Legendre, 1805

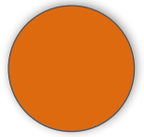
- **Empirical Risk Minimisation (ERM):** probably the most popular approach to design learning algorithms.
- General idea: considering the empirical error

$$\hat{\mathcal{E}}(f) = \frac{1}{n}\sum_{i=1}^{n} V(y_i, f(x_i))$$

as a proxy for the expected risk

$$\mathcal{E}(f) = \mathbf{E}[V(y, f(x))] = \int p(x, y) V(y, f(x)) dx dy$$

Regularisation. Expected Risk Minimisation



- V measures the price we pay predicting $f(x)$ when the true label is y
- $\mathcal{E}(f)$ cannot be directly computed, since $p(x, y)$ is unknown

To turn the above idea into an actual algorithm, we need to

- Fix a suitable hypothesis space H
- Minimise $\hat{\mathcal{E}}(f)$ over H

The space should allow *feasible computations* and be *rich*, since the complexity of the problem is not known a priori.

Simplest example: Space of linear functions

$$H = \{f : \mathbb{R}^D \to \mathbb{R} : \exists\, \omega \in \mathbb{R} \text{ such that } f(x) = x^T\omega,\ \forall x \in \mathbb{R}^D\}$$

- Each function f is defined by the vector ω : $f(x) = x^T\omega$

If *H* is rich enough, solving ERM may cause overfitting (solutions highly dependent on the data)

Regularisation. Tikhonov regularisation

Regularisation techniques restore stability and ensure generalisation

Consider the Tikhonov regularisation scheme,

$$\min_{\omega \in \mathbb{R}^d} \hat{\mathcal{E}}(f) + \lambda \|\omega\|^2$$

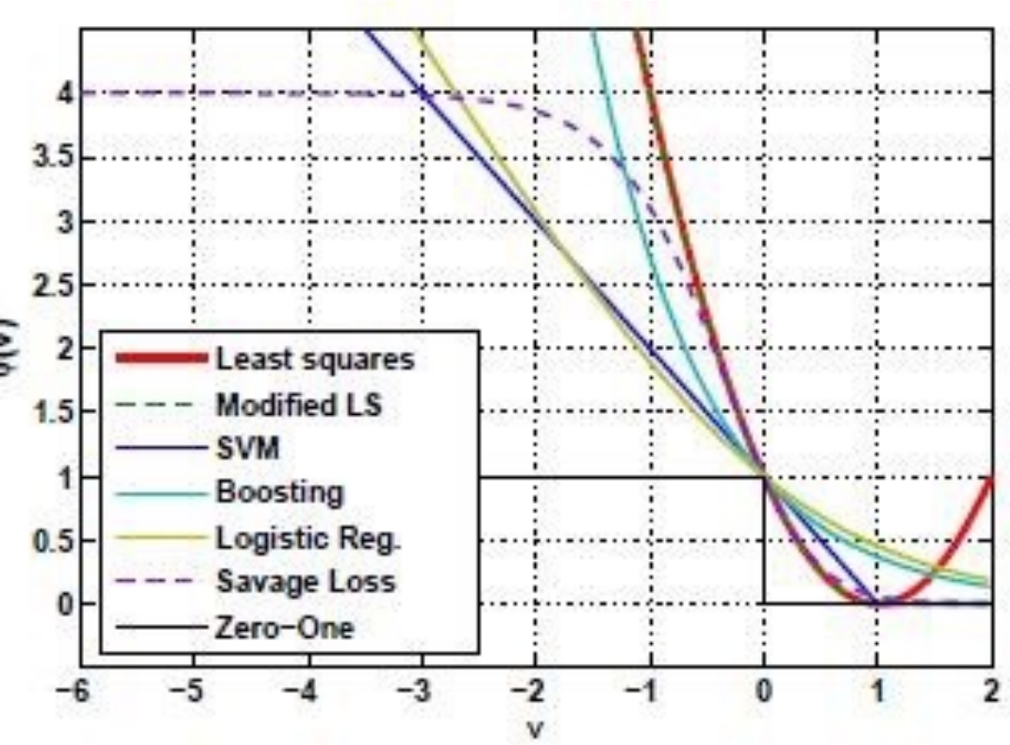
?

Tikhonov '62 Phillips '62

It describes a large class of methods sometimes called **Regularisation Networks.**

- ω is called the regulariser, which controls the stability of the solution and prevents overfitting.
- λ balances the error term and the regulariser.

Regularisation. Loss functions



- Different loss functions V induce different classes of methods.
- We will see common aspects and differences in considering different loss function.
- There exists no general computational scheme to solve Tikhonov Regularisation.
- The solution depends on the considered loss function.

The Regularised Least Squares Algorithm

Regularised Least Squares (RLS): Tikhonov regularisation

$$\min_{\omega \in \mathbb{R}^d} \hat{\mathcal{E}}(f) + \lambda \|\omega\|^2, \quad \hat{\mathcal{E}}(f) = \frac{1}{n} \sum_{i=1}^{n} V(y_i, f(x_i))$$

Square loss function:

$$V(y, f(x)) = (y - f(x))^2$$

We then obtain the RLS optimisation problem (linear model):

$$\min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - \omega^T x_i)^2 + \lambda \omega^T \omega, \quad \lambda \geq 0$$

Computations? **Statistics?**

The Regularised Least Squares Algorithm

$$\min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - \omega^T x_i)^2 + \lambda \omega^T \omega, \quad \lambda \geq 0$$

Computations?

Notation $\frac{1}{n} \sum_{i=1}^{n} (y_i - \omega^T x_i)^2 = \frac{1}{n} \|Y_n - X_n \omega\|^2$

Setting gradients.... $-\frac{2}{n} X_n^T (Y_n - X_n \omega), \quad \text{and}, \quad 2\omega$

... to zero $(X_n^T X_n + \lambda n I)\omega = X_n^T Y_n$

What does regularisation doing???

λ controls invertibility of $(X_n^T X_n + \lambda n I)$

The Regularised Least Squares Algorithm

$$\min_{\omega \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - \omega^T x_i)^2 + \lambda \omega^T \omega, \quad \lambda \geq 0$$

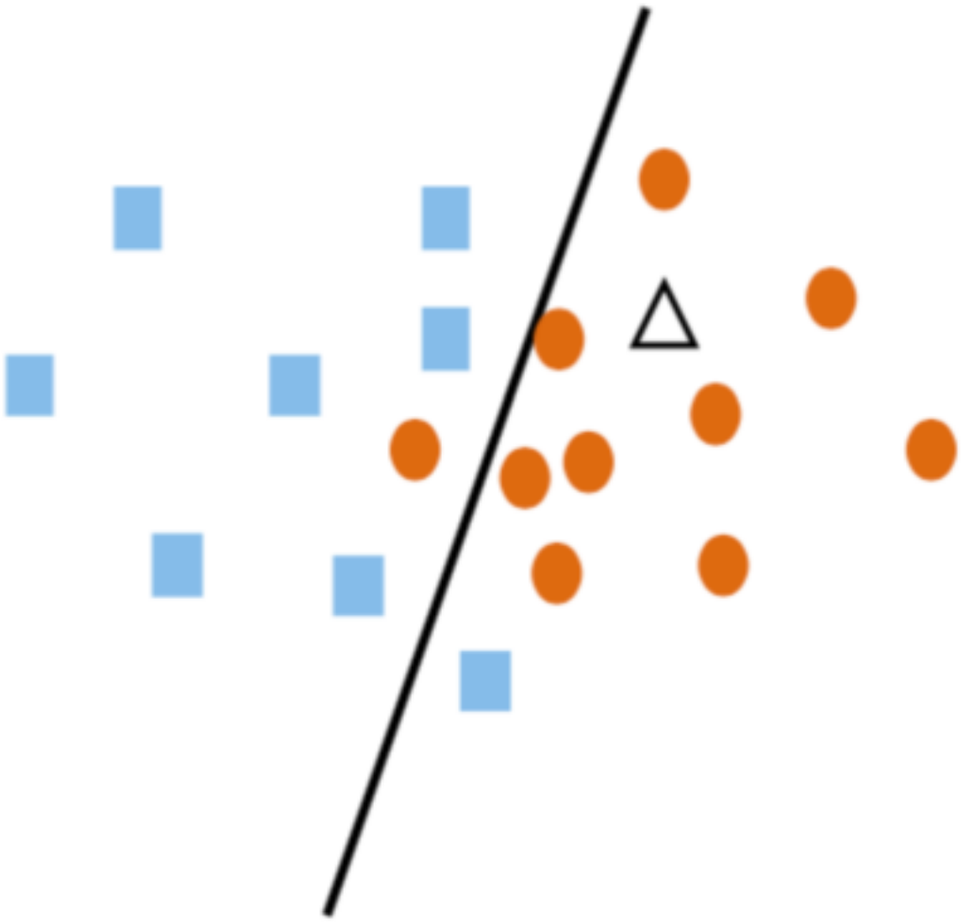
Statistics?

$$(X_n^T X_n + \lambda n I)\omega = X_n^T Y_n$$

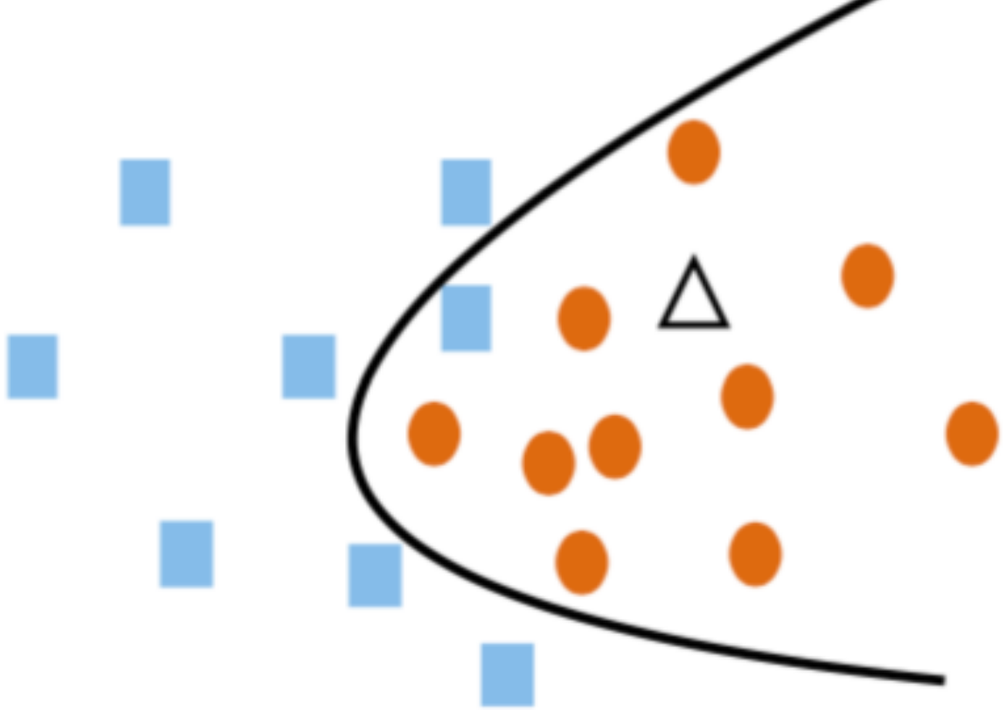
another story that shall be told another time

(Stein 1956, James and Stein 1961)

Shrinkage - Stein Effect- Admissible Estimator



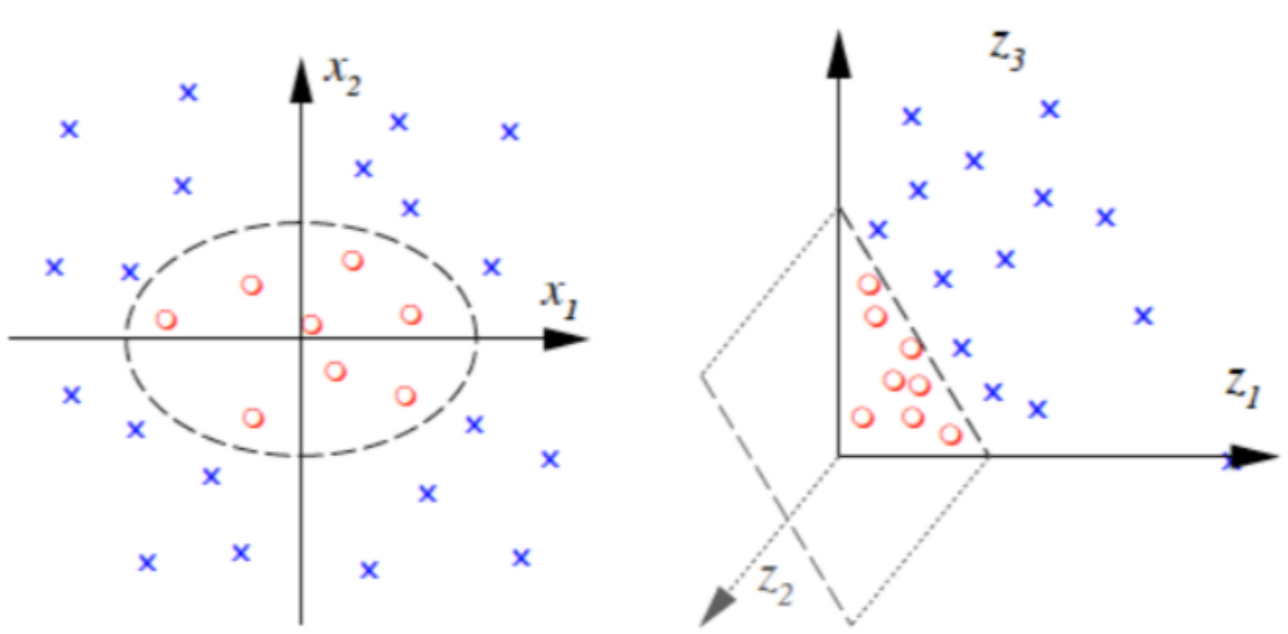
Why a linear decision rule?



Dictionaries

$$x \mapsto \tilde{x} = (\phi_1(x), \ldots, \phi_p(x)) \in \mathbb{R}^p$$

$$f(x) = \omega^T \tilde{x} = \sum_{j=1}^{p} \phi_j(x)\omega^j$$



$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$

$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

Computational Complexity

$$(X_n^T X_n + \lambda n I)\omega = X_n^T Y_n \longrightarrow (\tilde{X}_n^T \tilde{X}_n + \lambda n I)\omega = \tilde{X}_n^T Y_n$$

What about computational complexity?

Complexity vademecum

$M \in \mathbb{R}^{n \times p}$ is a matrix and $v, v' \in \mathbb{R}^p$ are $p-$dimensional vectors

- $v^T v' \mapsto O(p)$
- $Mv' \mapsto O(np)$
- $MM^T \mapsto O(n^2 p)$
- $(MM^T)^{-1} \mapsto O(n^3)$

Computational Complexity

$$(X_n^T X_n + \lambda n I)\omega = X_n^T Y_n \longrightarrow (\tilde{X}_n^T \tilde{X}_n + \lambda n I)\omega = \tilde{X}_n^T Y_n$$

What about computational complexity?

$$O(p^3) + O(p^2 n)$$

What if *p* is much larger than *n*?

$$(X_n^T X_n + \lambda n I)^{-1} X_n^T = X_n^T (X_n X_n^T + \lambda n I)^{-1}$$

$$\omega = X_n^T \underbrace{(X_n X_n^T + \lambda n I)^{-1} Y_n}_{c} = \sum_{i=1}^{n} x_i^T c_i$$

Computational complexity: $O(n^3) + O(pn^2)$

Kernels

$$(X_n^T X_n + \lambda n I)^{-1} X_n^T = X_n^T (X_n X_n^T + \lambda n I)^{-1}$$

$$\omega = X_n^T \underbrace{(X_n X_n^T + \lambda n I)^{-1} Y_n}_{c} = \sum_{i=1}^{n} x_i^T c_i$$

$$f(x) = x^T \omega = \sum_{i=1}^{n} \underbrace{x^T x_i}_{K(x,x_i)} c_i$$

$$c = (K_n + \lambda n I)^{-1} Y_n, \qquad (K_n)_{i,j} = K(x_i, x_j)$$

- linear kernel $K(x, x') = x^T x'$
- polynomial kernel $K(x, x') = (x^T x' + 1)^d$
- Gaussian kernel $K(x, x') = e^{-\|x - x'\|^2 / 2\sigma^2}$

Things I won't tell you about this time….

$$\hat{f}(x) = \sum_{i=1}^{n} K(x_i, x) c_i$$

- Reproducing Kernel Hilbert Spaces
- Gaussian Processes
- Integral Equations
- Sampling Theory / Inverse Problems

- Multi-task, labels, outputs, classes

Things I will tell you about tomorrow ….

- Loss functions: SVM, Logistic Regression…

End of sub-Part II

We considered the regularised empirical risk minimisation with squared loss function, leading to regularised least squares algorithm. We discussed how to deal with non linear problems via kernels and dictionaries.

Regularised Least Squares - Dictionaries - Kernels

Today's Afternoon Lecture:

Machine Learning in Practice