Dealing with Data: Machine Learning and Data Mining

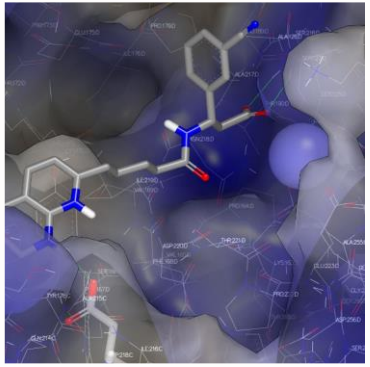
Illustrations, Applications, and Research

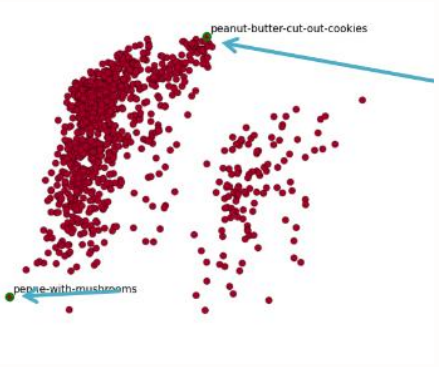
$$k_\times(G_1, G_2) = \mathbf{1}^\top \left[\lim_{n\to\infty} \sum_{i=0}^{n} \eta_i E_\times^i \right] \mathbf{1}$$

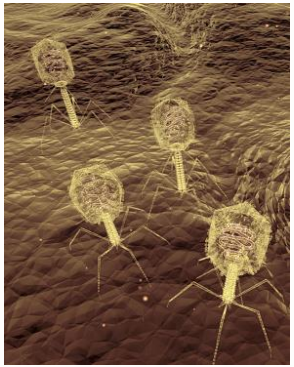
$$O\left(\log_2^k(2\mathbf{r}) + h\mathbf{r}^3\right)$$

Thomas Gärtner

Professor of Machine Learning
Faculty of Informatics







about.us

Prof. Thomas Gärtner: Computational aspects of machine learning.

Maximilian Thiessen: Graph-based machine learning, active learning, and learning theory. Connects (graph) convexity theory with machine learning.

Tamara Drucks: Background in bioinformatics and theoretical computer science. Now interested in graphs, optimization, and (deep) learning theory.

Fabian Jogl: Machine learning on graphs and especially graph neural networks. Combining ideas from algorithmics with machine learning.

David Penz: Background in economics and theoretical computer science. Working on disentangled representations, knowledge transfer, and neuro-symbolic AI.

Patrick Indri: Background in physics and data science. Interested in trustworthy machine learning focusing on privacy and robustness.

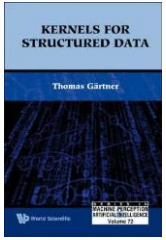
Anice Jahanjoo: Machine learning in Health Monitoring Systems and SAR Image processing.

about.me

Dipl-Ing (BA Mannheim, ‘99)
Information-technology

MSc (U Bristol, ‘00)
Machine Learning

Dr rer nat (U Bonn, ‘05)
Kernels for Structured Data



Emmy Noether (DFG, U Bonn, ‘10-’15)
Constructive ML $\approx 1.6M$ Euro

University of Nottingham (‘15-’19)
Professor of Data Science

TU Wien (’19-?)
Professor of Machine Learning

collaborations

Awards
‘00, ‘03, ‘05, ‘06, ’07, ’09, ‘15, ’16, ‘19

Workshop Co-Chair
CML’13, ’15, ’16; MLG ’06

Area Chair/SPC
IJCAI ’19, ICLR ’21
NeurIPS ’23 - ’18, ’17, ’15
ECML/PKDD ’23 - ’19, ‘15, ‘14, ‘13

Program Co-Chair ECML/PKDD’18

Journal-Track Co-Chair ECML/PKDD’16

Editor for **Machine Learning**

0

1

0

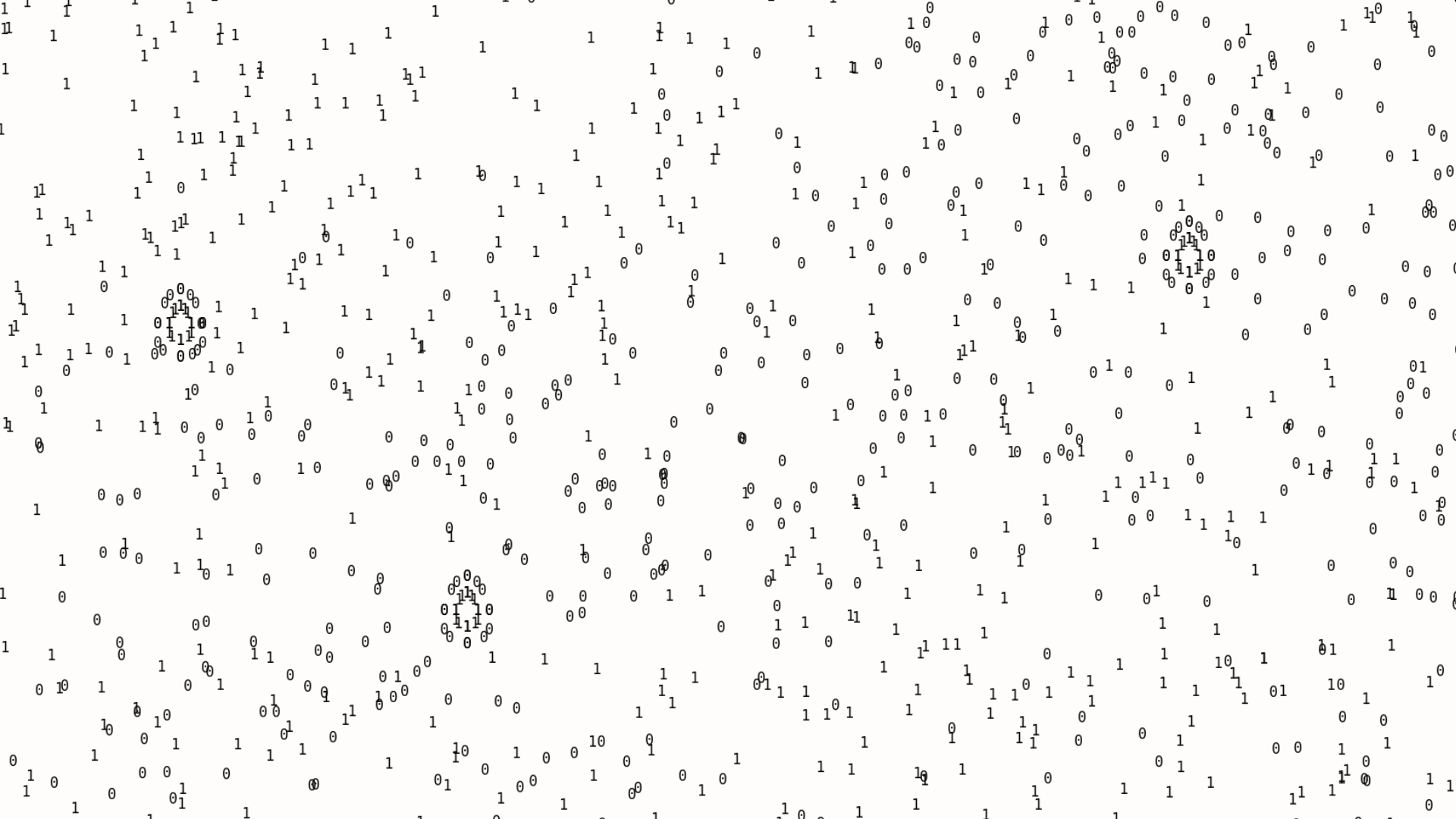
0

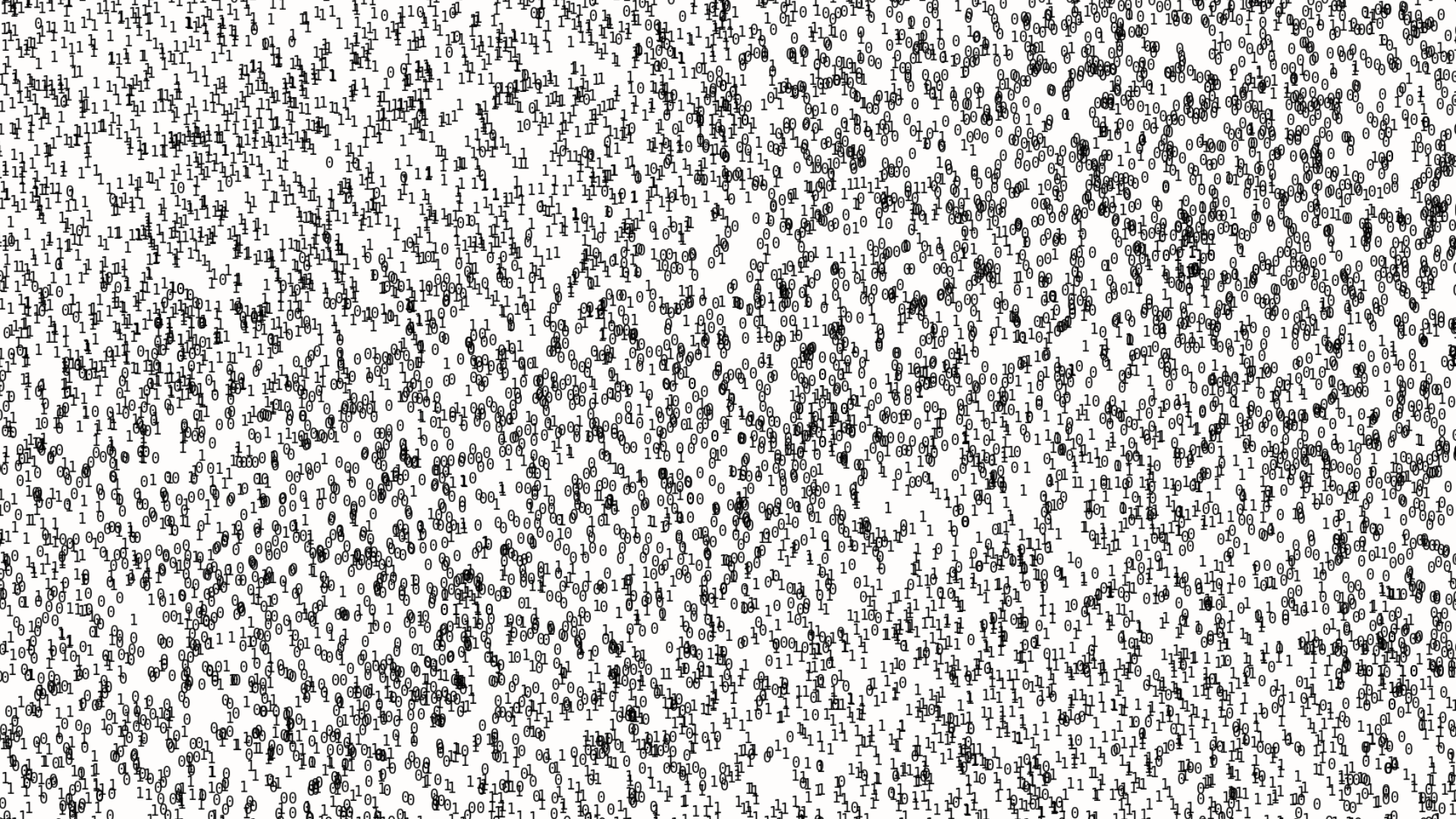
1

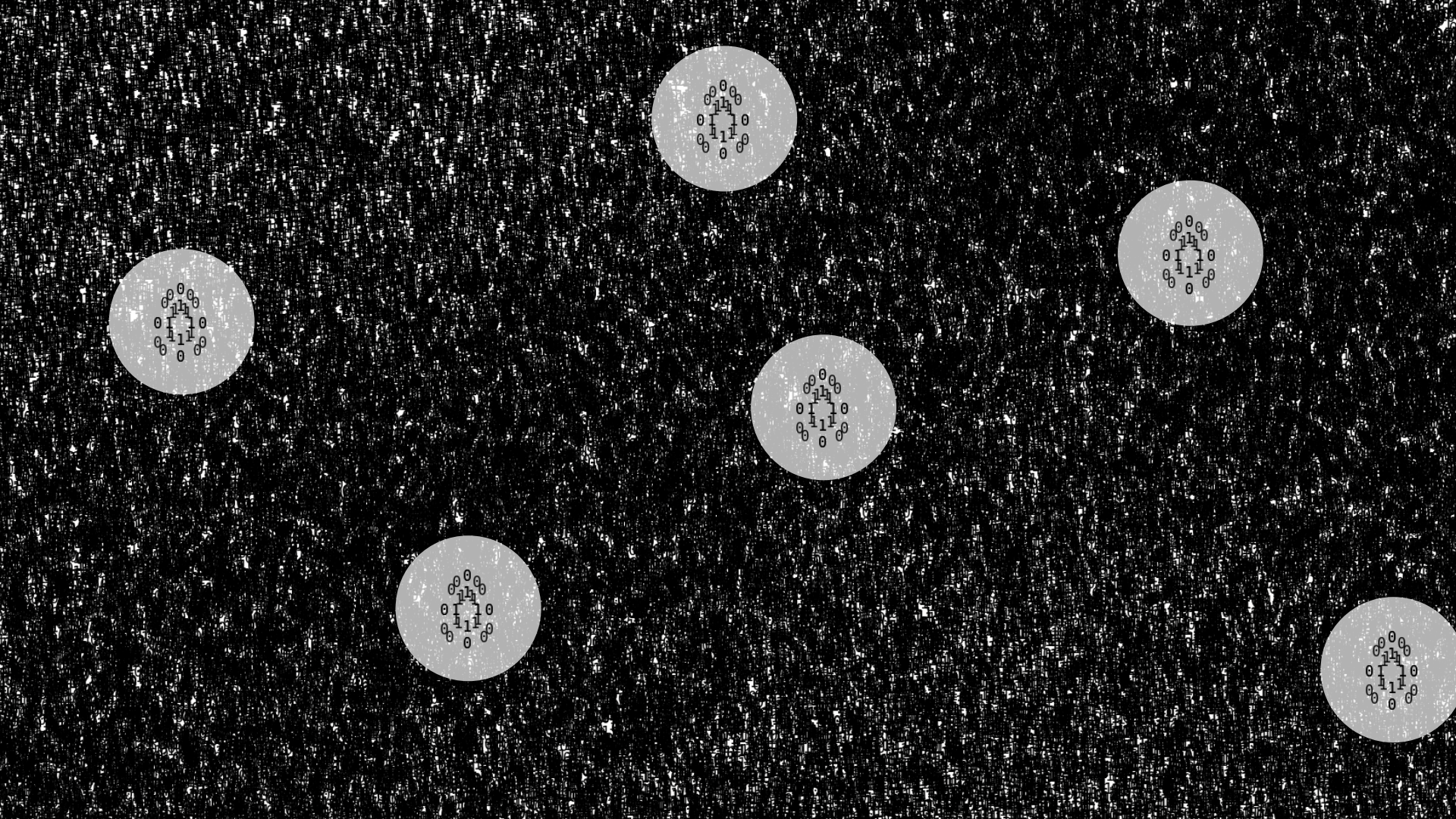
1

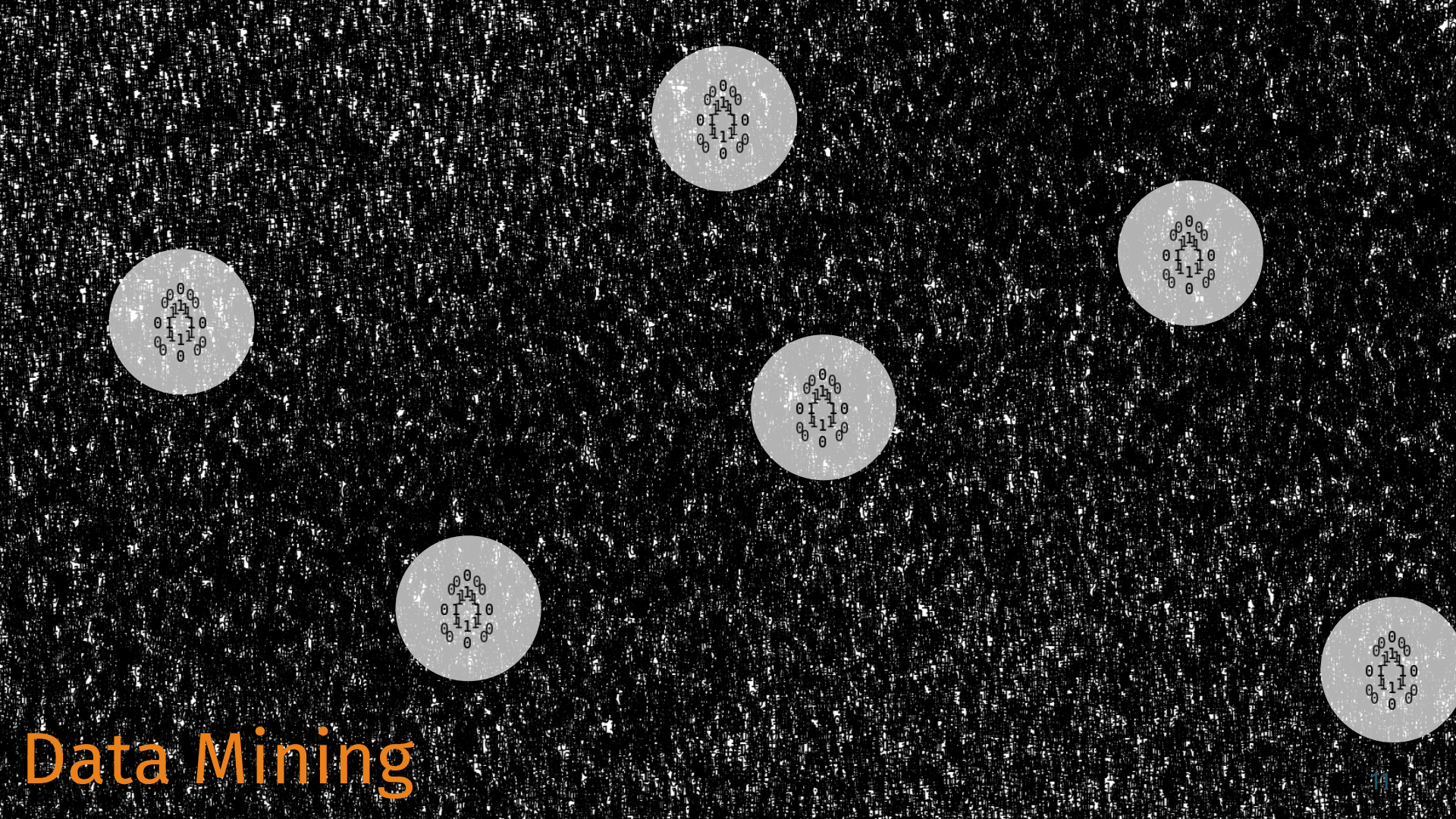
0

0

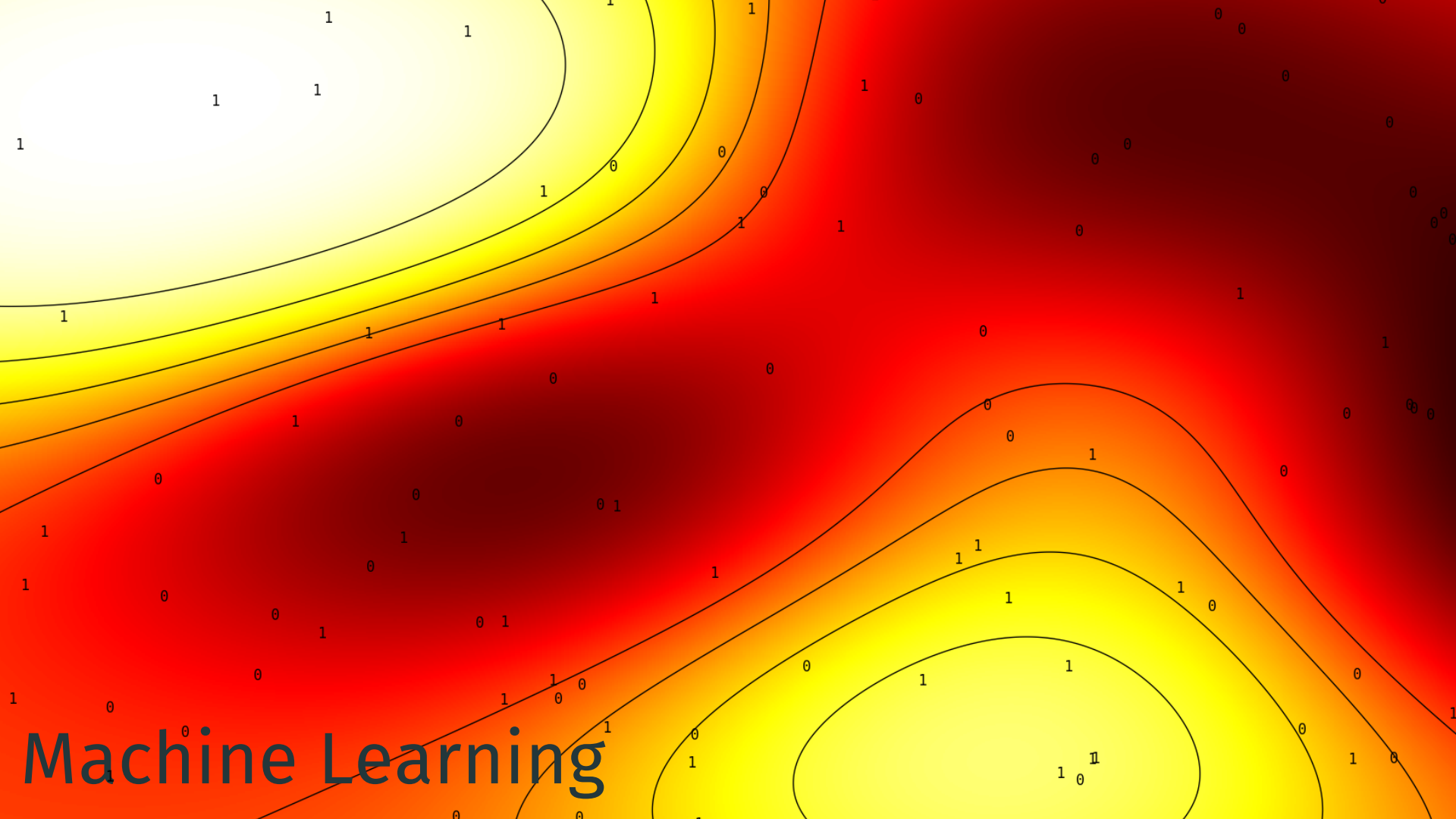








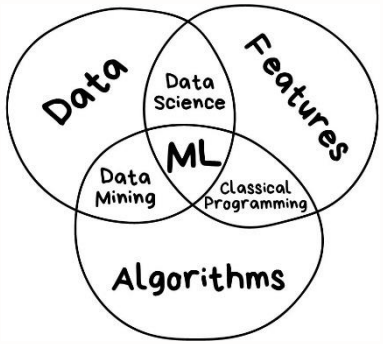
Data Mining



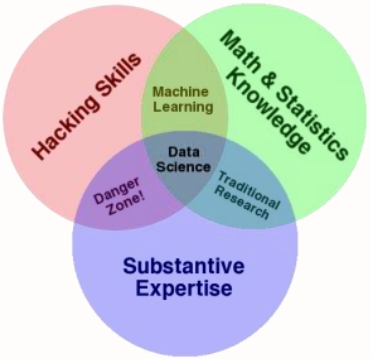
Machine Learning

Datalogy, Data Science, ...

Peter Naur (1974):
*The **science of dealing with data**, once they have been established, while the relation of the data to what they represent is delegated to other fields and sciences.*



(Vasily Zubarev, 2018)



(Drew Conway, 2010)

John W. Tukey:
The best thing about being a statistician is that you get to play in everyone's backyard

Don't forget that statisticians are the free-est of all scientists — they can work on anything. Take advantage.

John Quackenbush (2014):
Every revolution in science**—from Copernican heliocentric model to the rise of statistical and quantum mechanics, from Darwin's theory of evolution and natural selection to the theory of the gene—**has been driven by one and only one thing: access to data

Data Mining

Machine Learning

Artificial Intelligence

Pattern Recognition

Data Science

Data

Knowledge Discovery

Big Data

Deep Learning

Statistics

Data Analysis/Analytics

Information Processing

r Vision, Speech Recognition, Recommendation Systems, Computational Intelligence, **Datalogy**, Computational/Statistical/Algorithmic Learning Theory, Computational Statistics, Information Retrieval

“Hype or Hallelujah”?

(illustrations and examples of what ML can/cannot do)

What is your data?

Data—Example: Cars

Leistung
114 kW / 155 PS

0 - 100 km/h
8 s

Höchstgeschwindigkeit
215 km/h

CO2-Ausstoß (komb.)*
153 g/km

Verbrauch (komb.)*
6,5 l/100km

Energieeffizienzklasse*
-

Daten Motor

Kraftstoffe	Benzin
Zylinder/-anordnung	4/Reihe
Hubraum	1.368 cm³
Leistung	114 kW / 155 PS
Drehmoment	230 Nm
Beschleunigung	8 s (0-100 km/h)
Höchstgeschwindigkeit	215 km/h
Verbrauch (komb.)*	6,5 l/100 km
CO2-Ausstoß (komb.)*	153 g/km
Getriebe	Manuelles Getriebe
Antrieb	Vorderradantrieb
Tankvolumen	45 l
Anhängelast (gebremst)	500 kg

Allgemeine Modell Daten

Leistung	51 - 125 KW, 70 - 170 PS
Höchstgeschwindigkeit	160 - 219 km/h
Sitzplätze	4
Verbrauch (komb.)*	3,4 - 6,5 l/100 km
Antrieb	Vorderradantrieb
Kraftstoff	Benzin, Diesel
Getriebe	Automatik, Manuelles Getriebe

Maße und Abmessungen

Länge	4.063 mm
Breite	1.720 mm
Höhe	1.446 mm
Radstand	2.511 mm
Ladevolume	950 l

Machine Learning in the Media *(i)*

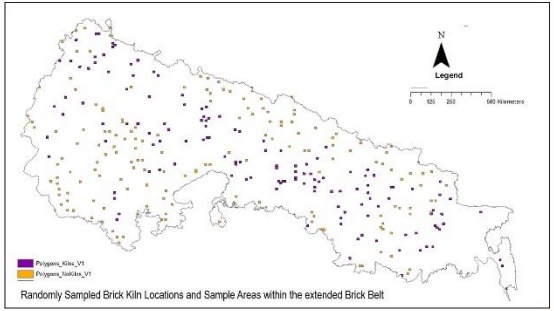
Machine Learning in the Media *(ii)*

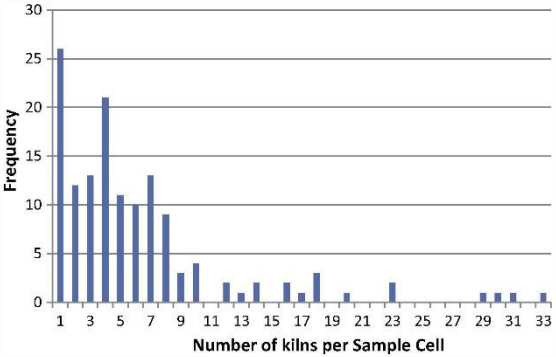
[Center for Data Innovation]

Data Science

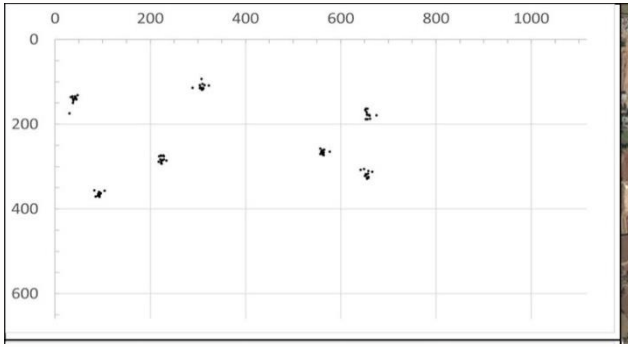
For Social Good

Active Search for Modern Slavery Sites (eg Brick Kilns @ Rights Lab)





Aim: use active search to minimise the amount of manual expert supervision needed



was released in [...] 2003

a space-based, persistent world, massively multiplayer online role-playing game (MMORPG)

players of Eve Online can participate in a number of in-game professions and activities, including mining, piracy, manufacturing, trading, exploration, and combat

a total of 7,800 star systems that can be visited by players.

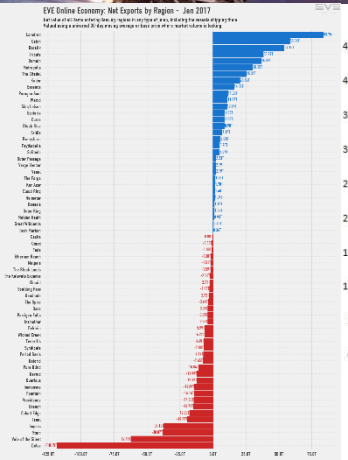
In February 2013, *Eve Online* reached over 500,000 subscribers

Eve Online is functionally a single-universe game. (Wikipedia)

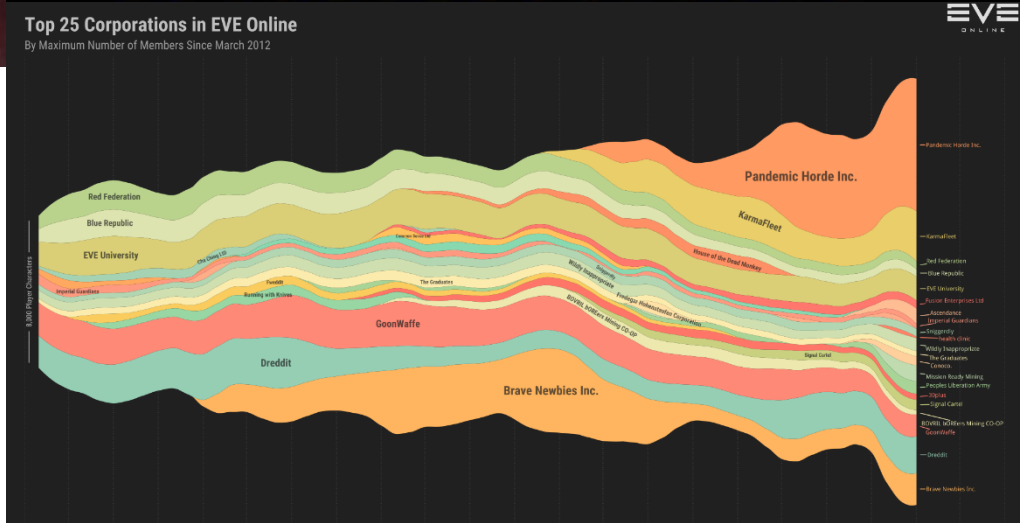
Virtual world hires real economist
(Reuters)

EVE®
ONLINE

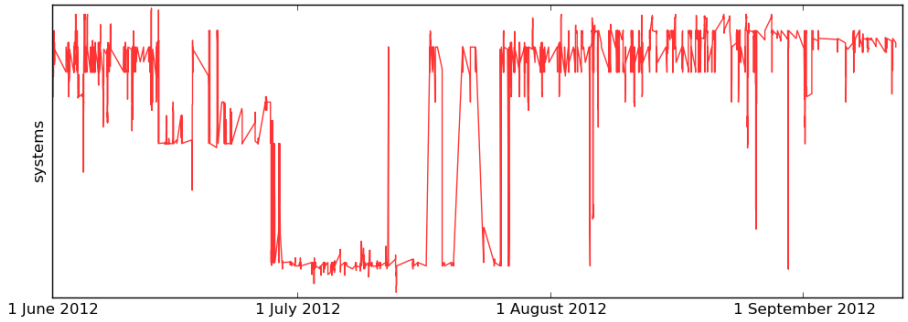
Trade value vs. number of inhabitants
Inhabitants
Trade value
Billions
200,000
180,000
160,000
140,000
120,000
100,000
80,000
60,000
40,000
20,000
0
1,800
1,600
1,400
1,200
1,000
800
600
400
200
0
2003-10-07
2004-01-07
2004-04-07
2004-07-07
2004-10-07
2005-01-07
2005-04-07
2005-07-07
2005-10-07
2006-01-07
2006-04-07
2006-07-07
2006-10-07
2007-01-07
2007-04-07

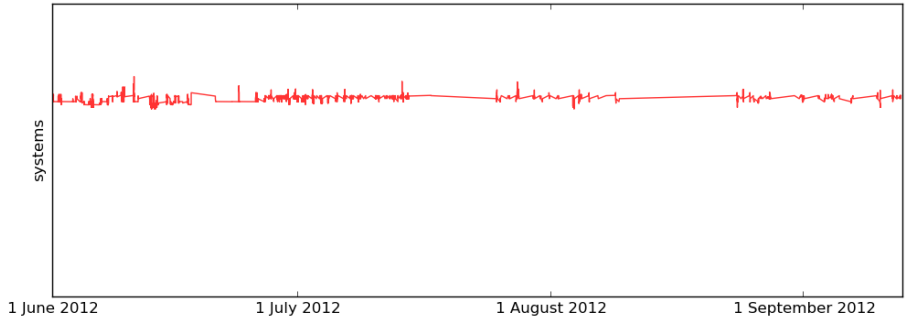

EVE Online Economy: Net Exports by Region - Jan 2017

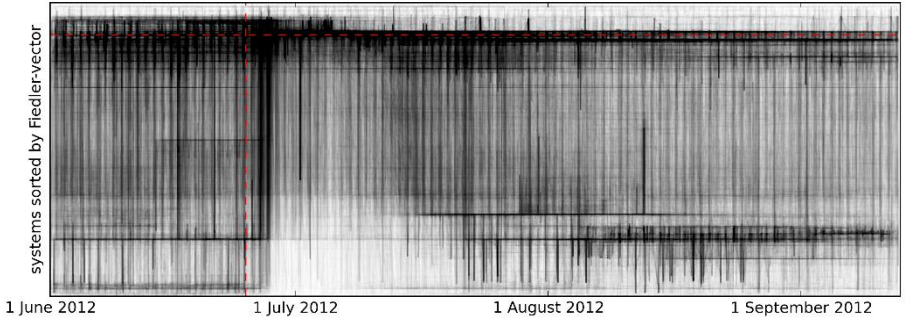
ISK Prices from June 2007 to November 2011
© www.mmobux.com
Highest: $39.96
on June 19, 2007
Lowest: $9.34
on June 18, 2009
45.00
40.00
35.00
30.00
25.00
20.00
15.00
10.00
5.00
0.00
June 13, 2007
August 13, 2007
October 13, 2007
December 13, 2007
February 13, 2008
April 13, 2008
June 13, 2008
August 13, 2008
October 13, 2008
December 13, 2008
February 13, 2009
April 13, 2009
June 13, 2009
August 13, 2009
October 13, 2009
December 13, 2009
February 13, 2010
April 13, 2010
June 13, 2010
August 13, 2010
October 13, 2010
December 13, 2010
February 13, 2011
April 13, 2011
June 13, 2011
August 13, 2011
October 13, 2011


Top 25 Corporations in EVE Online
By Maximum Number of Members Since March 2012
Pandemic Horde Inc.
KarmaFleet
Red Federation
Blue Republic
EVE University
GoonWaffe
Dreddit
Brave Newbies Inc.

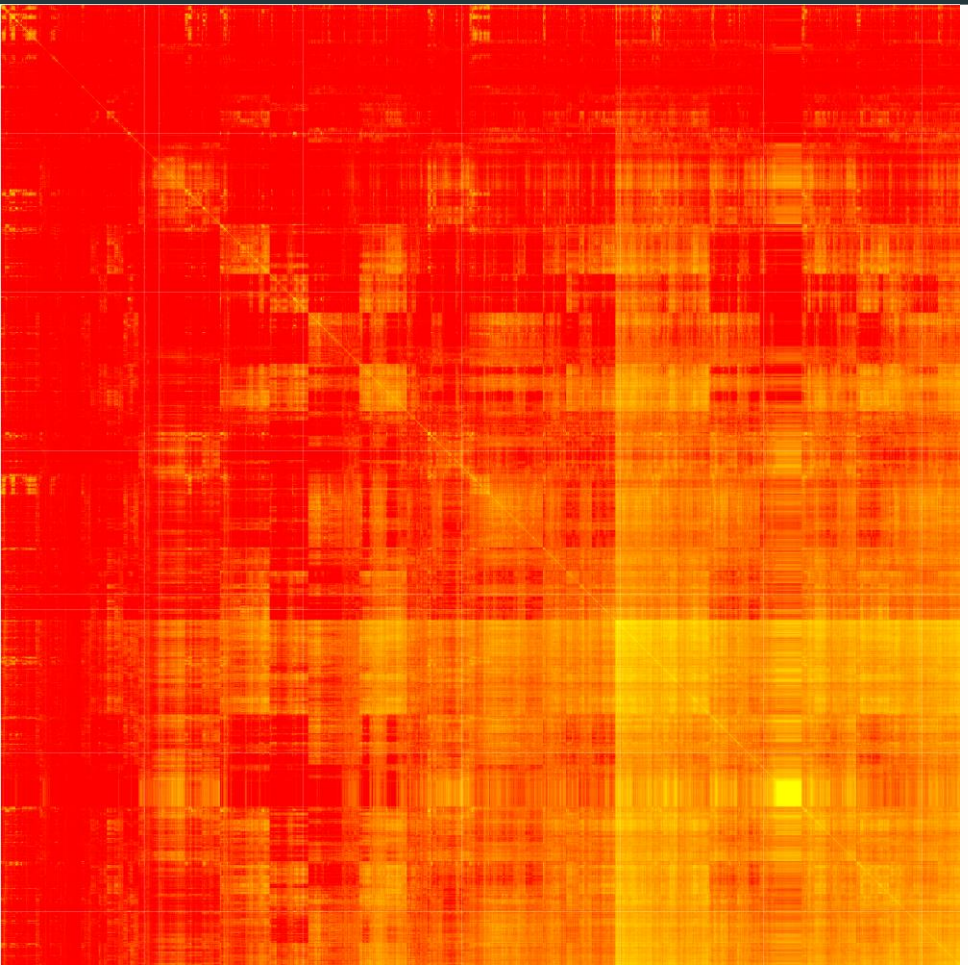
Load Balancing

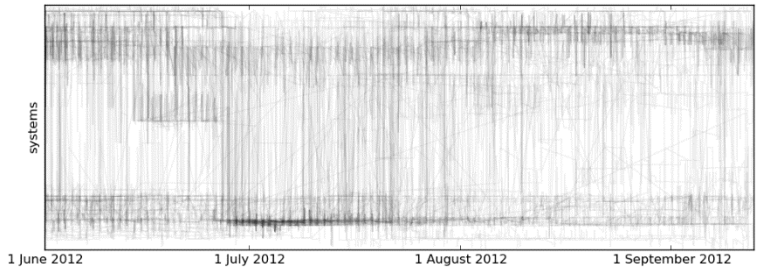


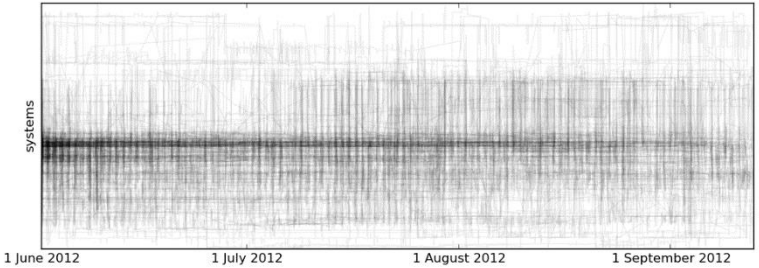


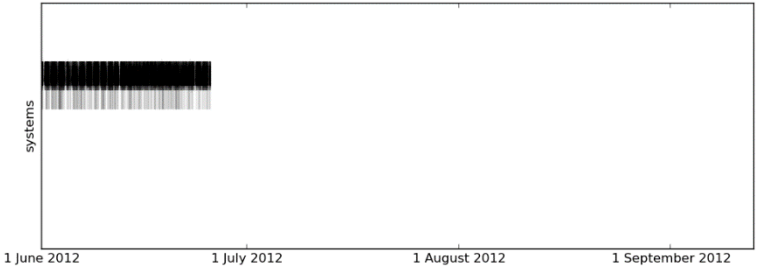


Group detection

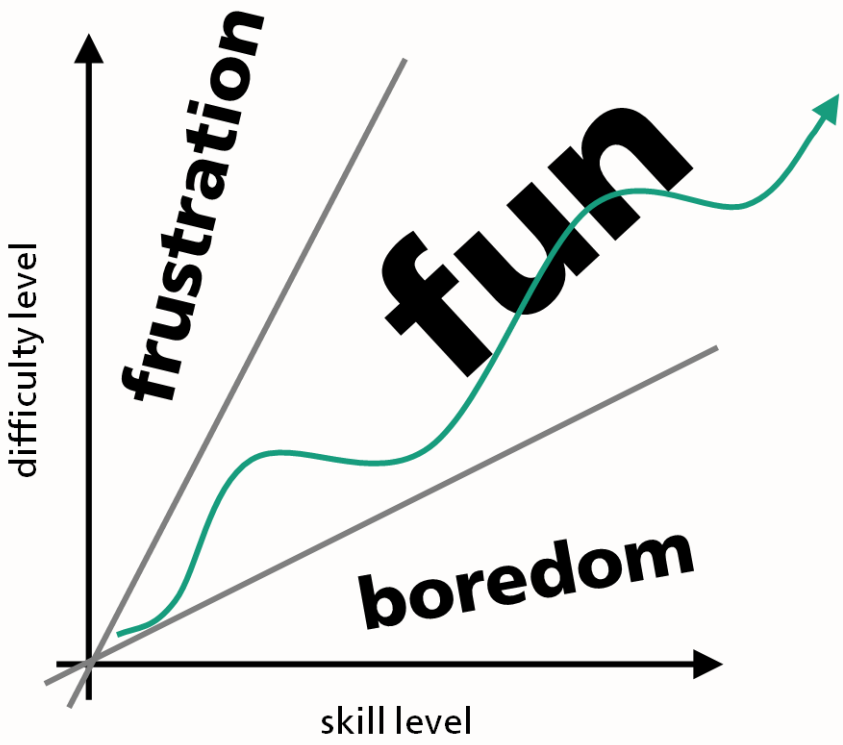






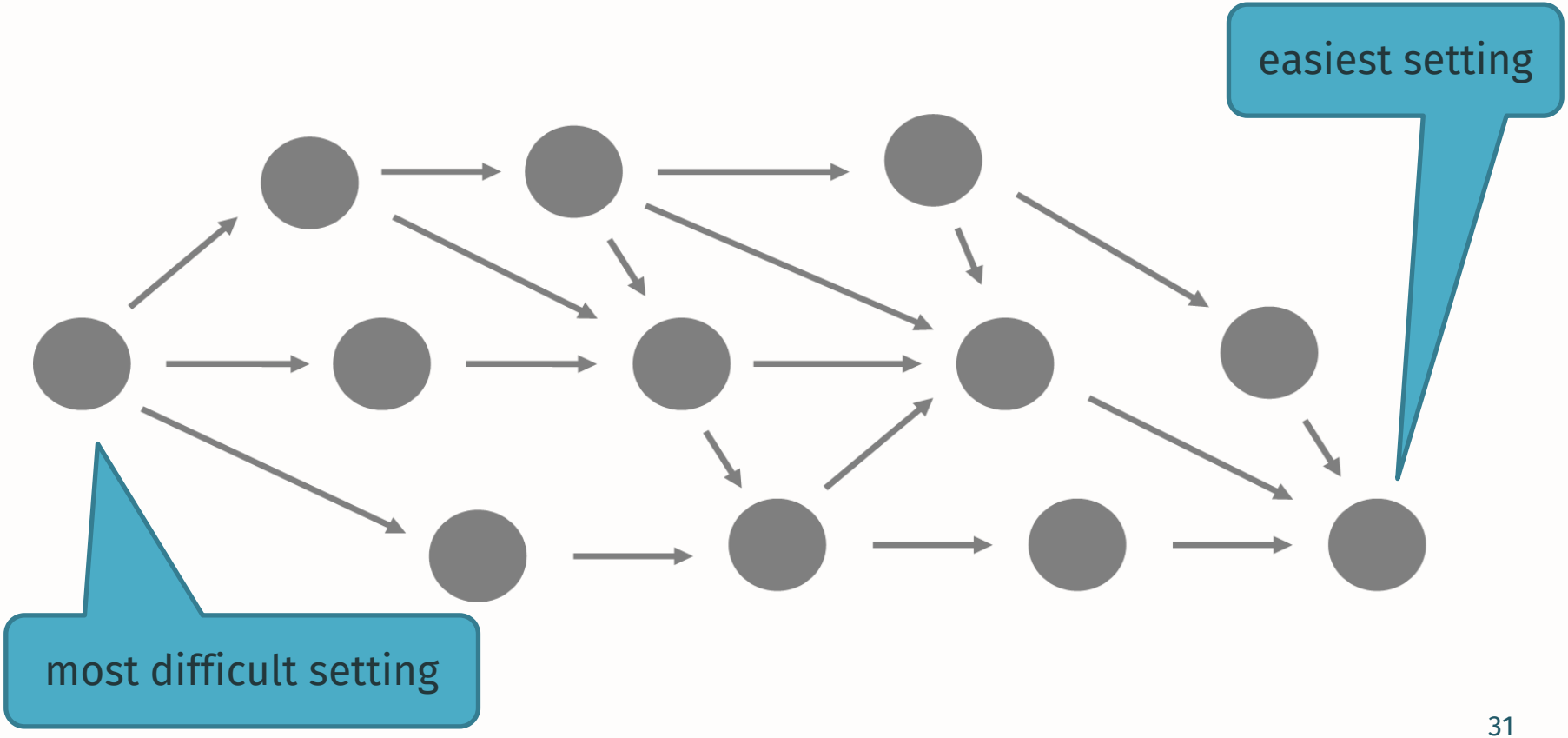


Automagic: Dynamical Difficulty Adjustment in Computer Games

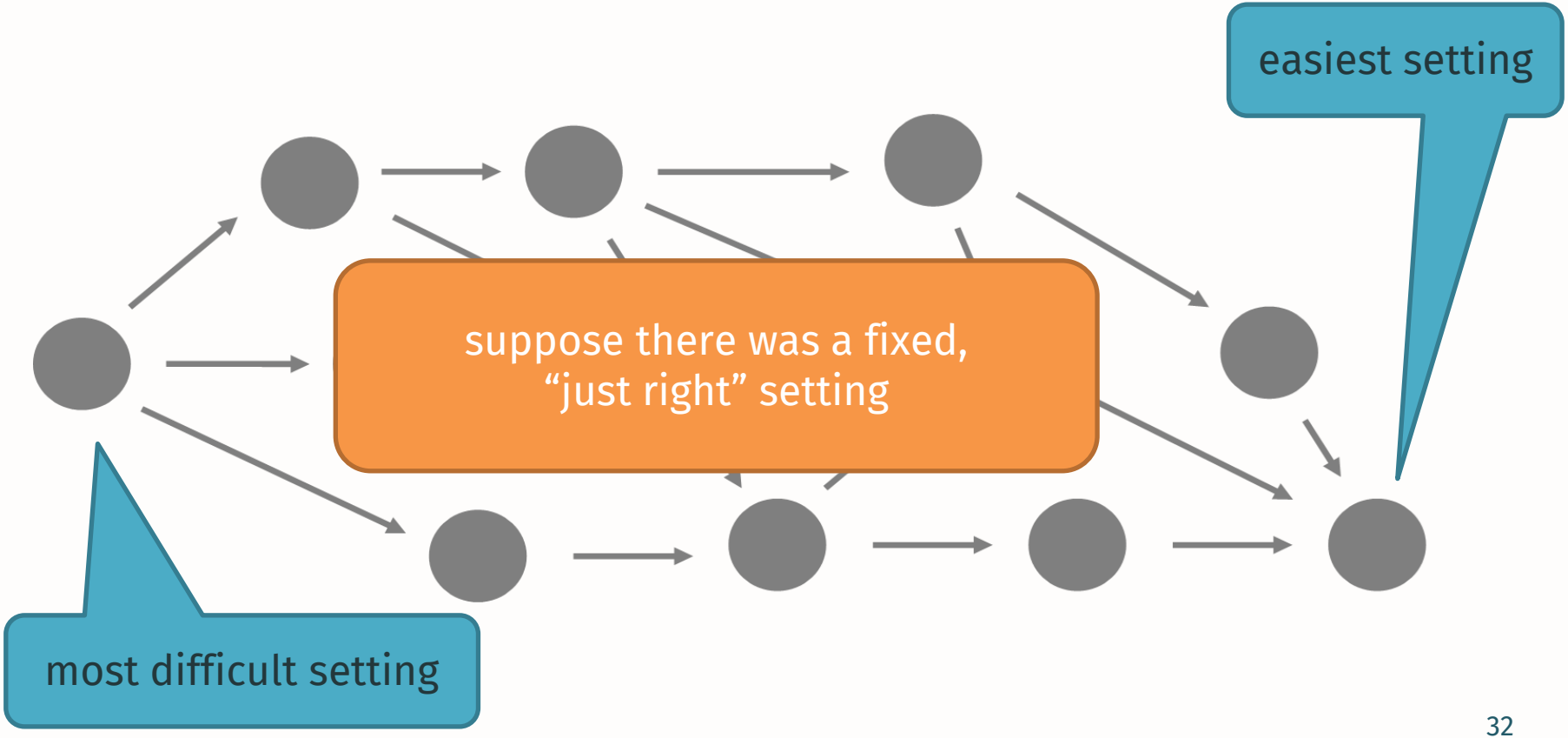


Traditional Difficulty Adjustment

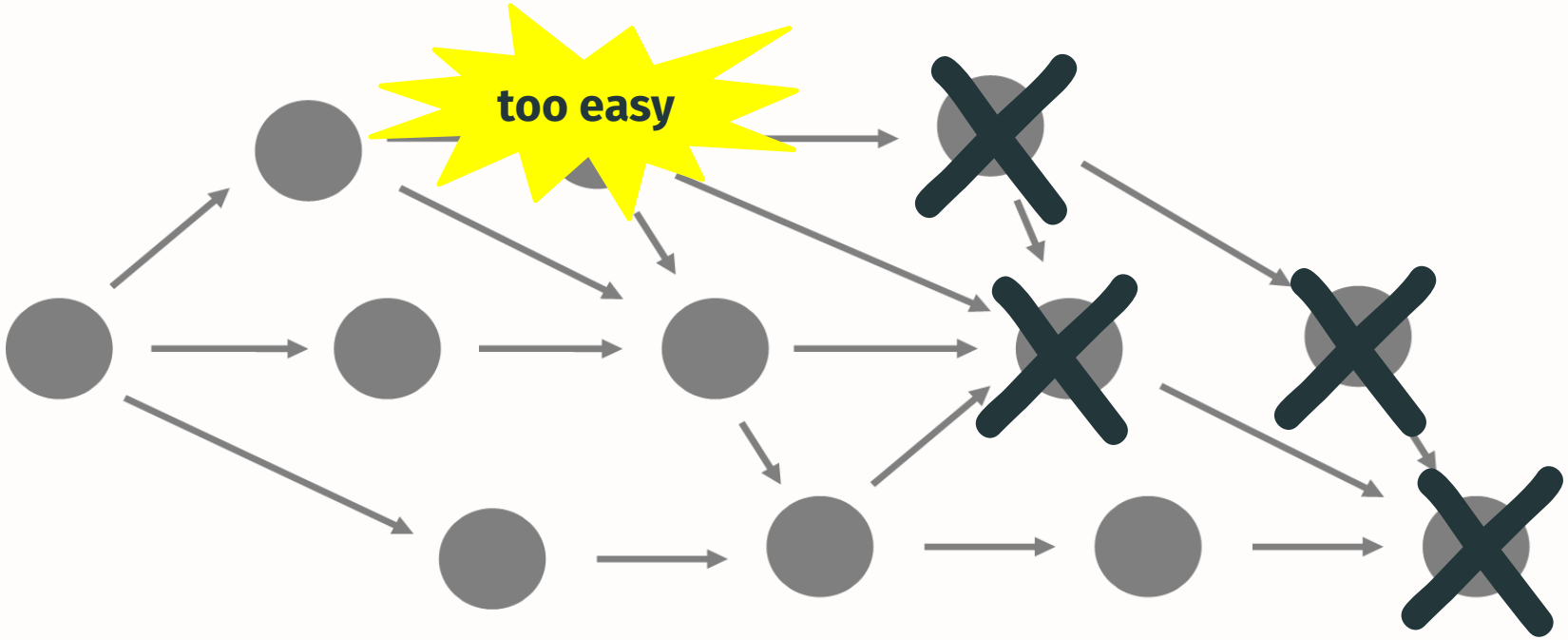
Difficulty Modelling



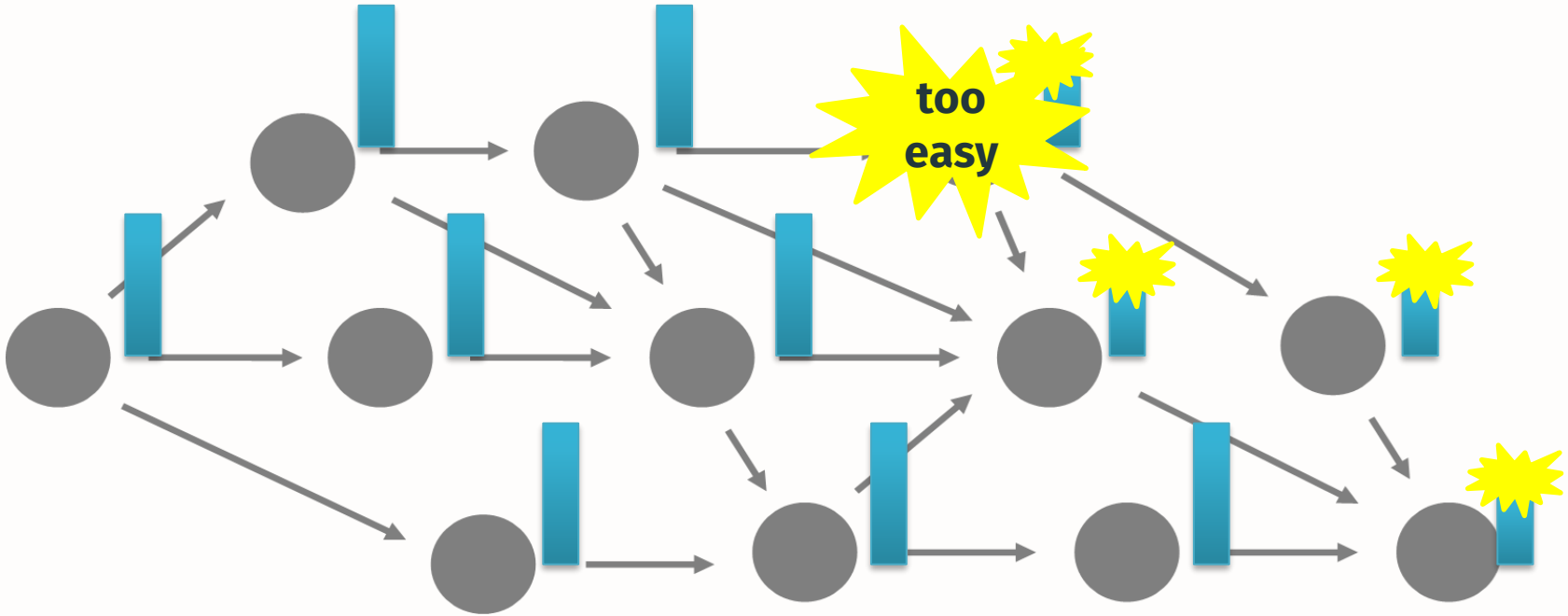
Static Difficulty Adjustment



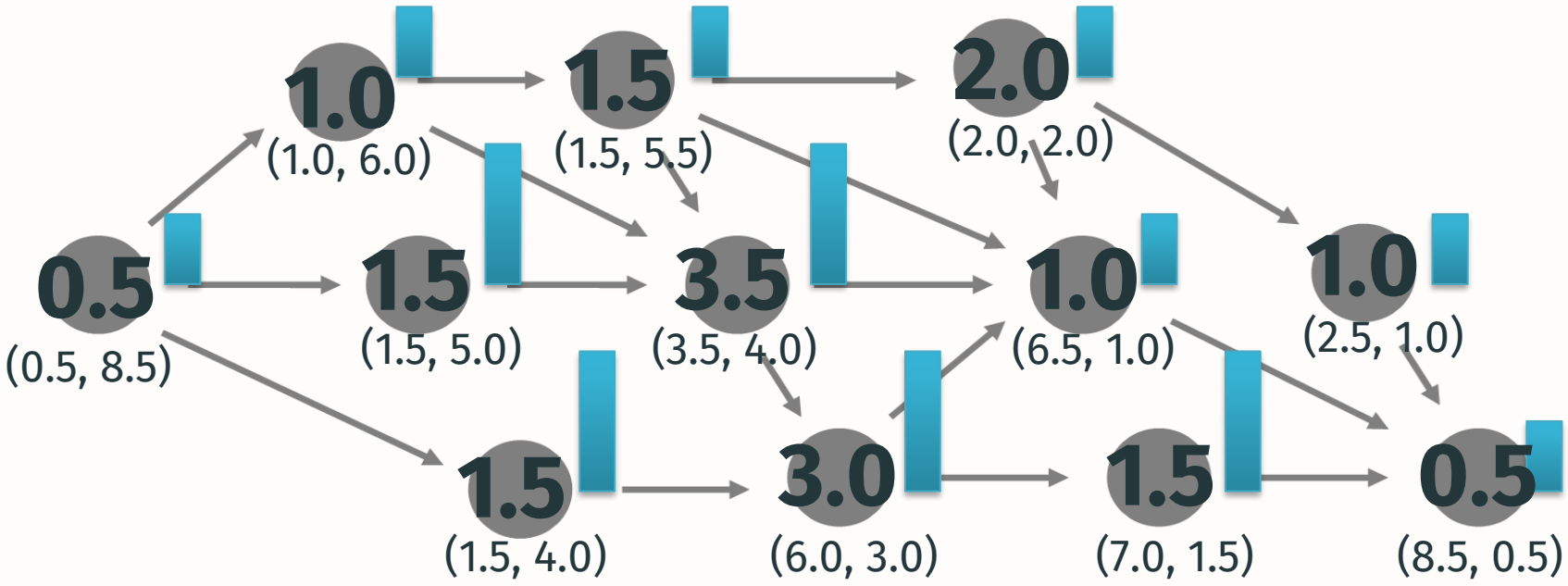
Observations lead to deductions



Belief update in partially ordered sets



Pick vertex maximising the min(up, down)-belief



simple algorithm with a guaranteed bound on the number of mistakes can (easily) be generalised to other structured spaces

Dynamic Difficulty Adjustment

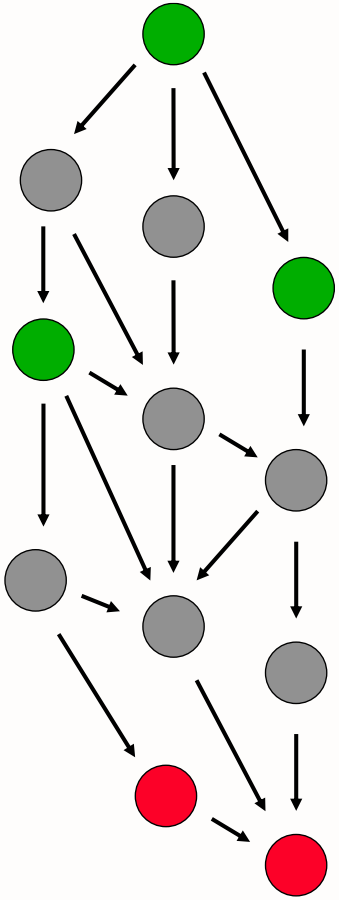
1) **game master predicts** a difficulty setting

2) **player plays** one round of the game in this setting

3) **game master observes** whether the setting was too easy, just right, or too difficult

this game is played on a partially ordered set reflecting the **`is more difficult'-relation**
the goal of the game master is to predict `just right'-settings as often as possible

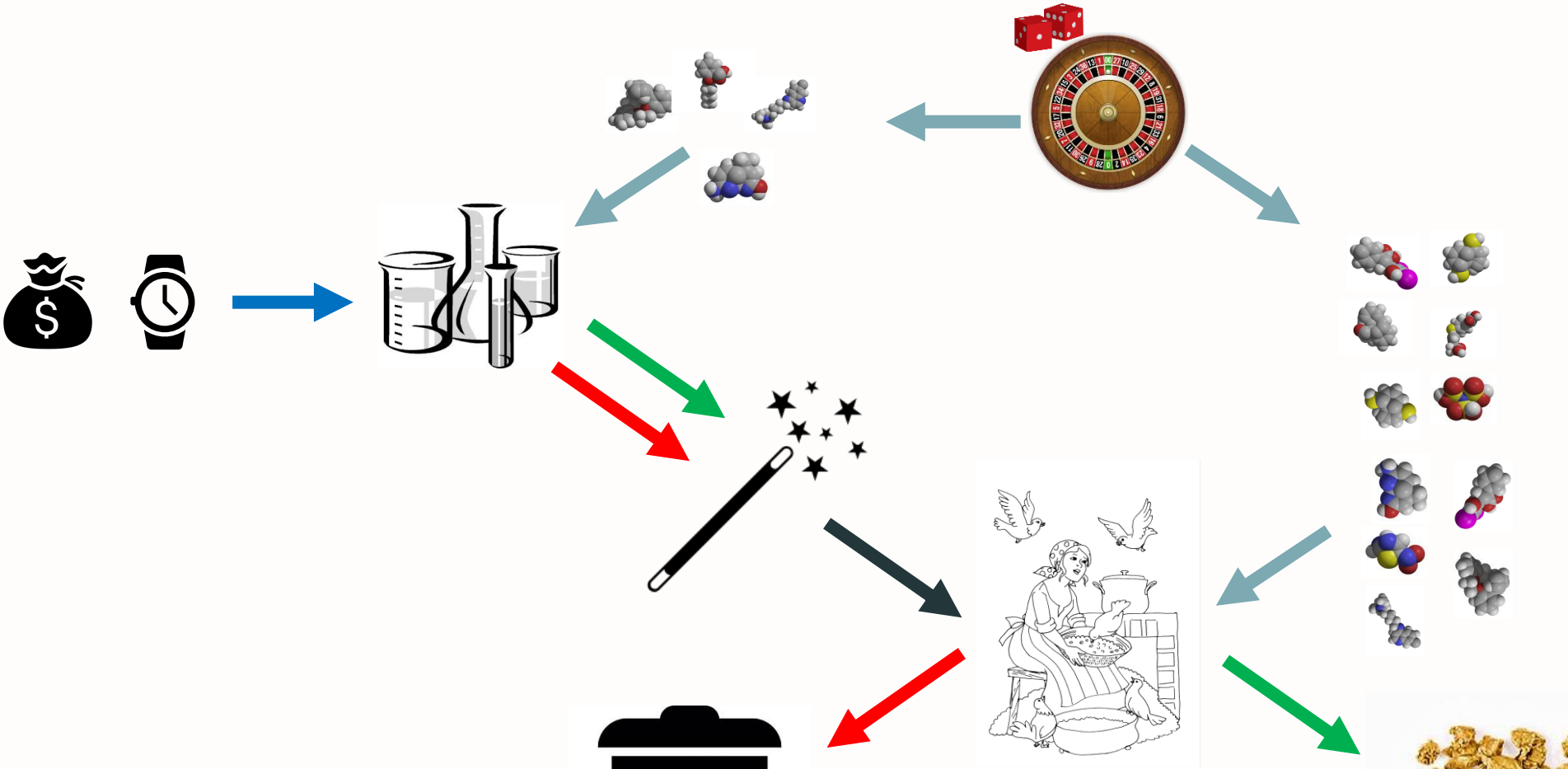
we can bound the **number of mistakes** in this setting



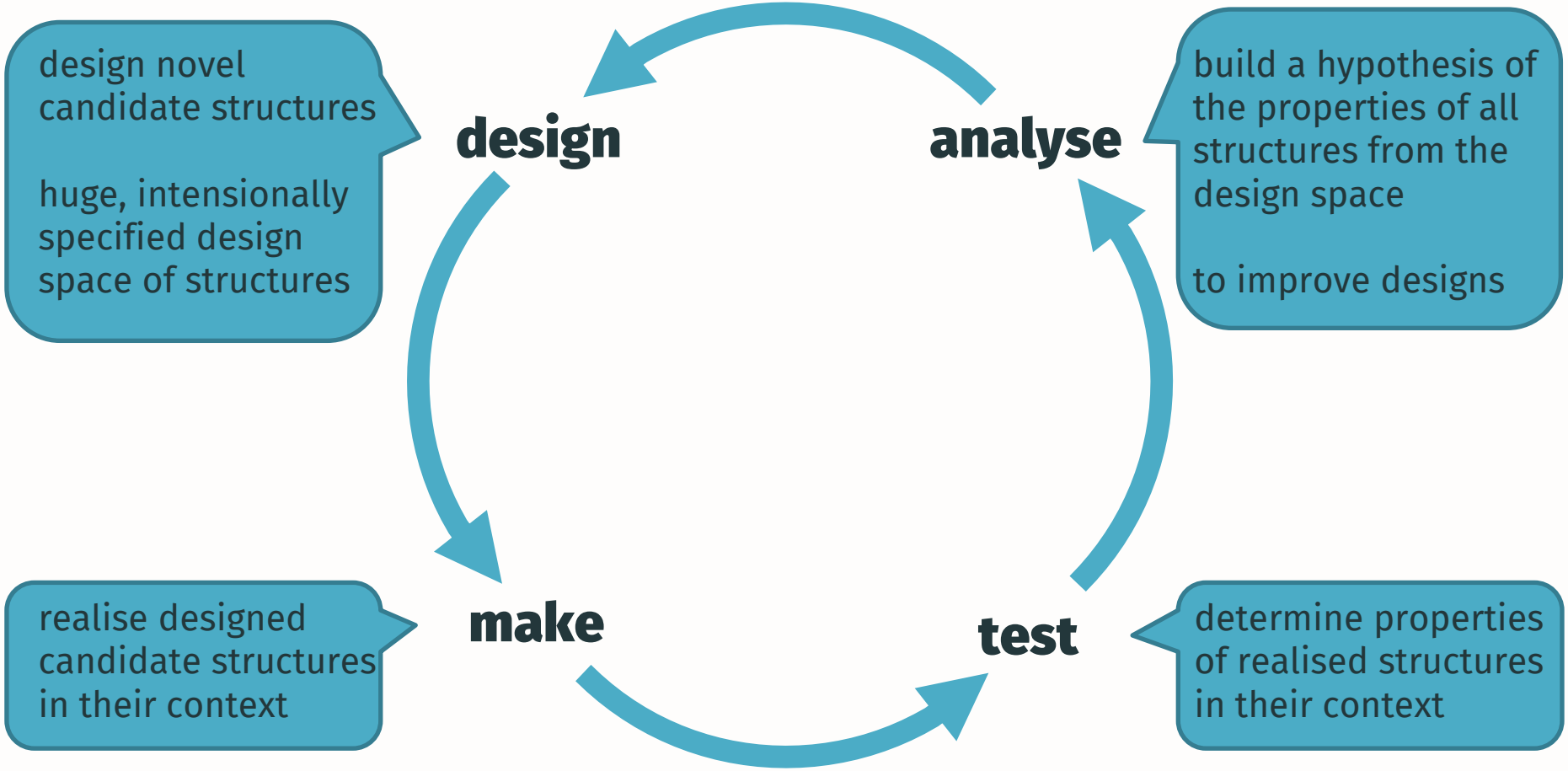
Use-Cases

Drug discovery from a supervised learning point of view

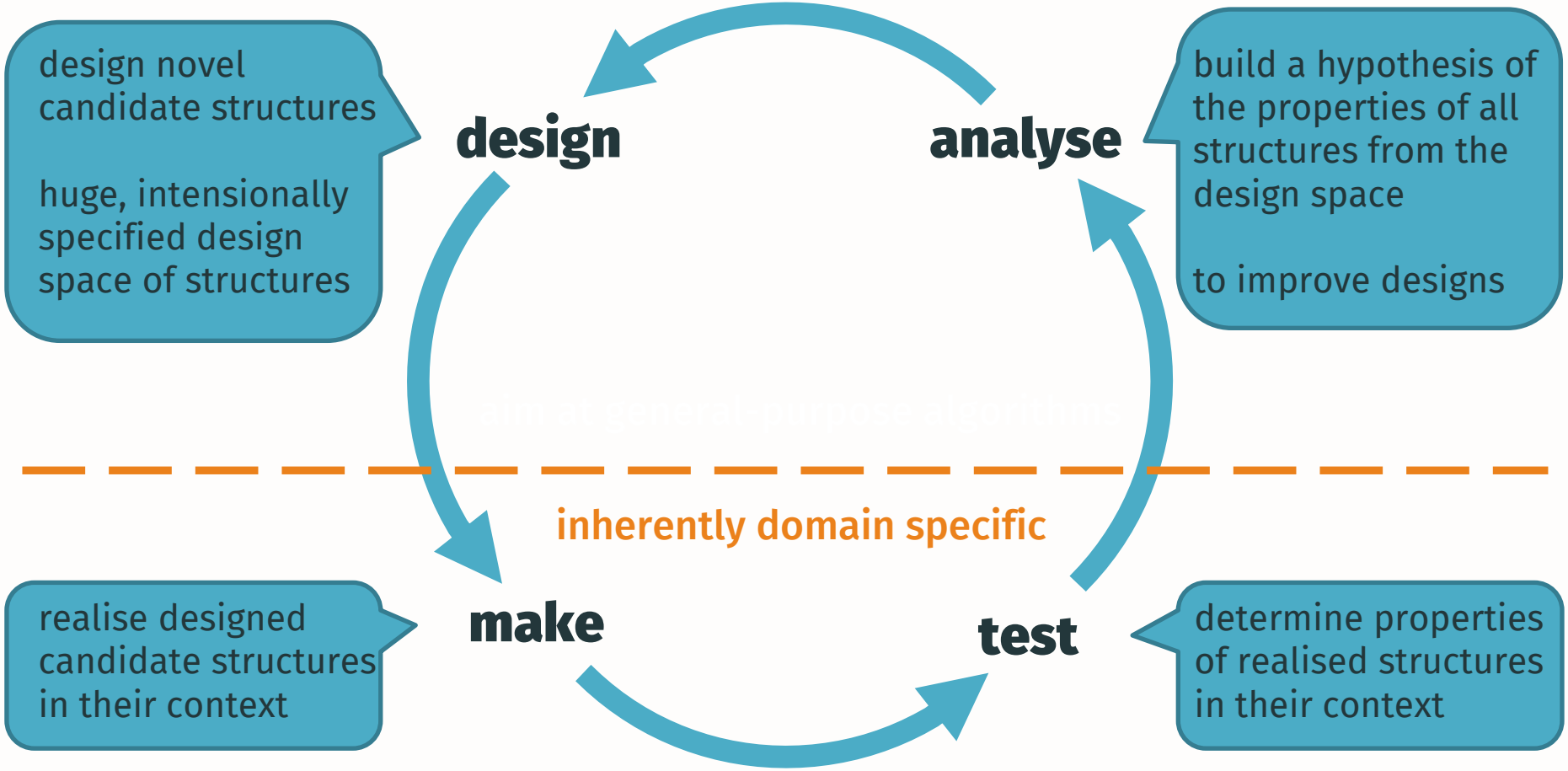
Drug discovery with supervised learning



Cyclic Discovery Processes



Cyclic Discovery Processes



Cyclic Discovery Processes – algorithmic challenges

design novel candidate structures

huge, intensionally specified design space of structures

design

analyse

build a hypothesis of the properties of all structures from the design space

to improve designs

cope with the cyclic nature

cope with the huge candidate space

aim at general-purpose algorithms

interface with domain experts

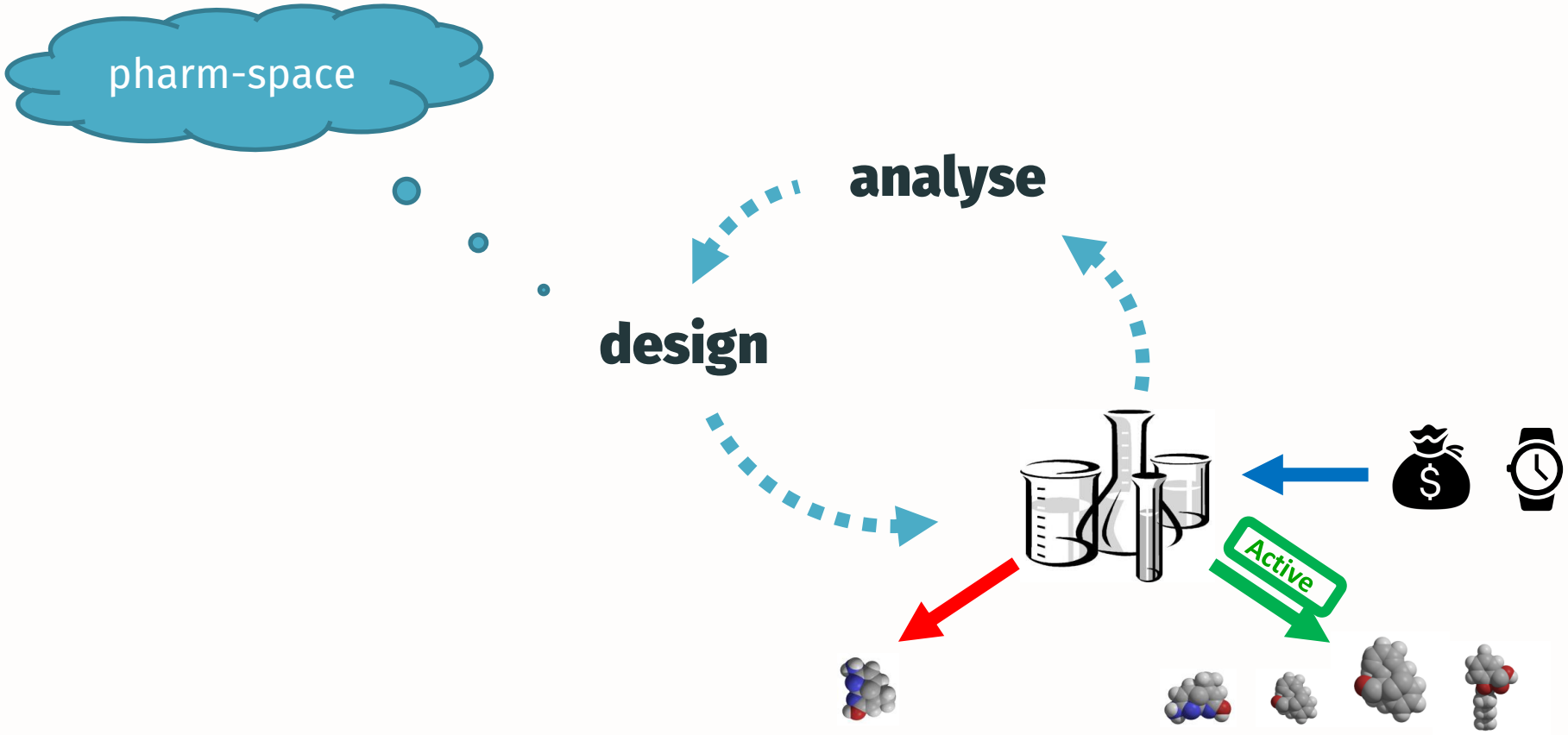
make

test

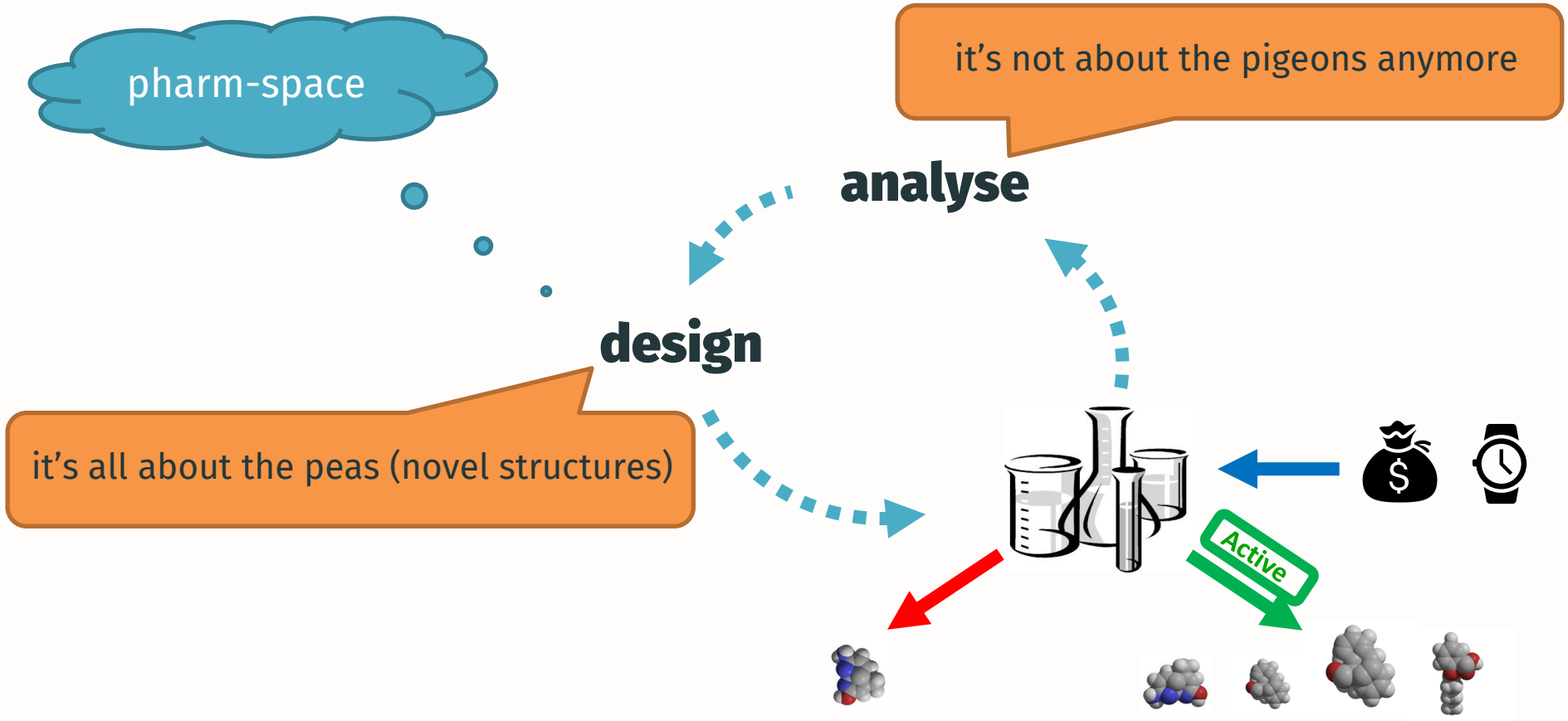
realise designed candidate structures in their context

determine properties of realised structures in their context

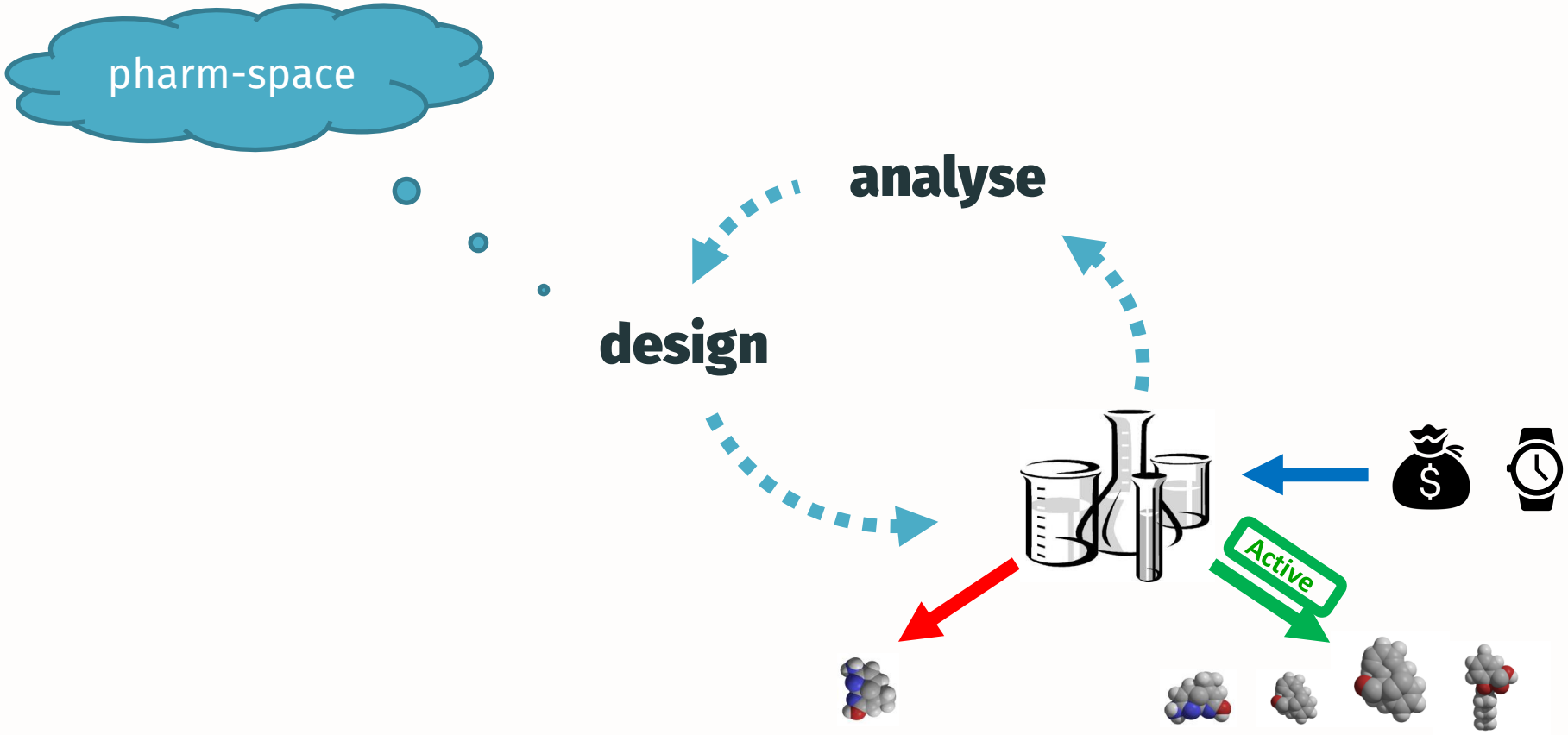
de novo Drug Design



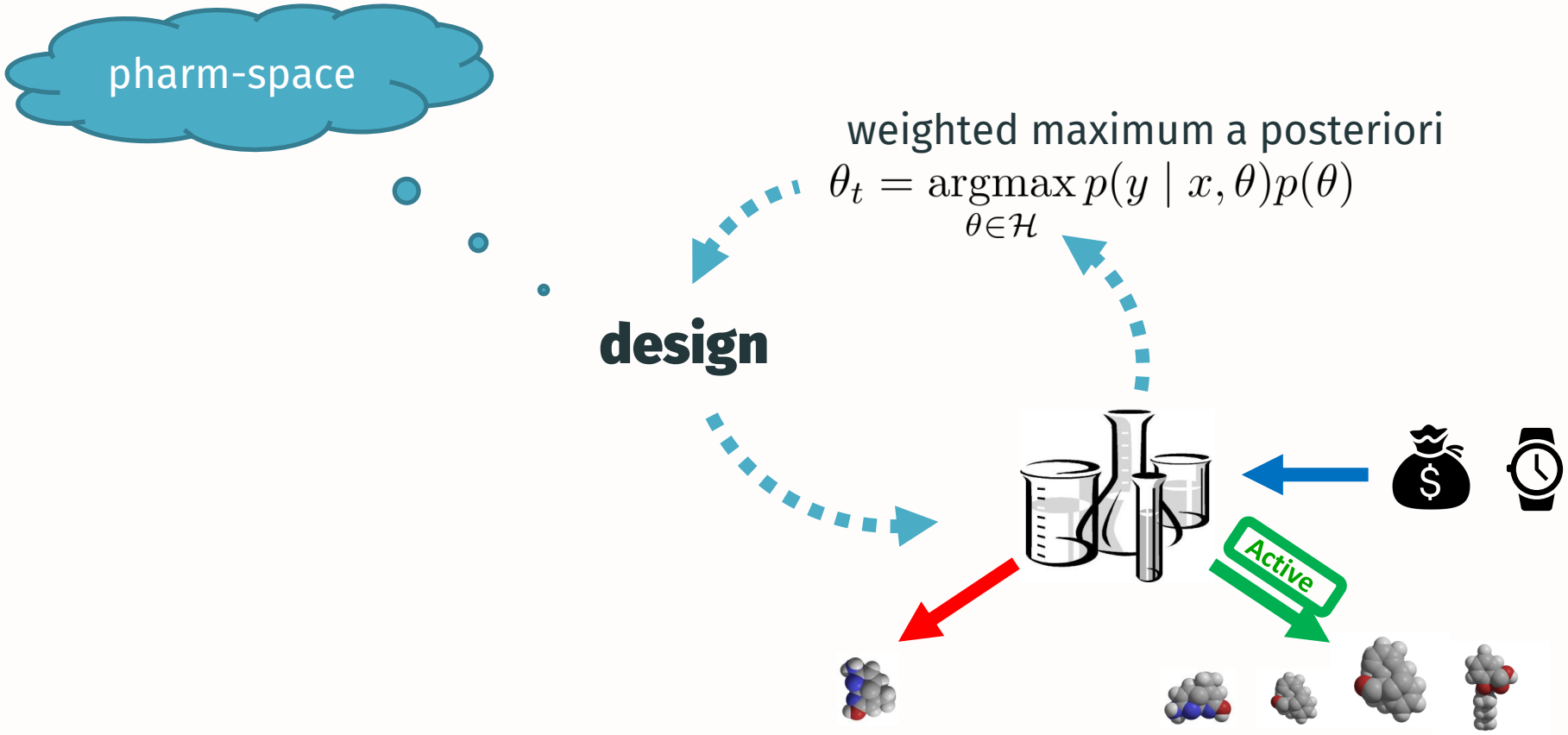
de novo Drug Design



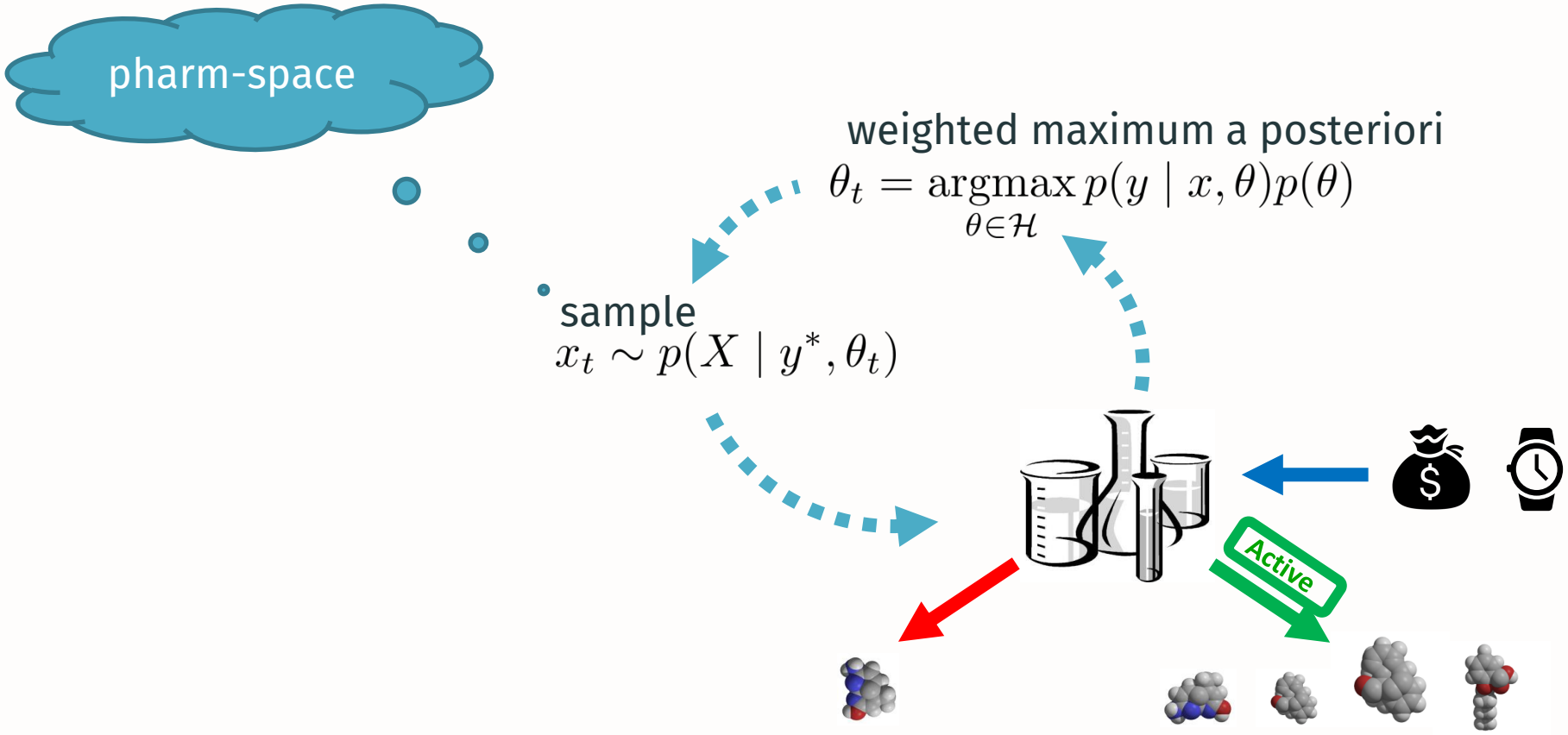
de novo Drug Design



de novo Drug Design



de novo Drug Design



de novo Drug Design

pharm-space

sample

$x \sim Q \quad u \sim \mathcal{U}[0,1]$

$$u < \frac{p(y^* \mid x, \theta)}{p(y^* \mid x_t, \theta)}$$

$$\Downarrow$$

$x_t \leftarrow x$

until converged

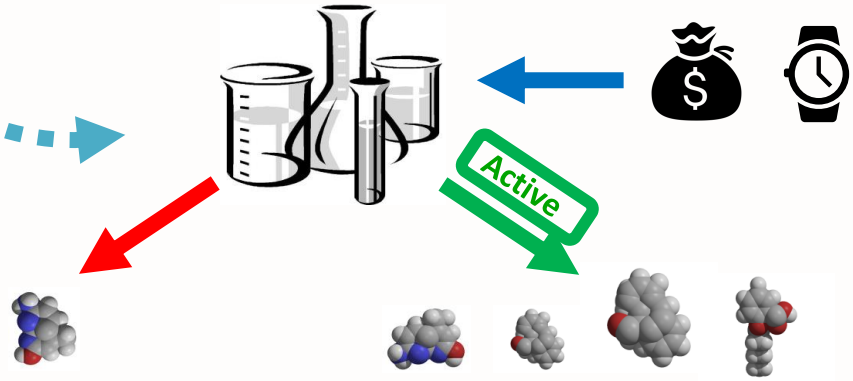
Metropolis-Hastings

sample

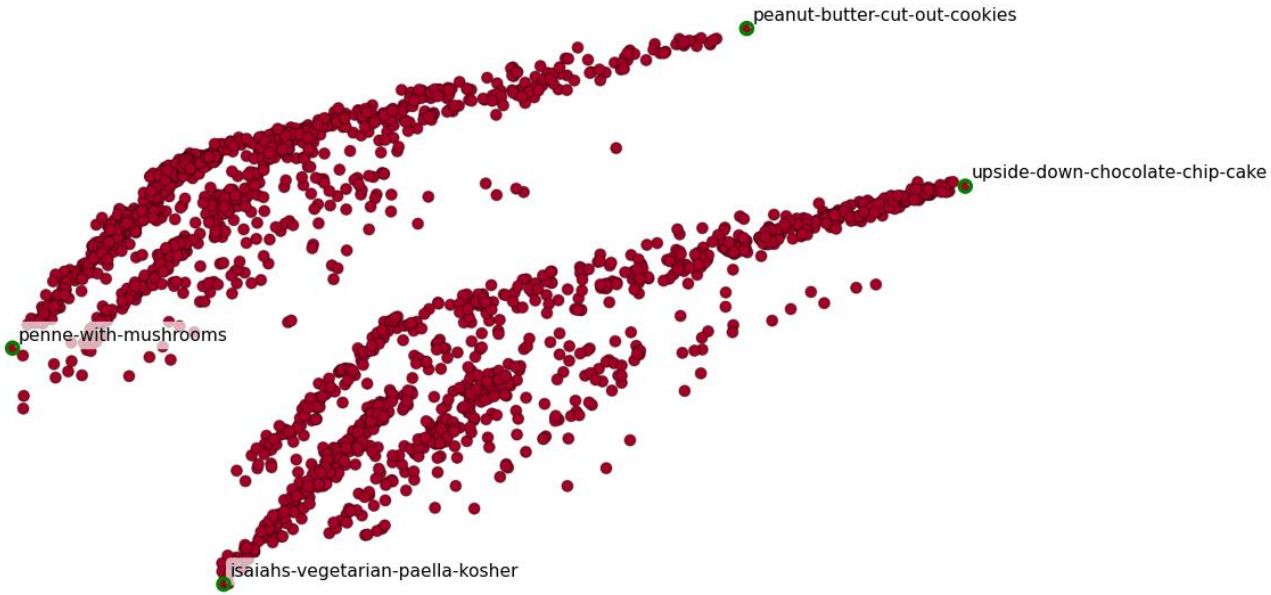
$x_t \sim p(X \mid y^*, \theta_t)$

weighted maximum a posteriori

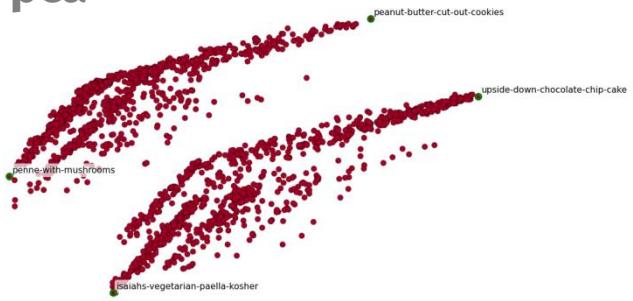
$$\theta_t = \operatorname*{argmax}_{\theta \in \mathcal{H}} p(y \mid x, \theta) p(\theta)$$



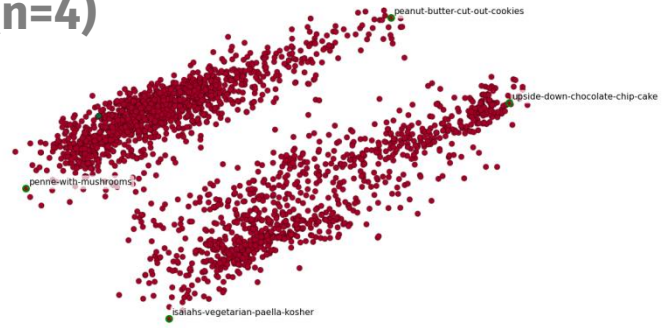
Visual Data Exploration



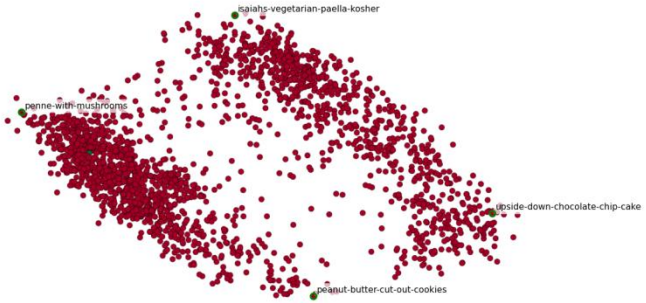
pca



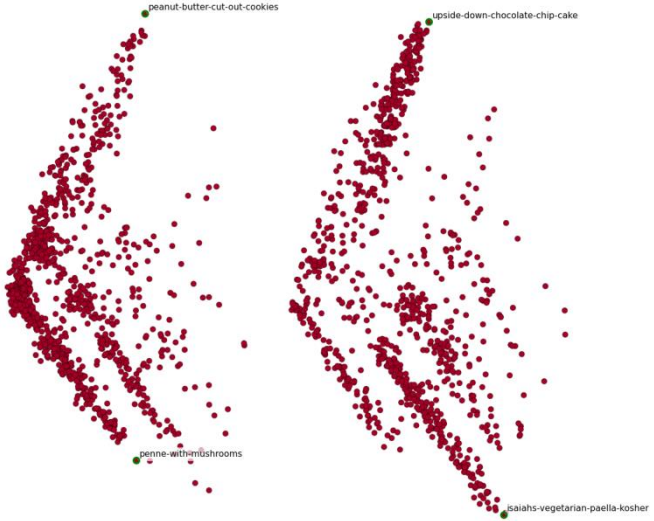
isomap (n=4)

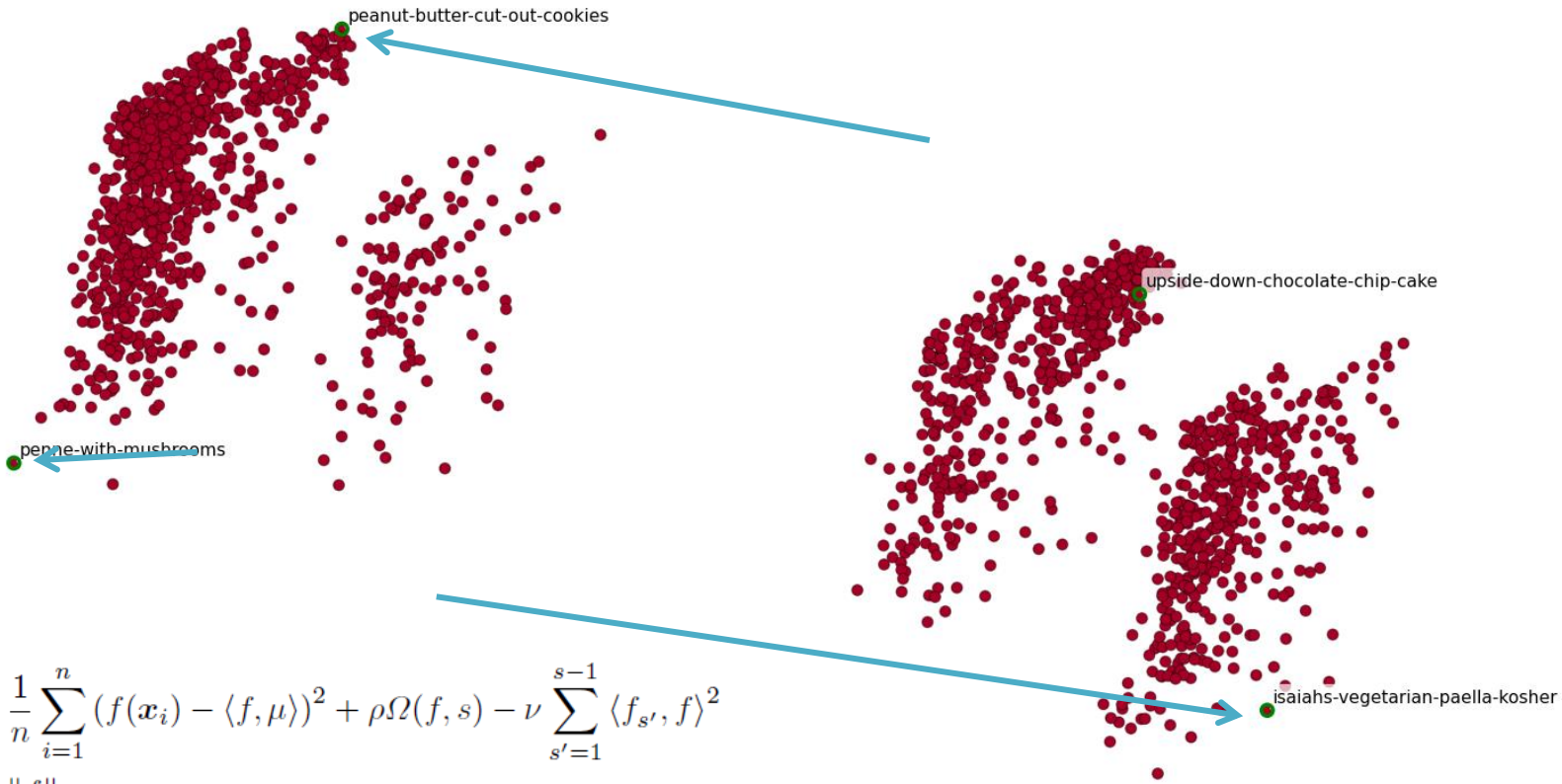


isomap (n=8)



ica





$$
\begin{aligned}
f_s = \underset{f \in \mathcal{H}}{\operatorname{argmax}} \quad & \frac{1}{n}\sum_{i=1}^{n}\left(f(\boldsymbol{x}_i) - \langle f, \mu \rangle\right)^2 + \rho\Omega(f, s) - \nu \sum_{s'=1}^{s-1} \langle f_{s'}, f \rangle^2 \\
\text{subject to} \quad & \|f\|_{\mathcal{H}} = r, \\
& \Upsilon(f, \boldsymbol{y}_s) = 0,
\end{aligned}
$$

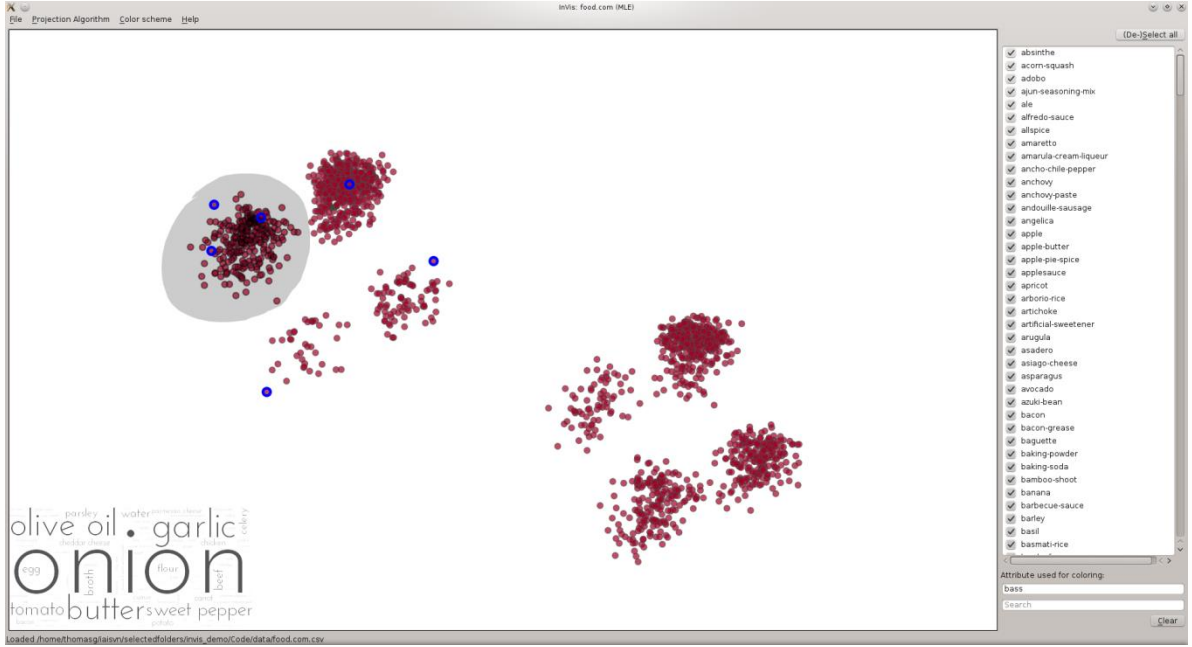
ML/DM algorithms as Tools [ecmlpkdd'13, ecmlpkdd'14]

interactive algorithms
response time & frame rate
sampling & any time

domain-knowledge

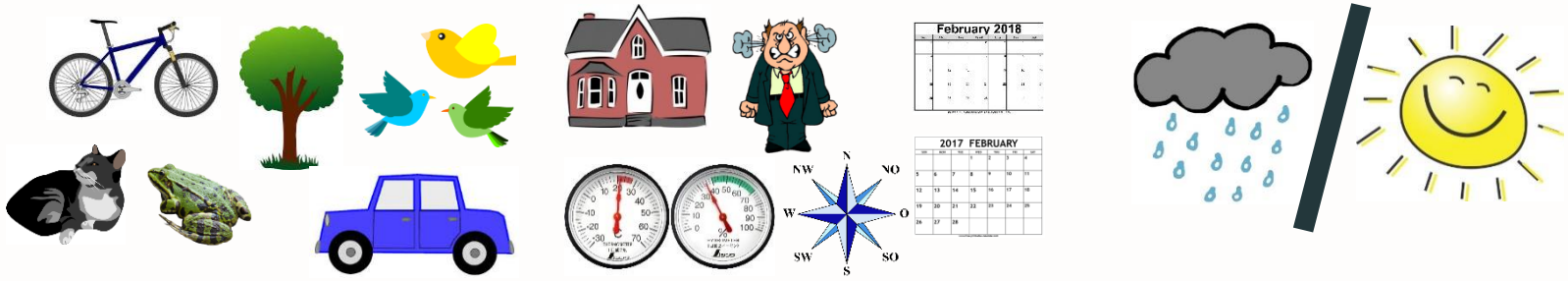
natural vs technical parameters

concept-independent guarantees

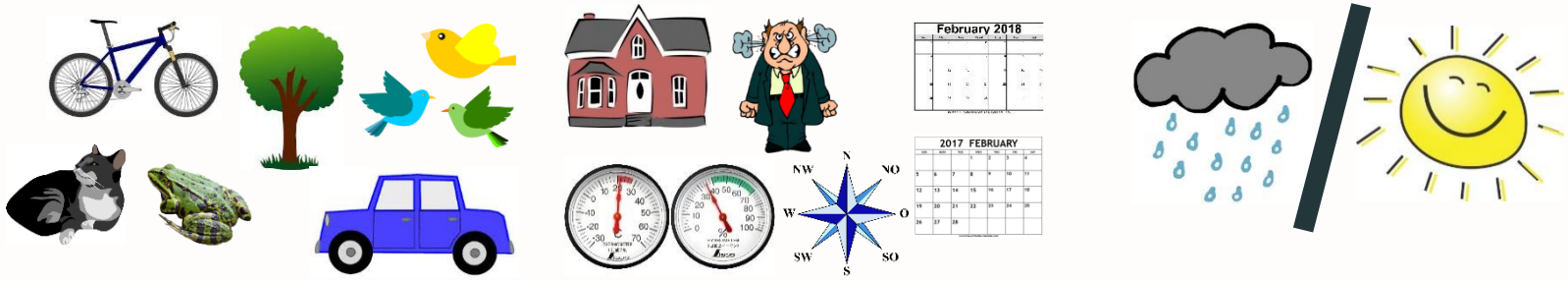


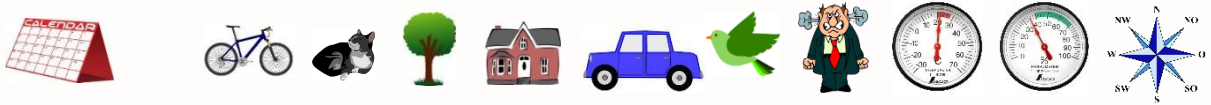
decision trees

Assume we collected a data set ... to predict the weather on our birthday, a week in advance...



Assume we collected a data set ... to predict the weather on our birthday, a week in advance...





thu	2016	✓	✗	✓	✓	✗	✓	✗	14,0	60	N
sat	2017	✗	✓	✓	✓	✓	✗	✗	18,9	40	W
sun	2018	✗	✗	✓	✓	✓	✓	✓	24,1	35	W

Decision Trees form a Hypothesis Class

consist of

- **inner nodes (splitting)**
  - specified by selecting one attribute/feature
  - branch to their children depending on the feature value of an instance
- **outer nodes (decision)**
  - specify the label that we predict for corresponding instances

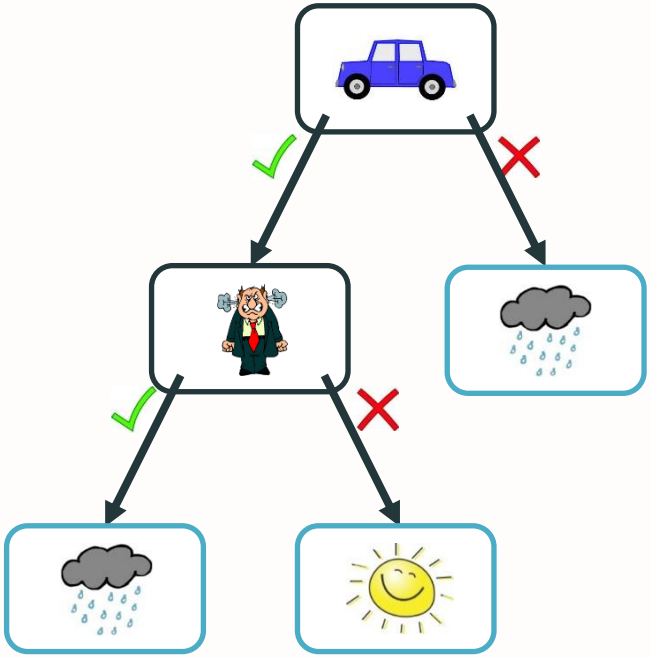
can be used for

- **classification**
- regression
- ranking
- probability

can deal with

- **binary**
- **categorical**
- numerical
- missing
- outliers

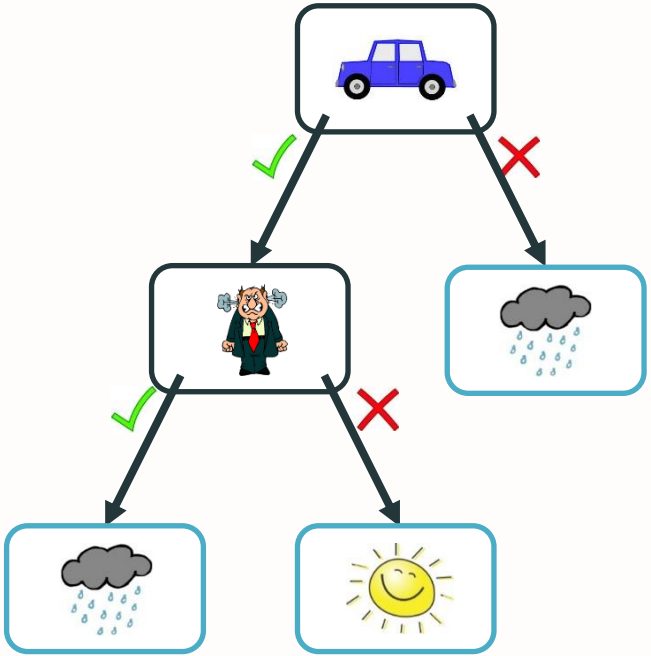
} variables



Decision Trees form a Hypothesis Class

consist of

- **inner nodes (splitting)**
  - specified by selecting one attribute/feature
  - branch to their children depending on the feature value of an instance
- **outer nodes (decision)**
  - specify the label that we predict for corresponding instances



can be used for

- **classification**
- regression
- ranking
- probability

can deal with

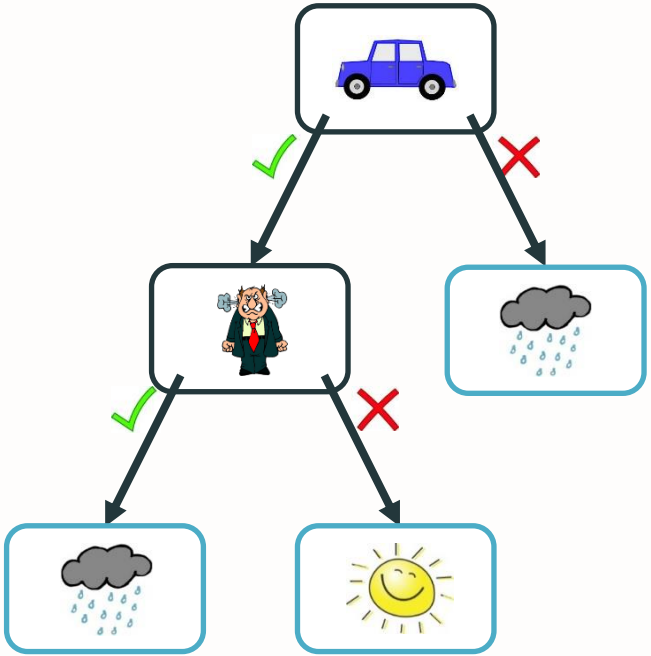
- **binary**
- **categorical**
- numerical
- missing
- outliers

} variables

Decision Trees form a Hypothesis Class

consist of

- **inner nodes (splitting)**
  - specified by selecting one attribute/feature
  - branch to their children depending on the feature value of an instance
- **outer nodes (decision)**
  - specify the label that we predict for corresponding instances



can be used for

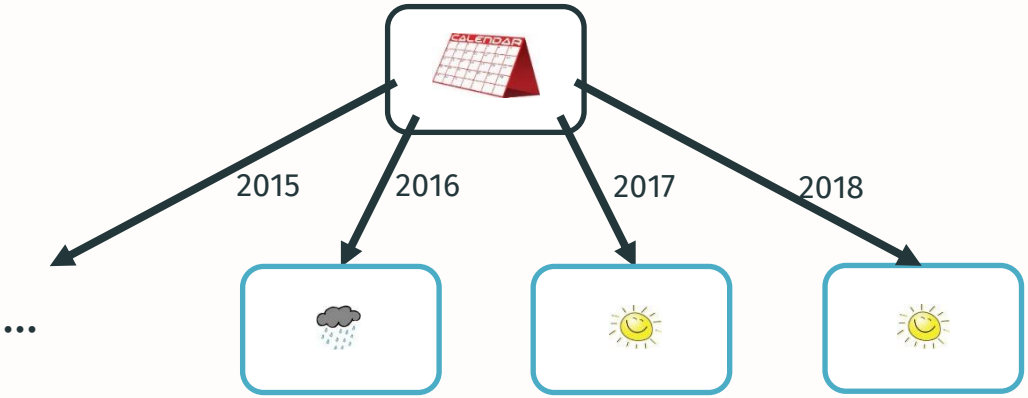
- **classification**
- regression
- ranking
- probability

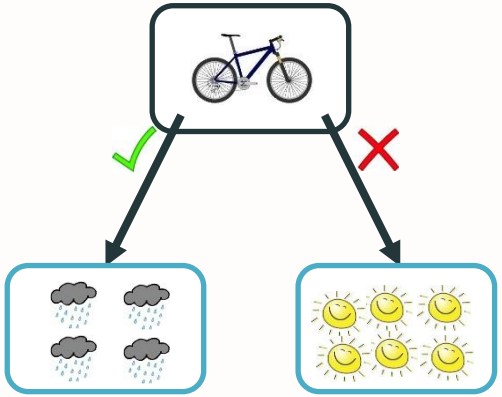
can deal with

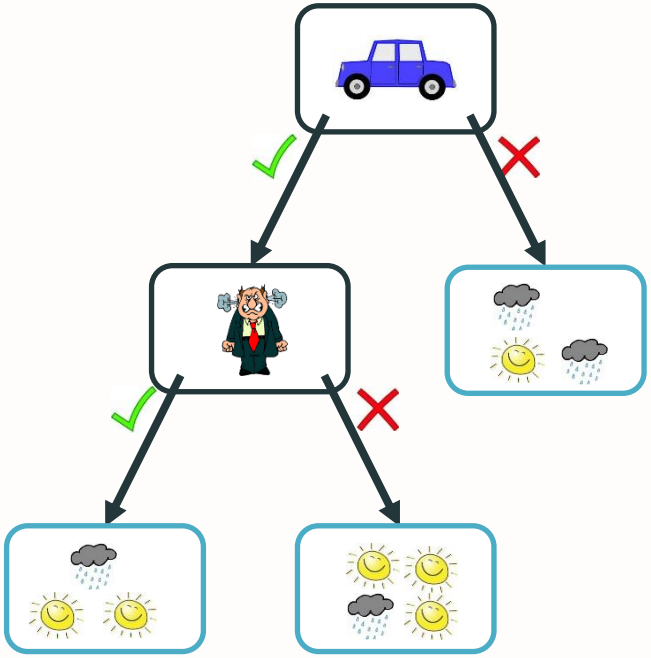
- **binary**
- **categorical**
- numerical
- missing
- outliers

} variables

Decision Trees form a Hypothesis Class



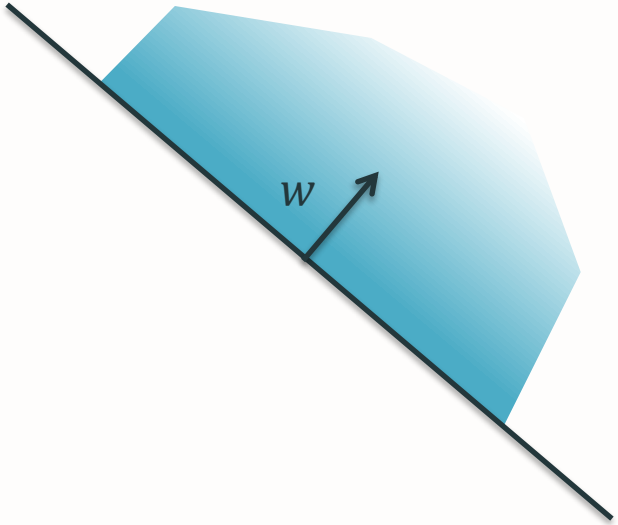


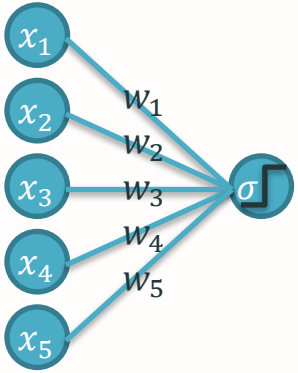


What is a good or even optimal decision tree?
Can we always find a good or even an optimal one?
How long does it take to find a good or even an optimal one?

neural networks

Perceptron

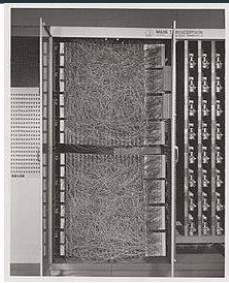




$$f(x) = \sigma(\langle x, w \rangle)$$

$$= \operatorname{sign}\left(\sum_{i=1}^{d} x_i w_i\right)$$

Perceptron ($x^{(t)} \in \mathbb{R}^d, y_t \in \{-1, +1\}$ for $t = 1, 2, \ldots$)



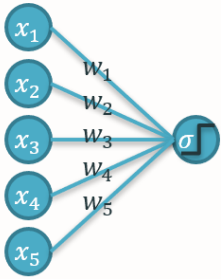
traditionally used in *online learning*

expressivity
thresholded linear hypotheses (**halfspaces**)
simple Boolean functions (**and**, **or**, ...)

learning theoretic bounds
- data dimension based
- radius-margin bound

downsides
- linear hypotheses not very expressive
- no xor/parity type data

fix
- kernel perceptron
- multi-layer perceptron



Rosenblatt, Frank (1957)
The Perceptron--a perceiving and recognizing automaton.
Report 85-460-1, Cornell Aeronautical Laboratory

Perceptron ($x^{(t)} \in \mathbb{R}^d, y_t \in \{-1,+1\}$ for $t = 1, 2, \ldots$)

traditionally used in *online learning*

expressivity
thresholded linear hypotheses (**halfspaces**)
simple Boolean functions (**and**, **or**, ...)

learning theoretic bounds
- data dimension based
- radius-margin bound

downsides
- linear hypotheses not very expressive
- no xor/parity type data

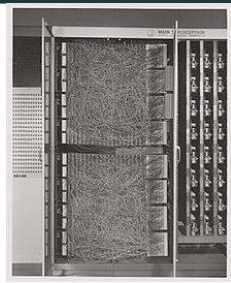
fix
- kernel perceptron
- multi-layer perceptron

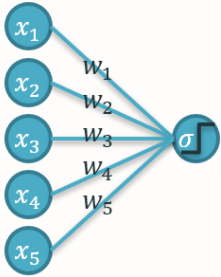
$$f_t(x) = \sigma\left[\langle x^{(t)}, w^{(t)} \rangle\right]$$

$$w^{(t+1)} = w^{(t)} + \begin{cases} 0 & \text{if } f_t\left(x^{(t)}\right) = y_t \\ y_t x^{(t)} & \text{otherwise} \end{cases}$$

therefore, on mistake
$$\langle x^{(t)}, w^{(t+1)} \rangle = \langle x, w^{(t)} \rangle + y_t \left\| x^{(t)} \right\|$$

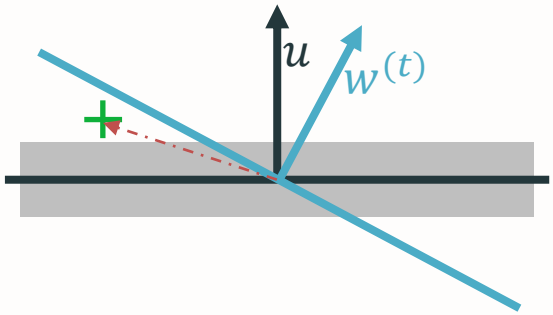
recall
σ: sign function
$\langle \cdot, \cdot \rangle$: inner product



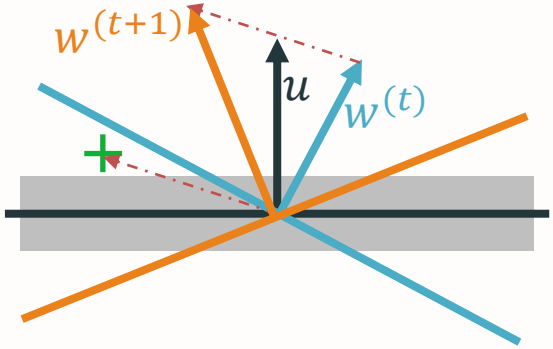


Rosenblatt, Frank (1957)
The Perceptron--a perceiving and recognizing automaton.
Report 85-460-1, Cornell Aeronautical Laboratory

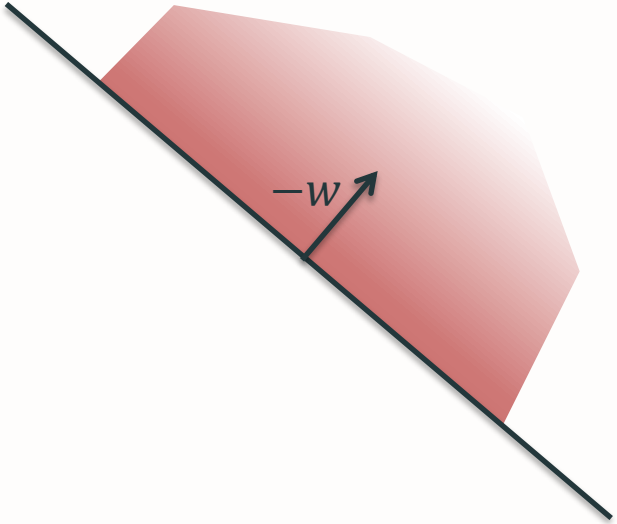
Update Illustration

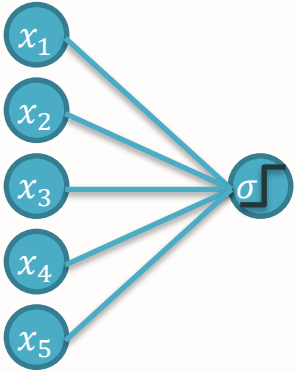


Update Illustration

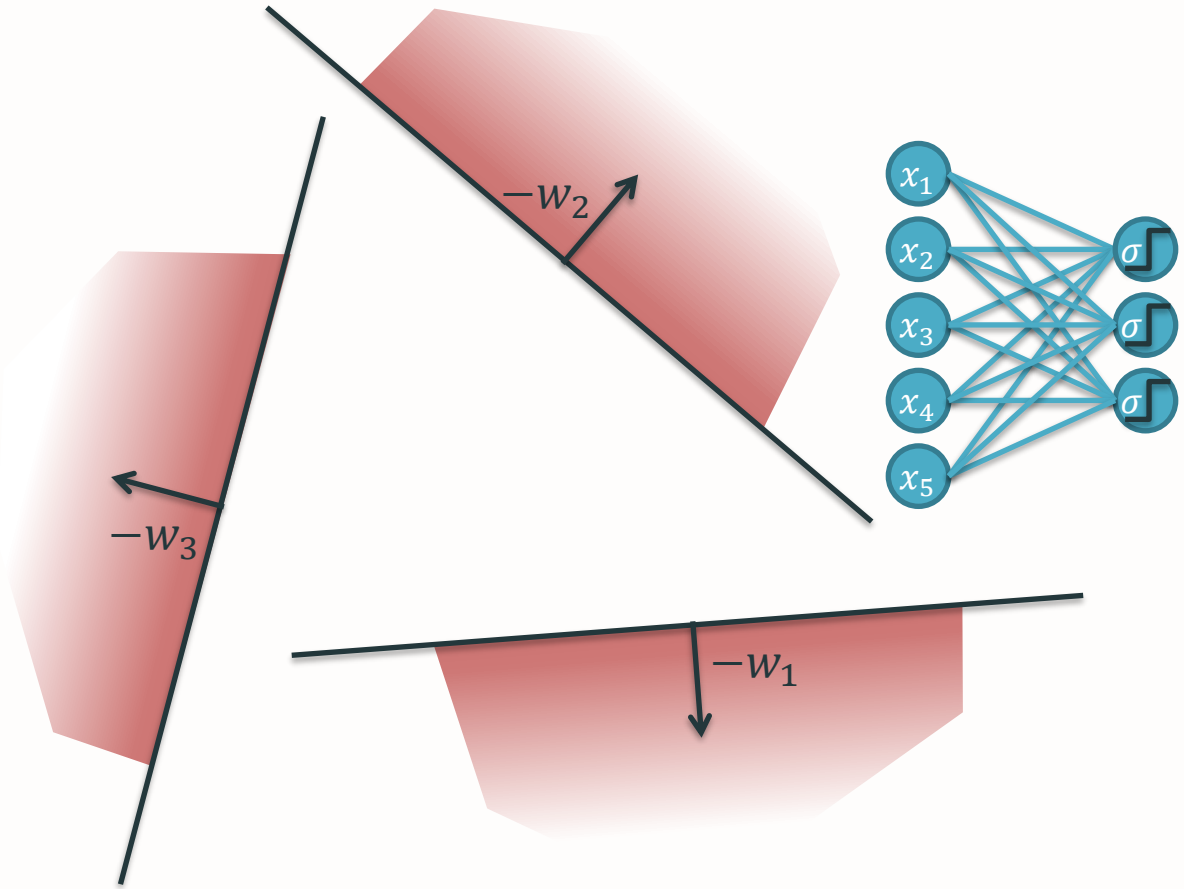


Feedforward Networks: Multi-Layer Perceptron & Deep Neural Nets

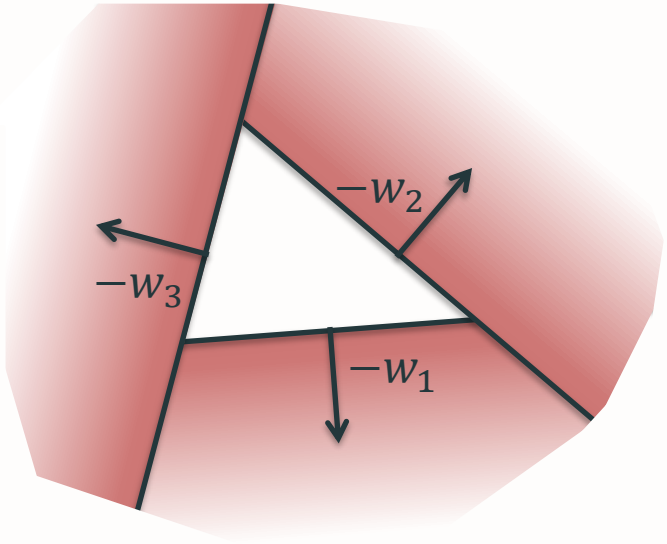


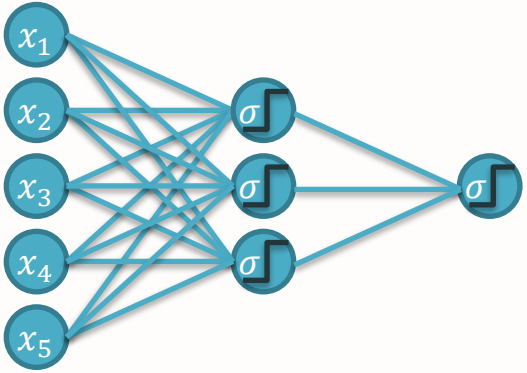


Feedforward Networks: Multi-Layer Perceptron & Deep Neural Nets

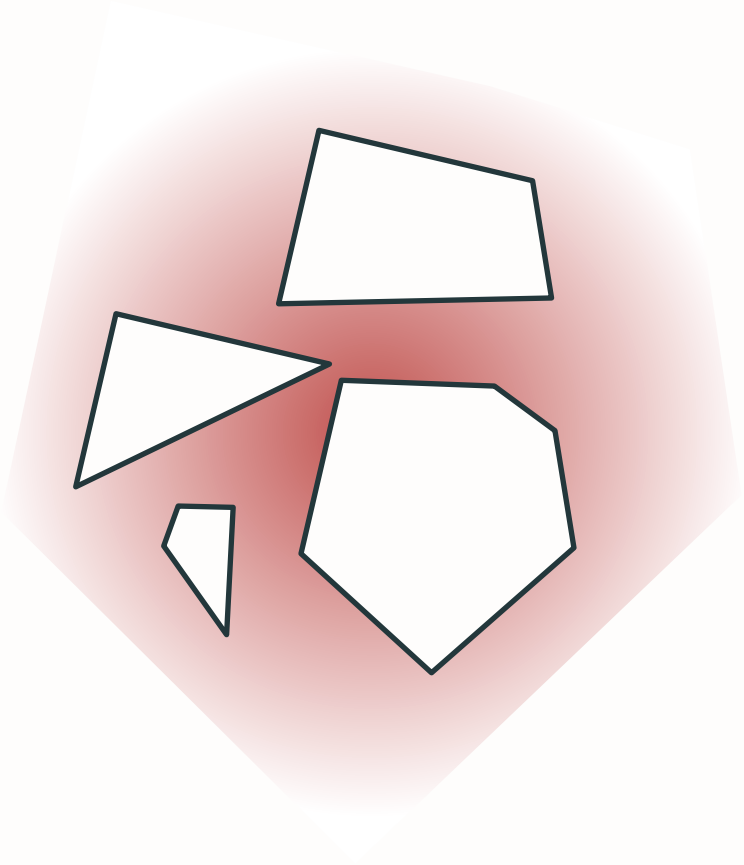


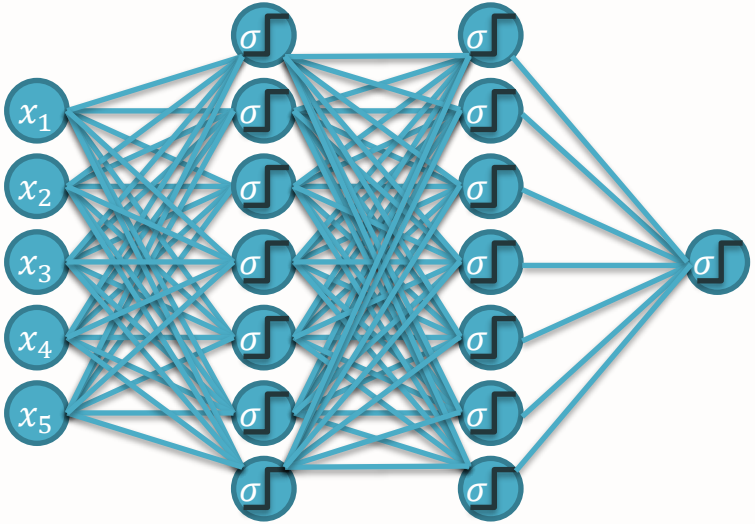
Feedforward Networks: Multi-Layer Perceptron & Deep Neural Nets



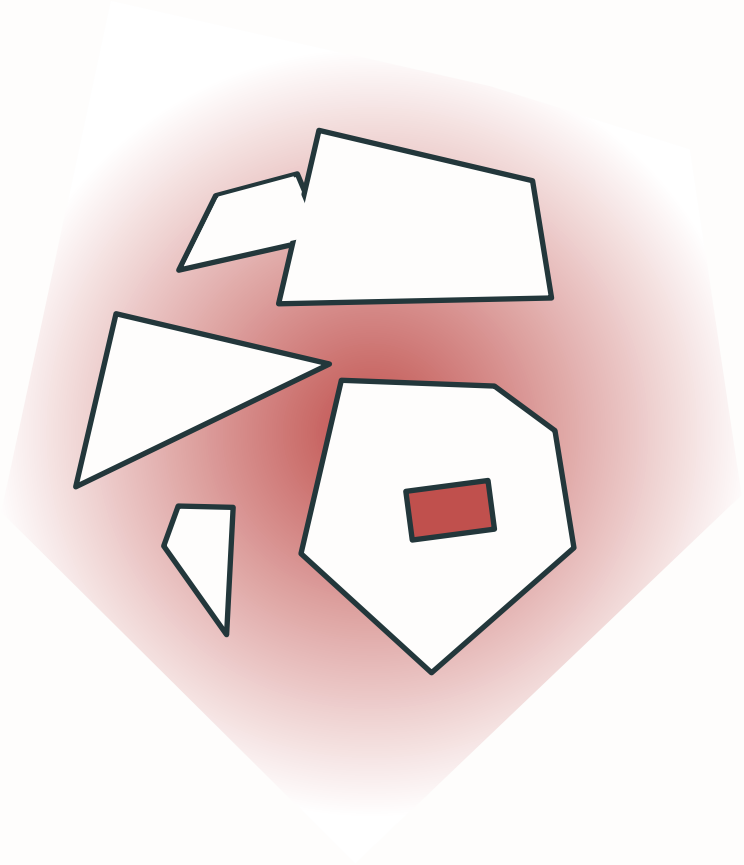


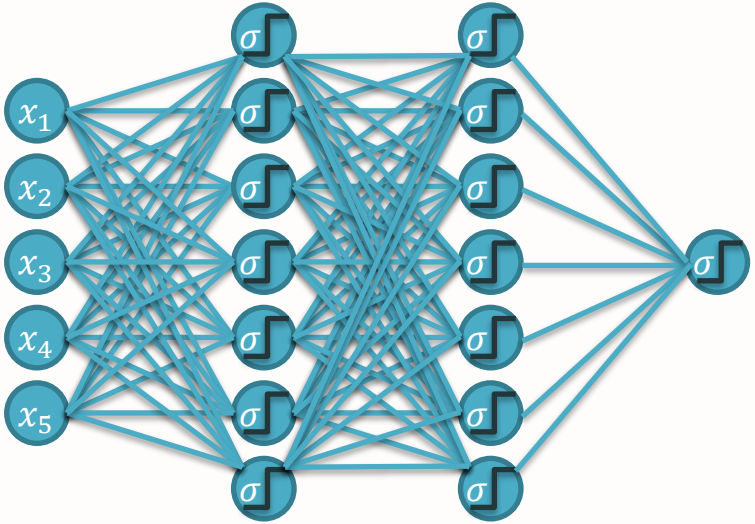
Feedforward Networks: Multi-Layer Perceptron & Deep Neural Nets



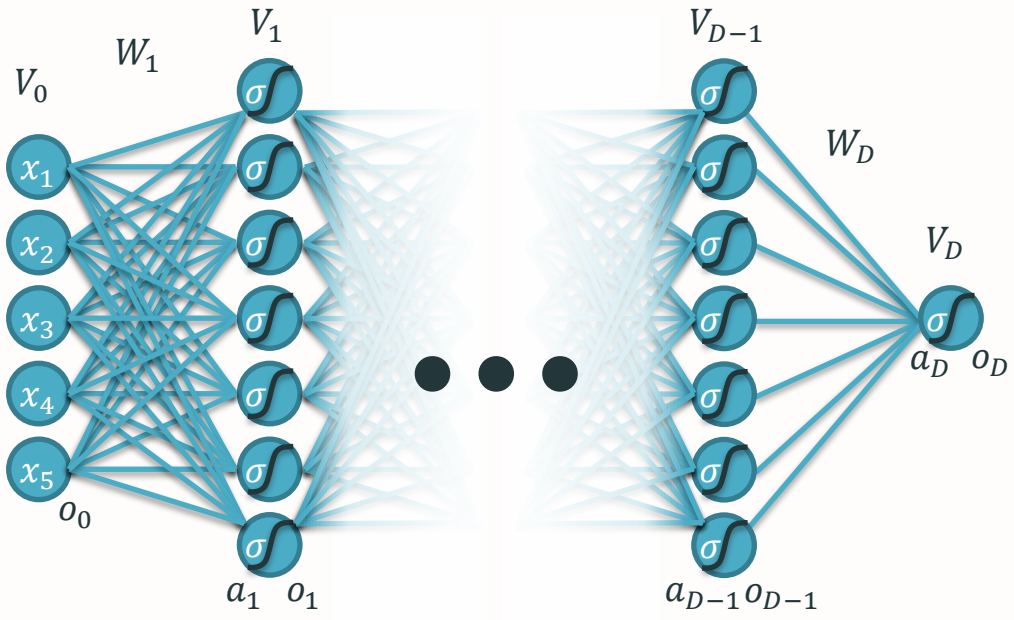


Feedforward Networks: Multi-Layer Perceptron & Deep Neural Nets





Feedforward Networks: Multi-Layer Perceptron & Deep Neural Nets



$$o_0 = x \qquad a_d = W_d o_{d-1} \qquad o_d = \sigma(a_d)$$

Feedforward Networks: Multi-Layer Perceptron & Deep Neural Nets

multiple, “stacked” “layers” of perceptrons

input layer
“hidden” feature-generating layer(s)
output layer

expressivity:

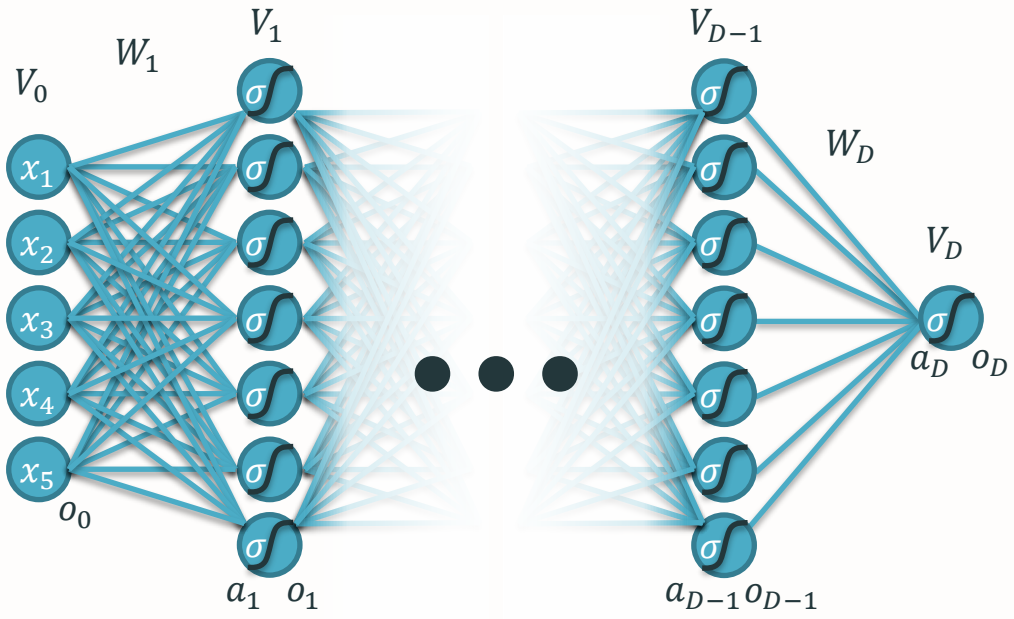
- universal even with just one hidden layer
- any Boolean function

downsides

- expressivity requires exponential size
- training becomes computationally hard

fix

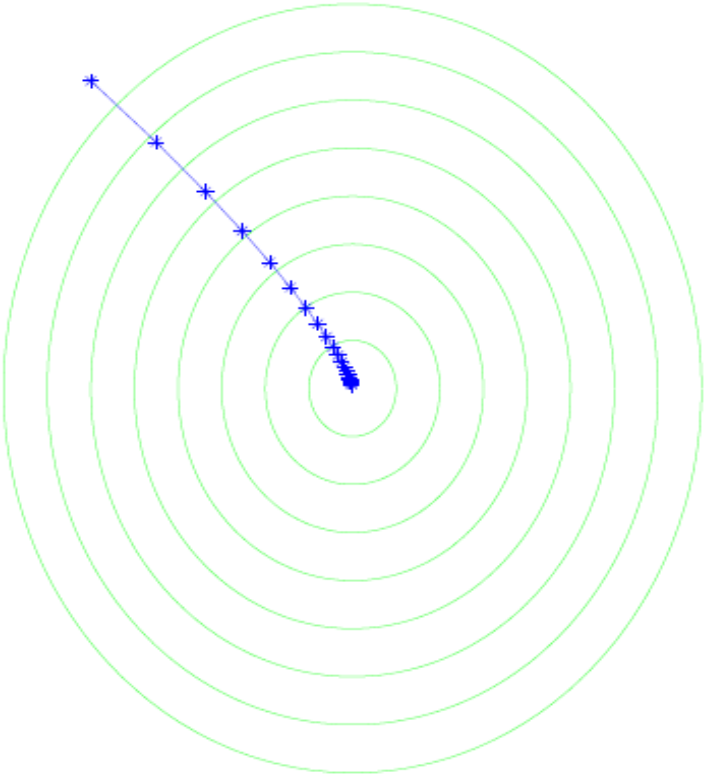
- heuristics work empirically very well



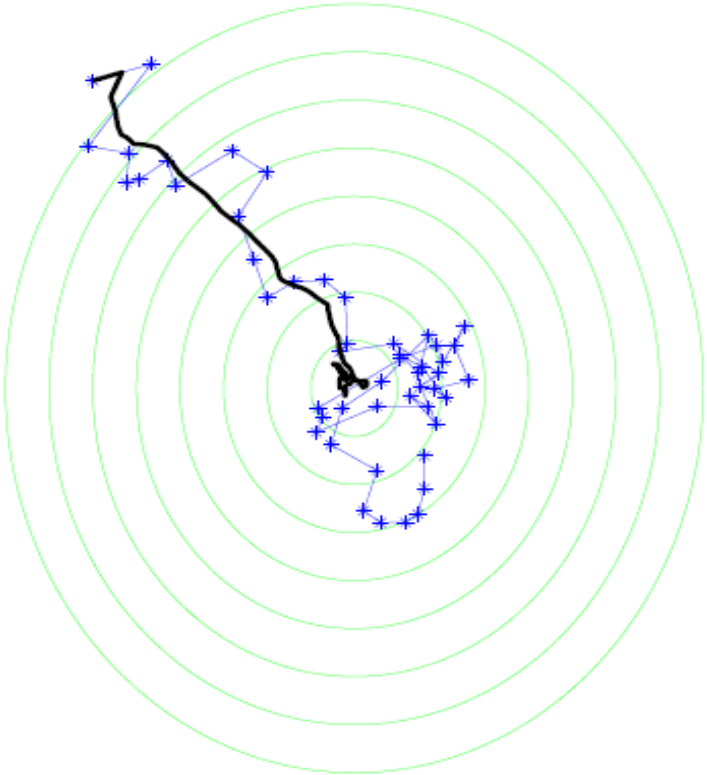
$$o_0 = x \qquad a_d = W_d o_{d-1} \qquad o_d = \sigma(a_d)$$

Finding Minima: Waterflow

Gradient Descent (DG) vs Stochastic Gradient Descent (SGD)



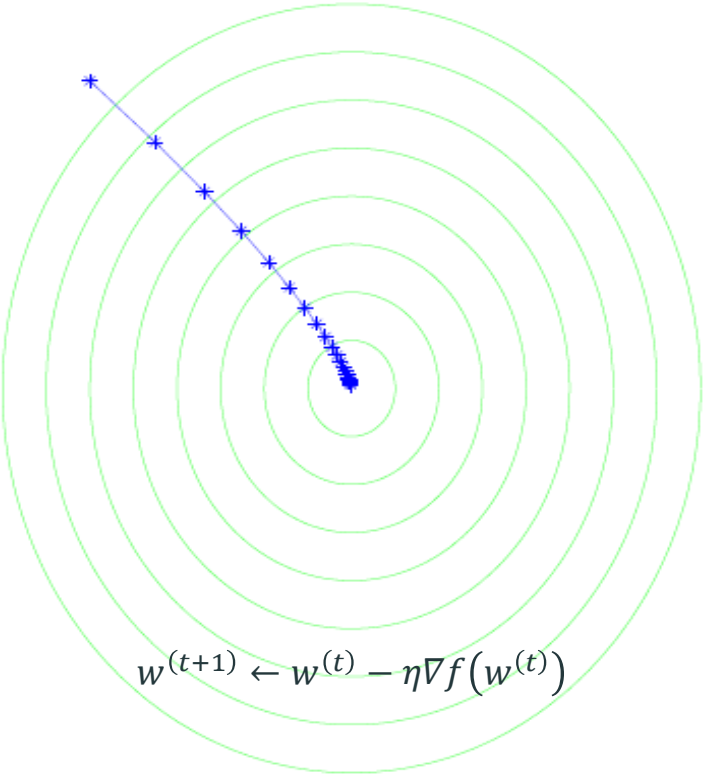
GD: deterministic updates
in the direction of the gradient



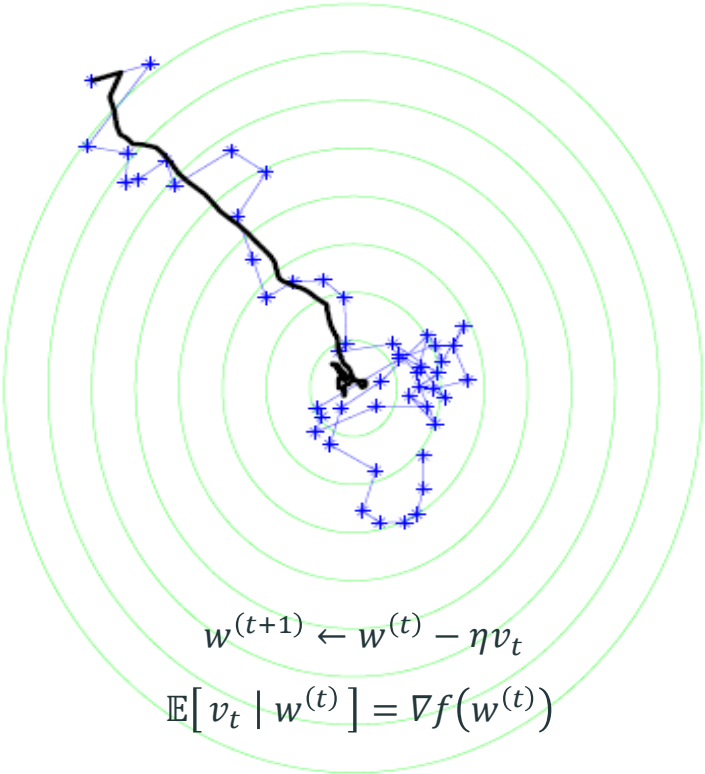
SGD: random updates whose expectation is
in the direction of the gradient

source: Understanding Machine Learning by Shai Ben-David and Shai Shalev-Shwartz

Gradient Descent (DG) vs Stochastic Gradient Descent (SGD)



GD: deterministic updates
in the direction of the gradient

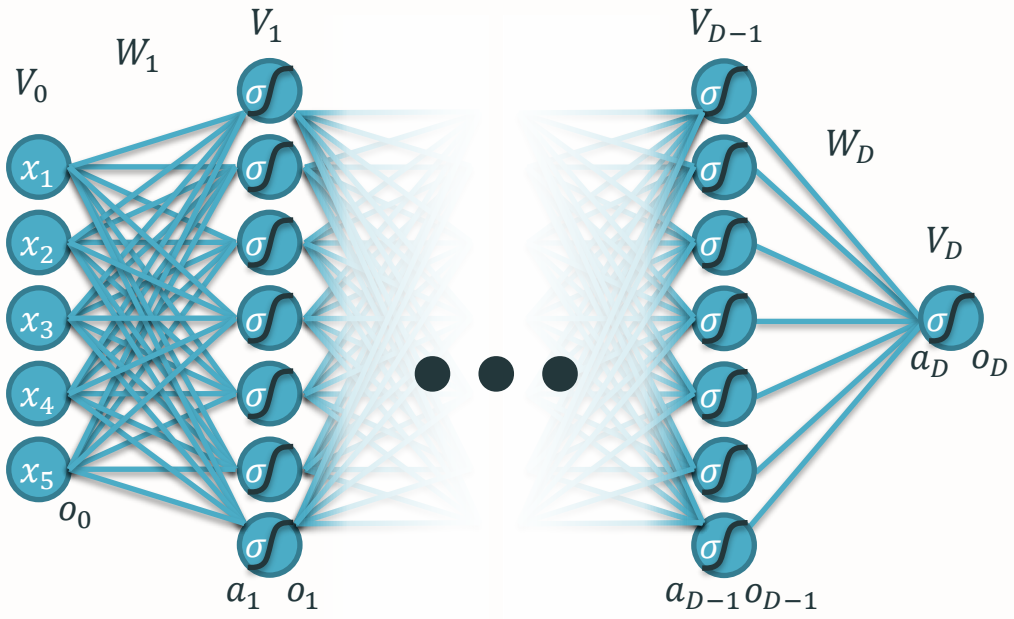


SGD: random updates whose expectation is
in the direction of the gradient

source: Understanding Machine Learning by Shai Ben-David and Shai Shalev-Shwartz

Backpropagation with Stochastic Gradient Descent (SGD)

```
for t = 1, ... T
  | sample x, y
  | δ ← propagate(x, y, w^(t), σ, σ')
  | for all edges (v_{d-1,j}, v_{d,i})
  |   | set u_{d,j,i} ← δ_{d,i} σ'(a_{d,i}) o_{d-1,j}
  | update w^(t+1) ← w^(t) − η_t (u + λ w^(t))
output best w^(t) wrt validation set
```



$o_0 \leftarrow x$ $\quad a_d \leftarrow W_d o_{d-1}$ $\quad o_d \leftarrow \sigma(a_d)$

$\delta_d \leftarrow W_d[\delta_{d+1} \otimes \sigma'(a_{d+1})]$ $\quad \delta_D \leftarrow o_D - y$

Stochastic Gradients for Neural Networks

We want to minimise $\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\ell(f_w(x), y)]$

With $(x, y) \sim \mathcal{D}$ we define a random variable $v_t = [\nabla_w \, \ell(f_w(x), y)] \left(w^{(t)}\right)$ such that

$$\mathbb{E}\left[v_t \mid w^{(t)}\right] = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\nabla_w \, \ell(f_w(x), y)] = \nabla_w \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\ell(f_w(x), y)]$$

Now

$$\frac{\partial L}{\partial w}$$

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial o_D} \cdot \frac{\partial o_D}{\partial w}$$

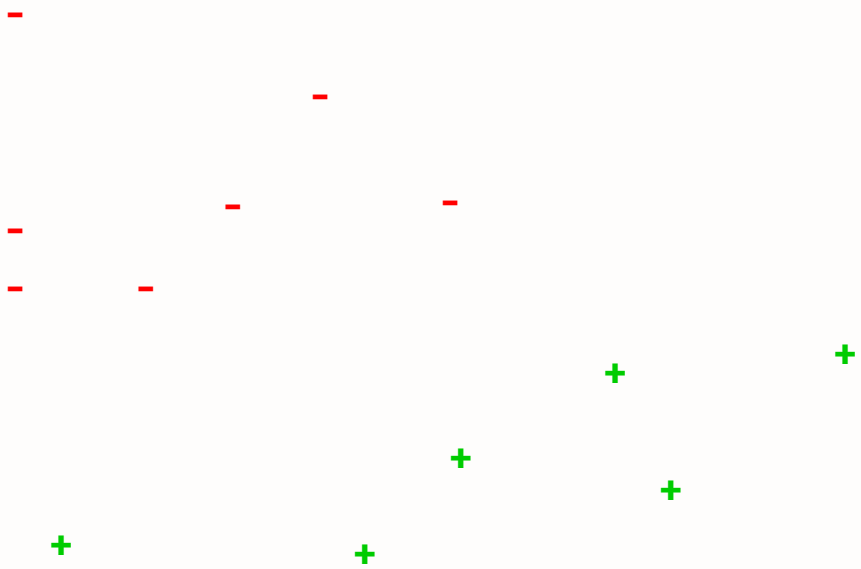
$$\frac{\partial o_D}{\partial w} = \frac{\partial o_D}{\partial a_D} \cdot \frac{\partial a_D}{\partial w}$$

$$\frac{\partial a_D}{\partial w} = \frac{\partial a_D}{\partial o_{D-1}} \cdot \frac{\partial o_{D-1}}{\partial w}$$

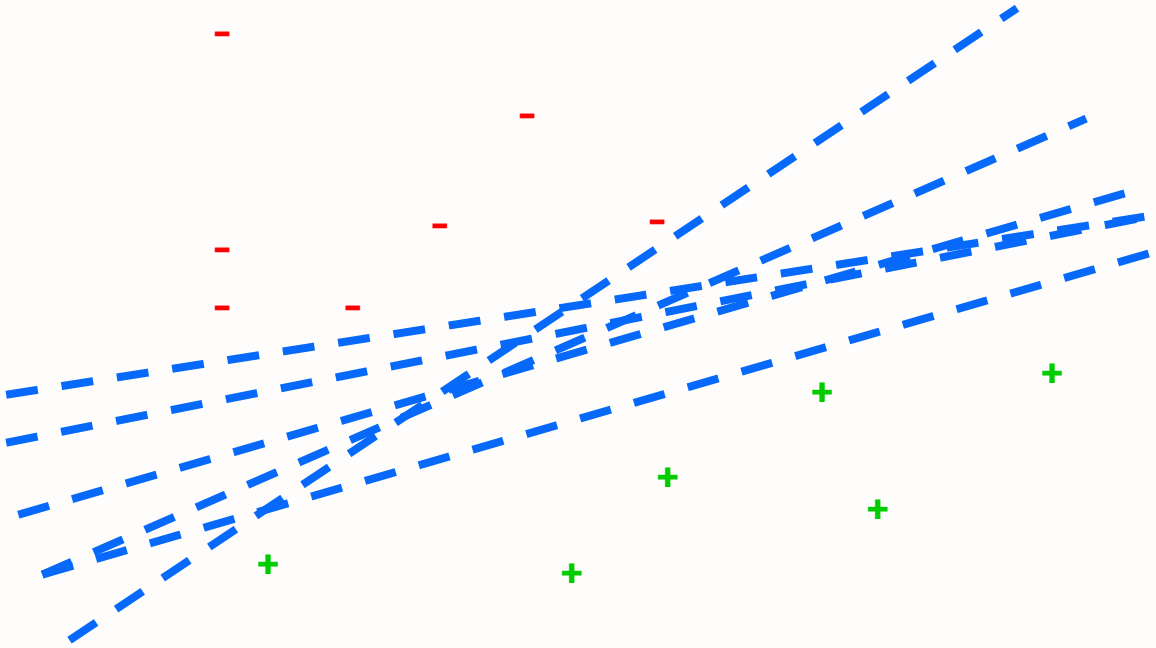
$$\frac{\partial o_{D-1}}{\partial w} = \cdots$$

Support Vector Machines

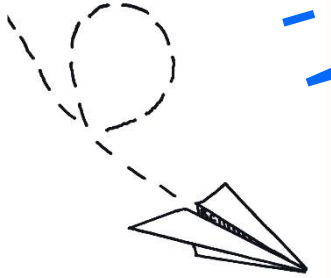
The Magic that isn’t that Magical...

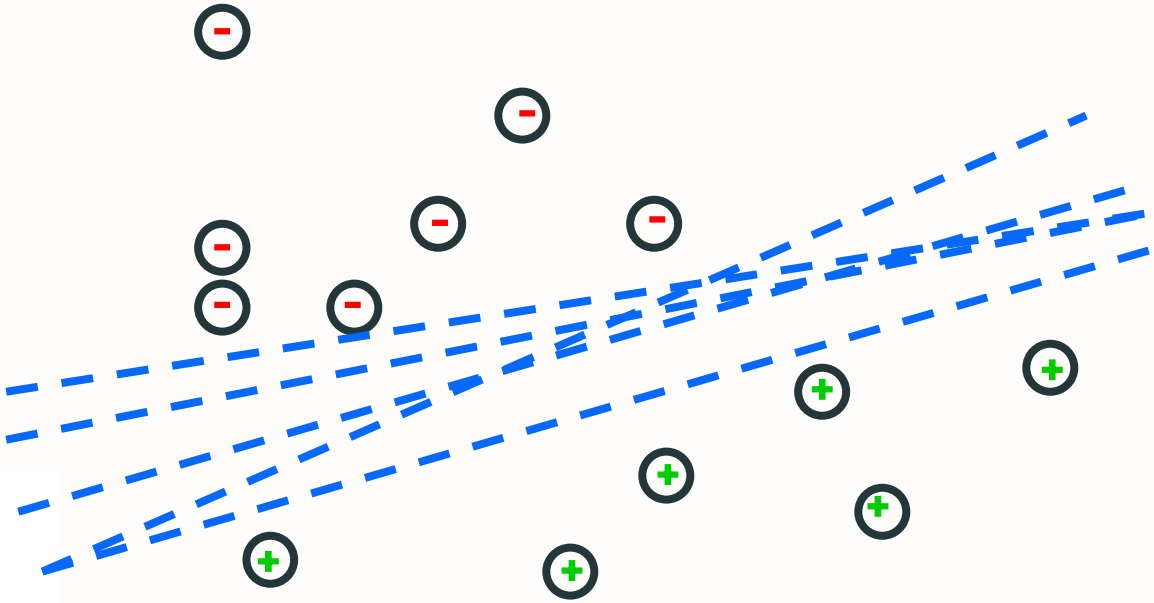


Classification by Linear Functions

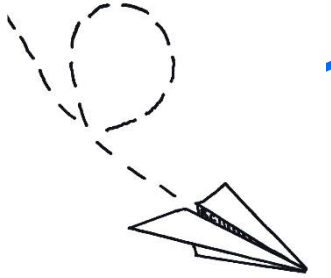


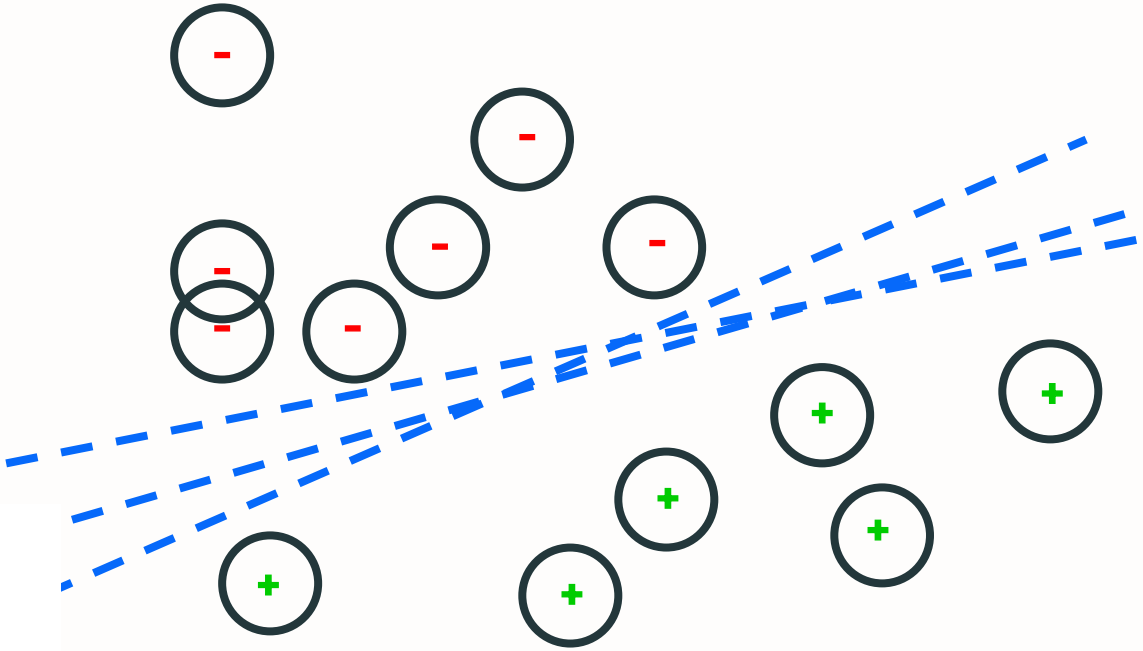
Motivation: Support Vector Machines



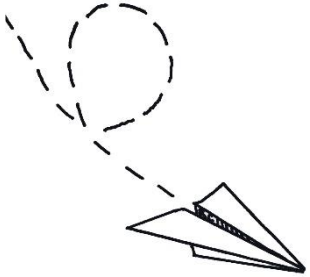


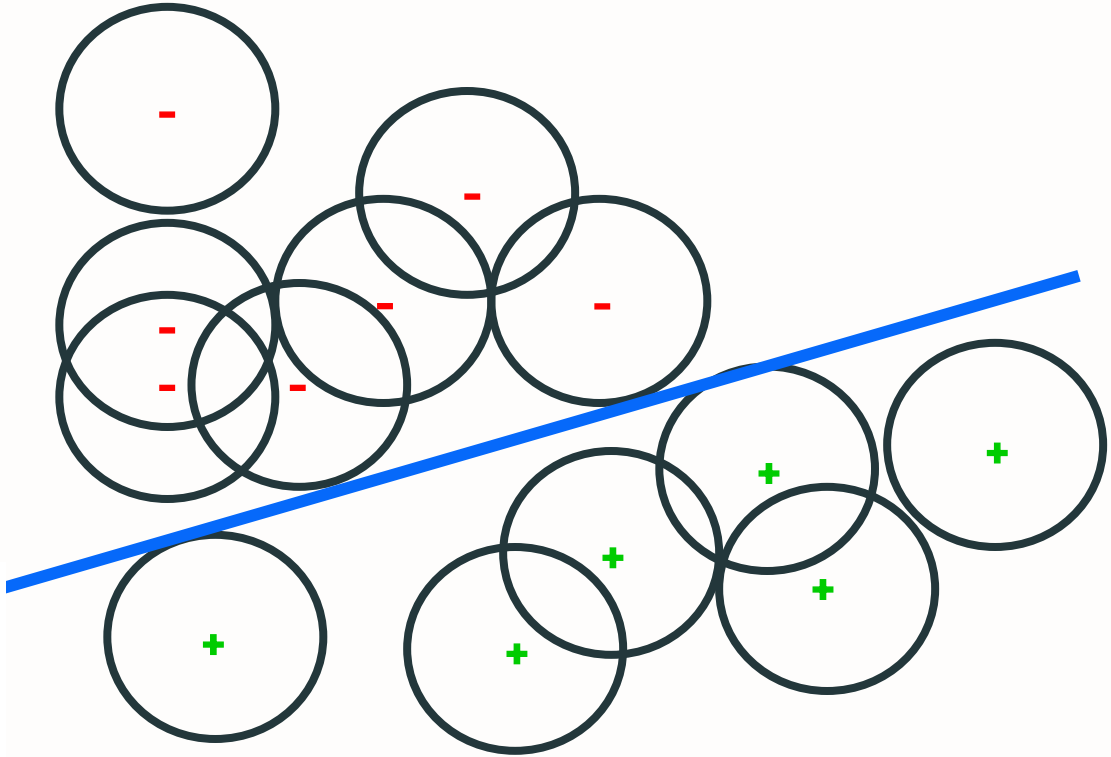
Motivation: Support Vector Machines



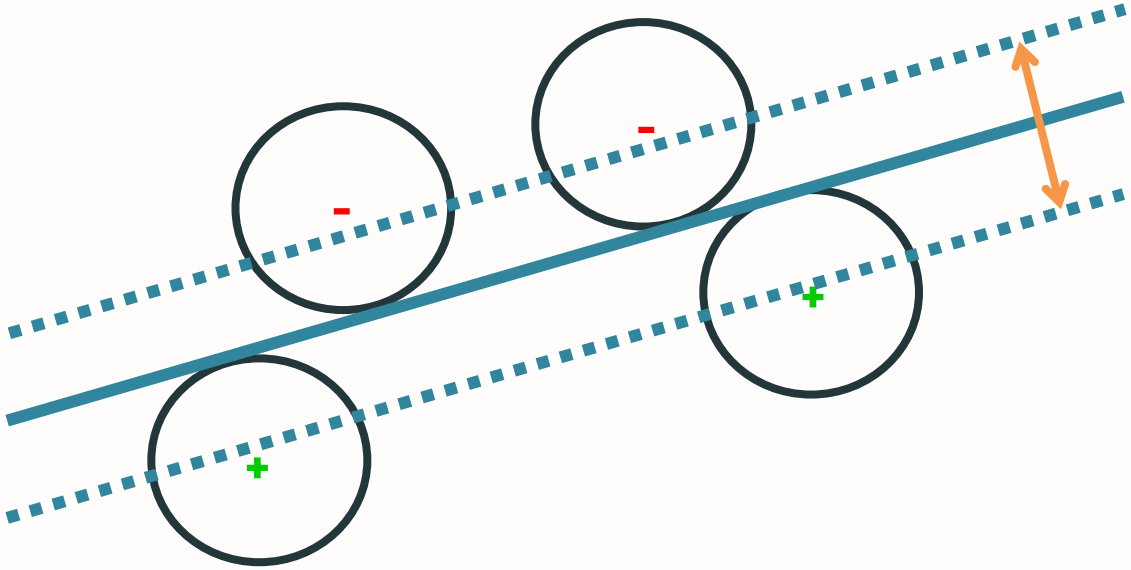


Motivation: Support Vector Machines

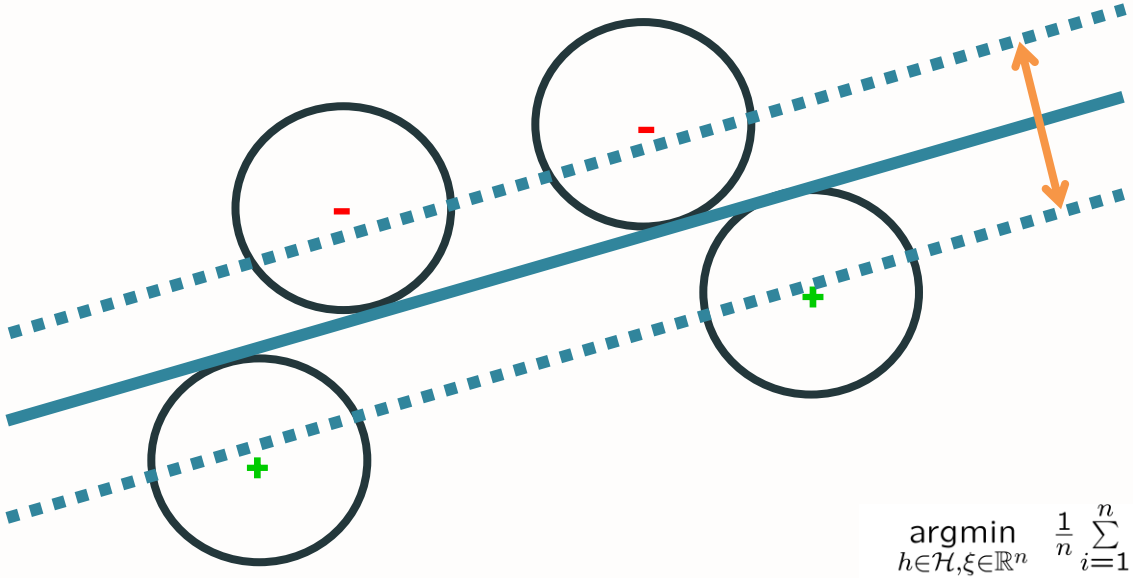




Maximum Margin Hyperplane

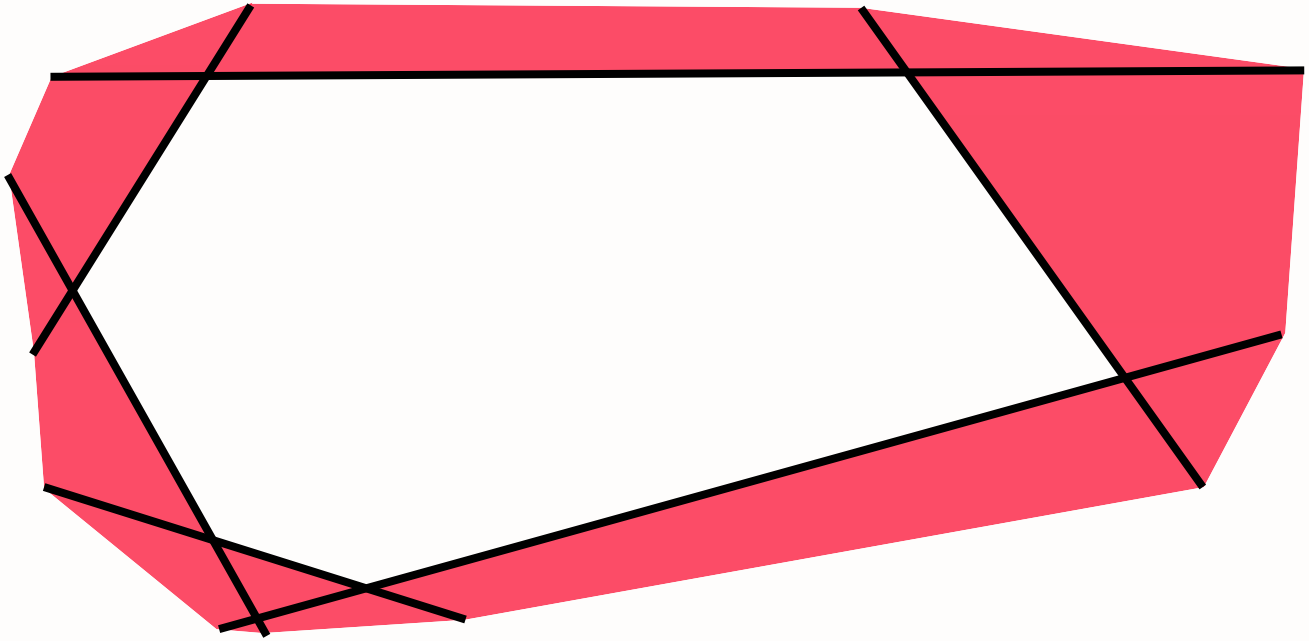


Maximum Margin Hyperplane

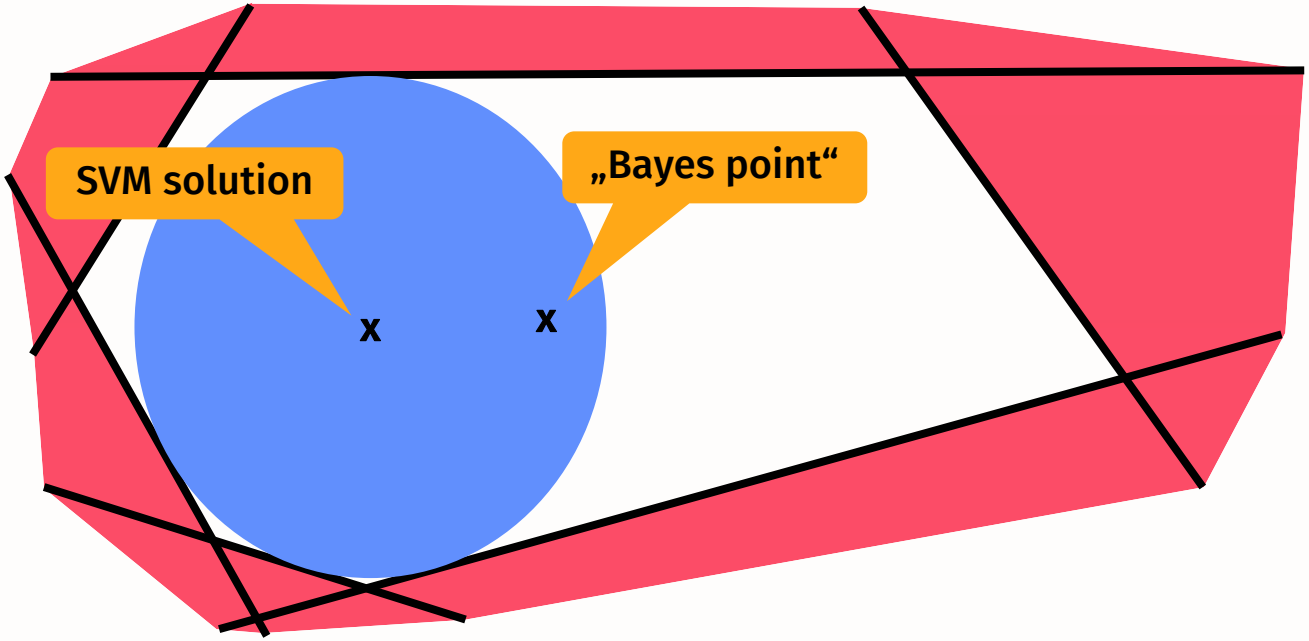


$$
\begin{aligned}
\underset{h\in\mathcal{H},\xi\in\mathbb{R}^n}{\text{argmin}} \quad & \frac{1}{n}\sum_{i=1}^{n}\xi_i + \nu\|h(\cdot)\|_{\mathcal{H}}^2 \\
\text{subject to:} \quad & y_i h(x_i) \geq 1-\xi_i \qquad i=1,\ldots,n \\
& \xi_i \geq 0 \qquad i=1,\ldots,n\ .
\end{aligned}
$$

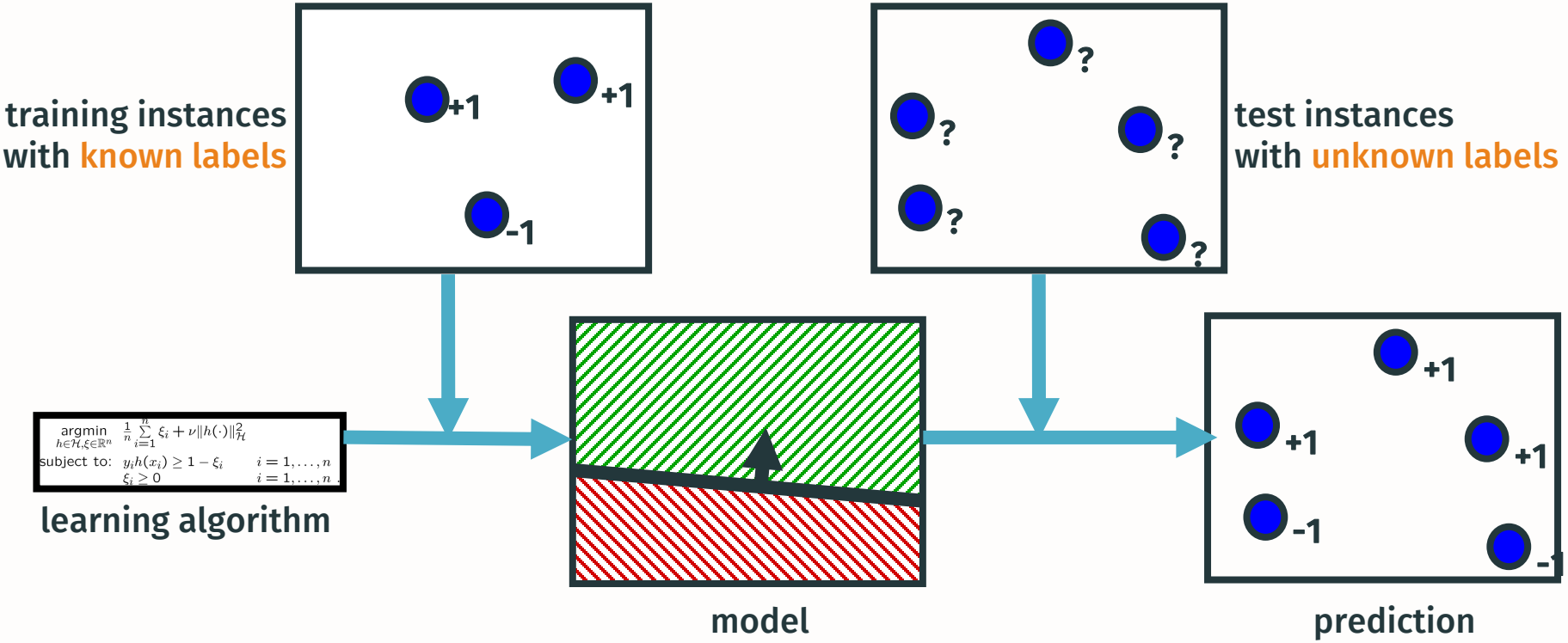
Version Space Geometry of SVMs



Version Space Geometry of SVMs



Supervised Classification



Kernel Methods, e.g., Support Vector Machines

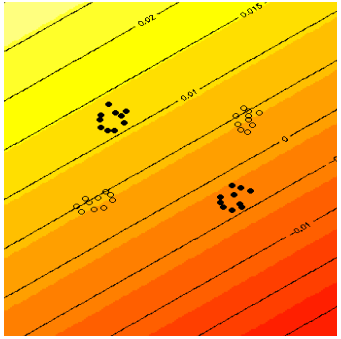
find linear separator in Hypothesis space (reproducing kernel Hilbert space)

maximally far from all labelled instances (robustness against measurement noise)

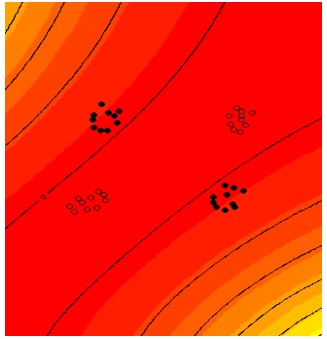
decision boundary in input space depends on kernel function

challenges for kernel design are validity (positive definite) and efficiency (polynomial time)

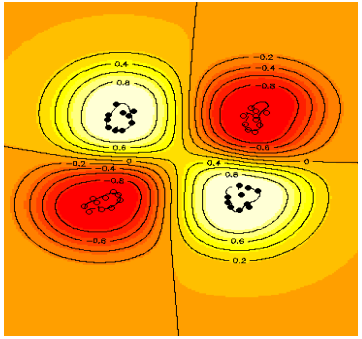
linear



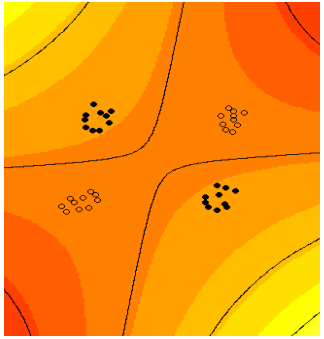
sigmoid



Gaussian



polynomial



kernel methods – a way out?

data zoo

algorithm zoo

2" IRWIN Marples

22

BLACK+DECKER
LITHIUM
20V

makita

DEWALT
20V MAX
DEWALT
20V MAX

BOSCH
Boschhammer

BLACK+DECKER

BOSCH

Fox

Kernels for Basic Terms

valid!

principles
represent individuals by typed terms
represent sets, multisets, etc. naturally

efficient!

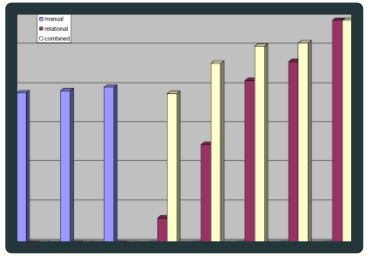
logic for learning (Hurd, 2003)
higher-order logic with polymorphic types
the terms of the logic are the terms of the typed λ-calculus,

effective?

basic terms
are essentially unambiguous closed terms of the logic
basic abstractions are used to represent sets, multisets, etc

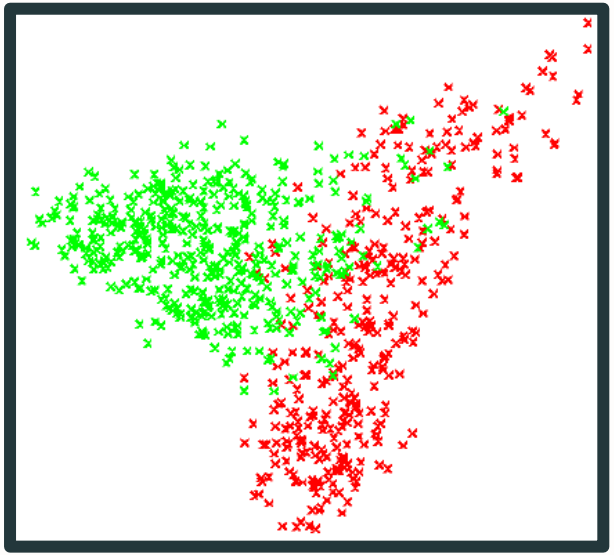
Structure Elucidation

```
type Spectrum      = Frequency -> Multiplicity
type Frequency     = Real with modifier Gaussian 0.6
data Multiplicity = s | d | t | q | 0 default 0
```

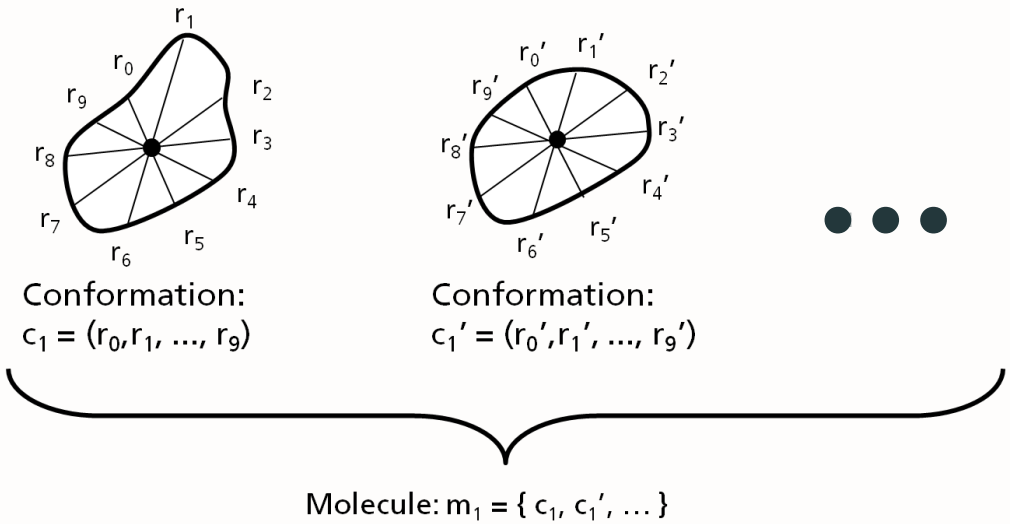


task
given the ^{13}C NMR spectrum of Diterpenes, predict the skeleton class it belongs to

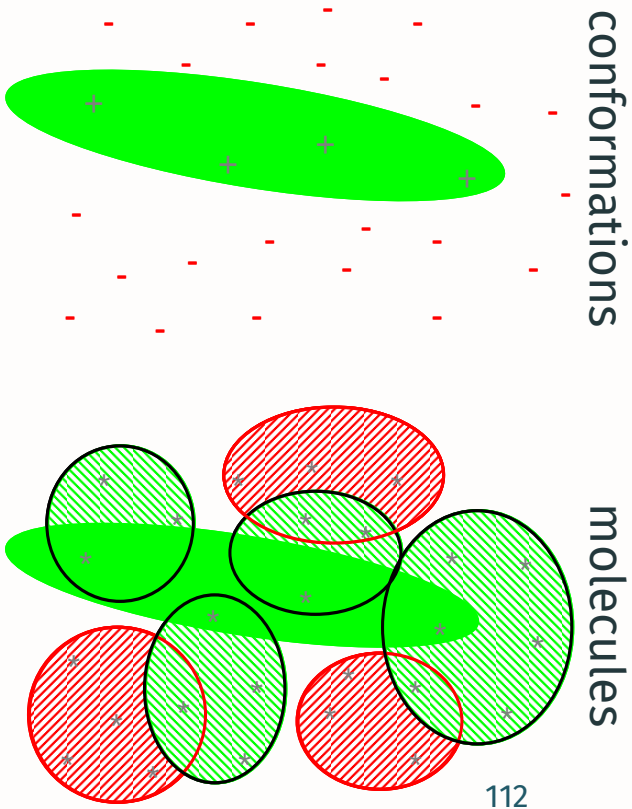
data
1503 spectra classified into 23 skeleton types



Multi-Instance Kernels and a glimpse of Learning Theory

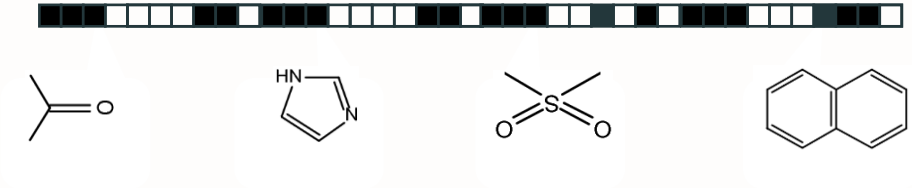


$$f^*_{MI}(m) = +1 \Leftrightarrow \exists c \in m : f^*_I(c) = +1$$



multi-instance concepts can be separated with kernels
for basic terms & the number of mistakes can be bounded.

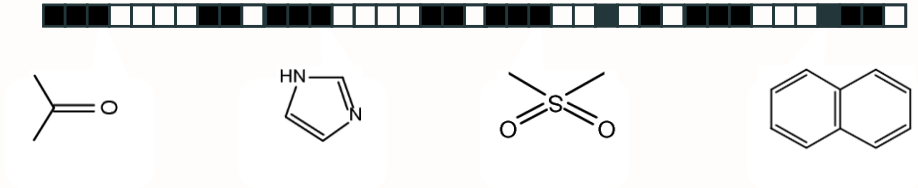
Graph Kernels



common approach:

define kernel as some function of the **common substructures**
(compare kernels for strings and trees)

Graph Kernels



common approach:

define kernel as some function of the **common substructures**
(compare kernels for strings and trees)

computing the “all patterns” graph kernel is NP-hard;
even for “all paths”, “all cycles”, or “all connected subgraphs”
(with subgraph-isomorphism as the embedding).

computing **complete graph kernels**
is at least as hard as deciding graph isomorphism

$$k(G, \cdot) = k(G', \cdot) \Leftrightarrow G \simeq G'$$

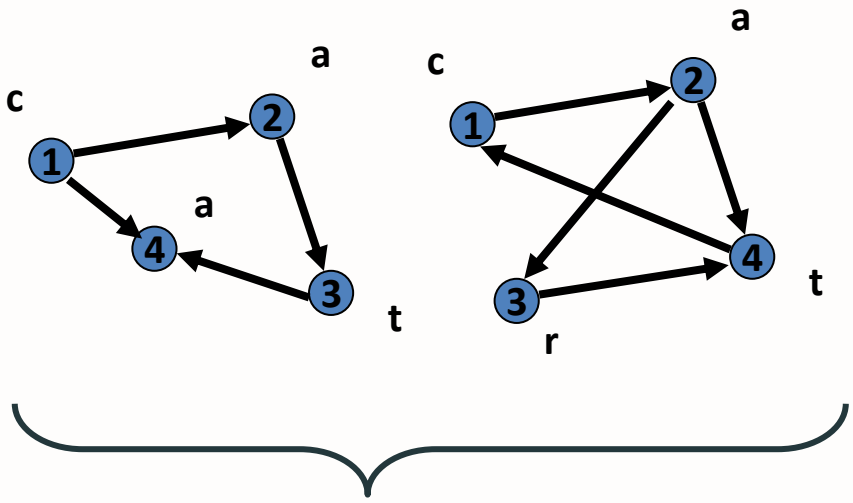
All-Walks Kernels

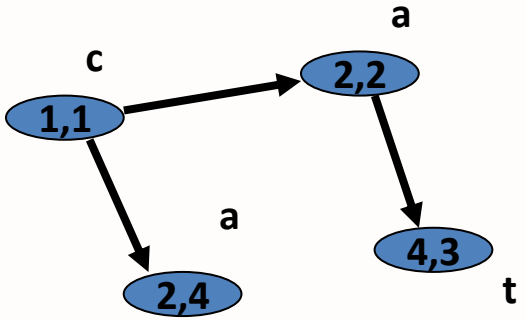
resort to non-complete graph kernels

consider as patterns the (multi-)set of all labelled walks contained in a graph

while there can be infinitely many walks we can measure the common ones efficiently

$$k_\times(G_1, G_2) = \mathbf{1}^\top \left[\lim_{n\to\infty} \sum_{i=0}^{n} \eta_i E_\times^i \right] \mathbf{1}$$

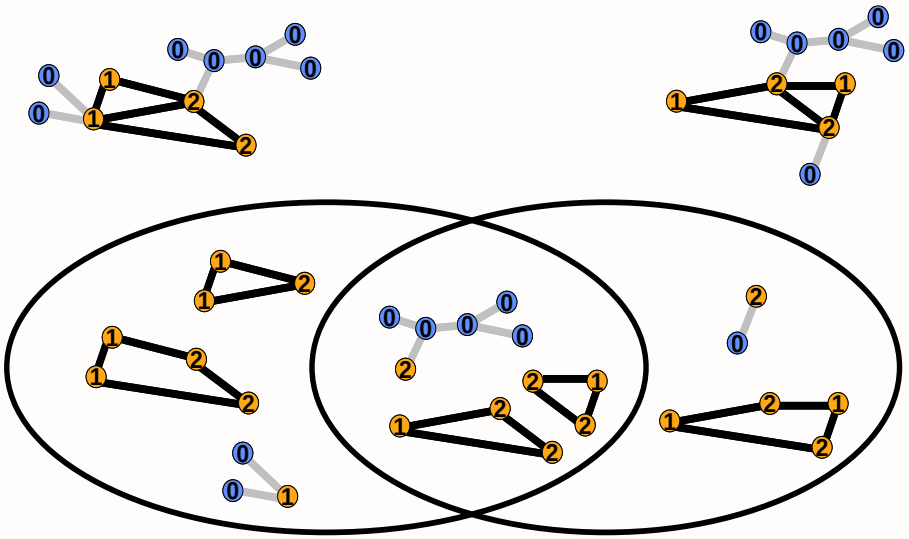


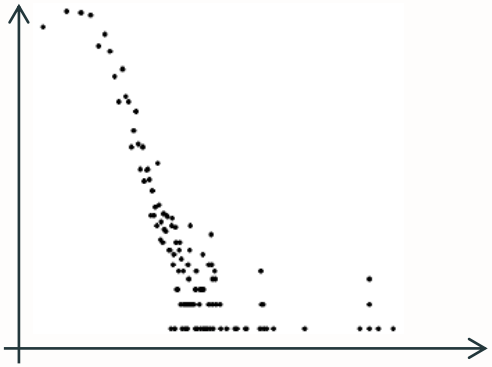


Cyclic Pattern Kernels

consider as patterns the set of all labelled cycles contained in a graph as well as all trees in the forrest of bridges

computing CPK is NP-hard but fixed-parameter tractable





log/log plot for HIV dataset
#cycles over #molecules

</talk>

Ta!

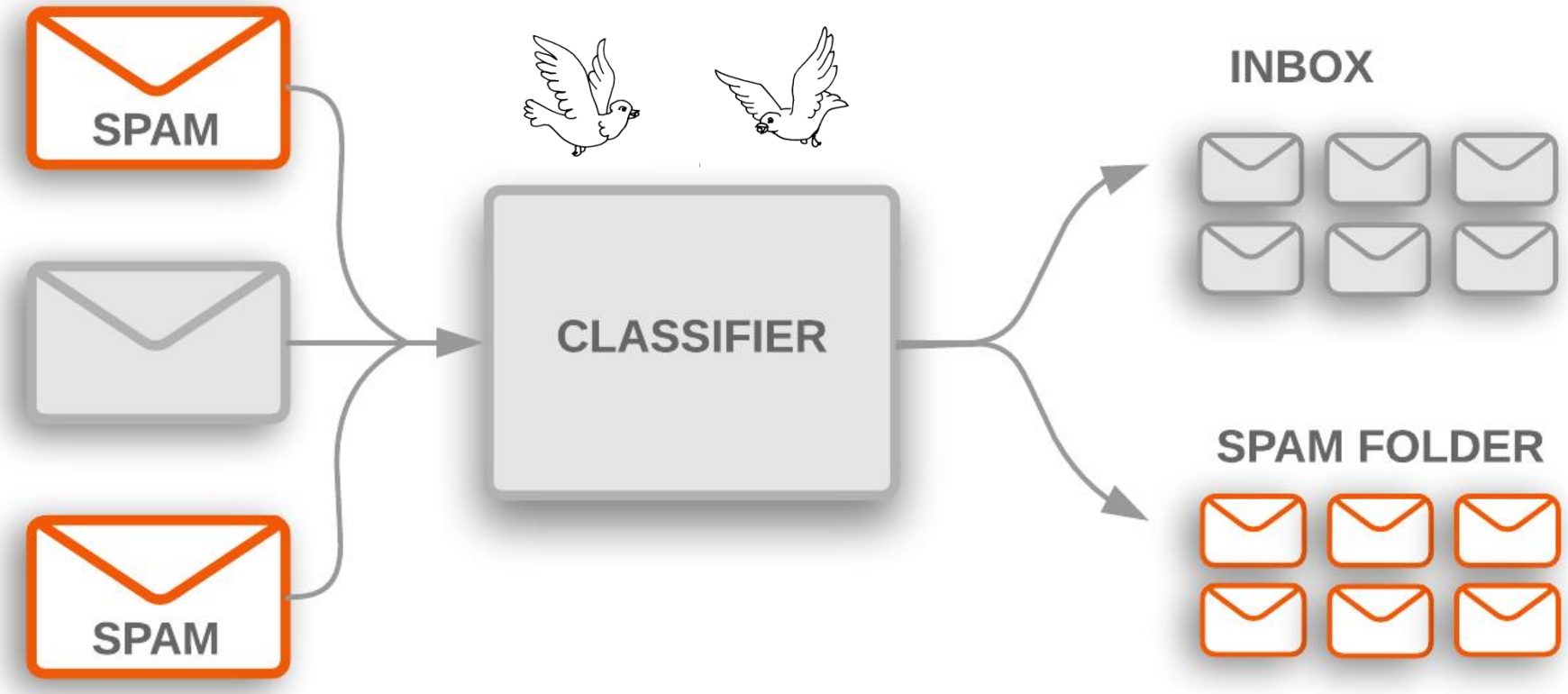
Acknowledgements

collaborators:
Prof. Peter Flach, Prof. John W. Lloyd, Prof. Alex Smola, Prof. Stefan Wrobel, Prof. Tobias Scheffer, Prof. Jürgen Bajorath, Prof. SVN Vishwanathan, Prof. Kristian Kersting, Prof. Stephane Canu, Prof. Jan Ramon, Prof. Gilles Gasso, Prof. Kurt Driessens, Prof. Quoc V. Le, Prof. Jonathan D. Hirst Simon J. F. Macdonald, Thomas Mcinally, Dr. Karina Zapien, Dr. Claudio Lucchese, Dr. Hanna Geppert, Dr. Dagmar Stumpfe, Dr. Yasmin Altun, Dr. Gemma C. Garriga, Dr. Adam Kowalczyk, Dr. Sandy Moens, Simon Burton, Steven A. Oatley,

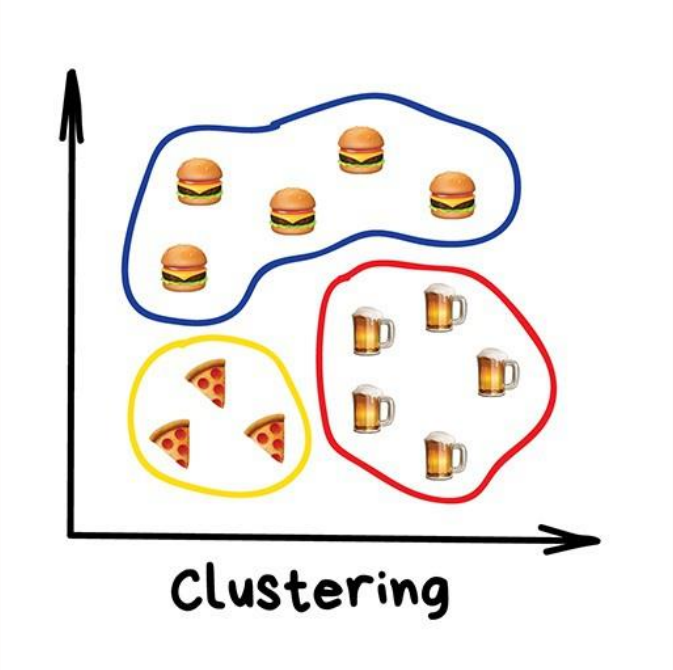
group & alumni:
Prof. Ulf Brefeld, Prof. Roman Garnett, Prof. Dr. Nicolò Navarin, Dr. Mario Boley, Dr. Tamas Horvath, Dr. Michael Kamp, Dr. Olana Missura, Dr. Daniel Paurat, Dr. Dino Oglic, Dr. Shankar Vembu, Bo Kang, Jens Humrich, Laurentiu Ilici, Jiojian Wang, *Katrin Ullrich, Alexe Haywood (UoN), Joeseph Redshaw (UoN), Maximilian Thiessen (TU Wien), & we are hiring@TU Wien*

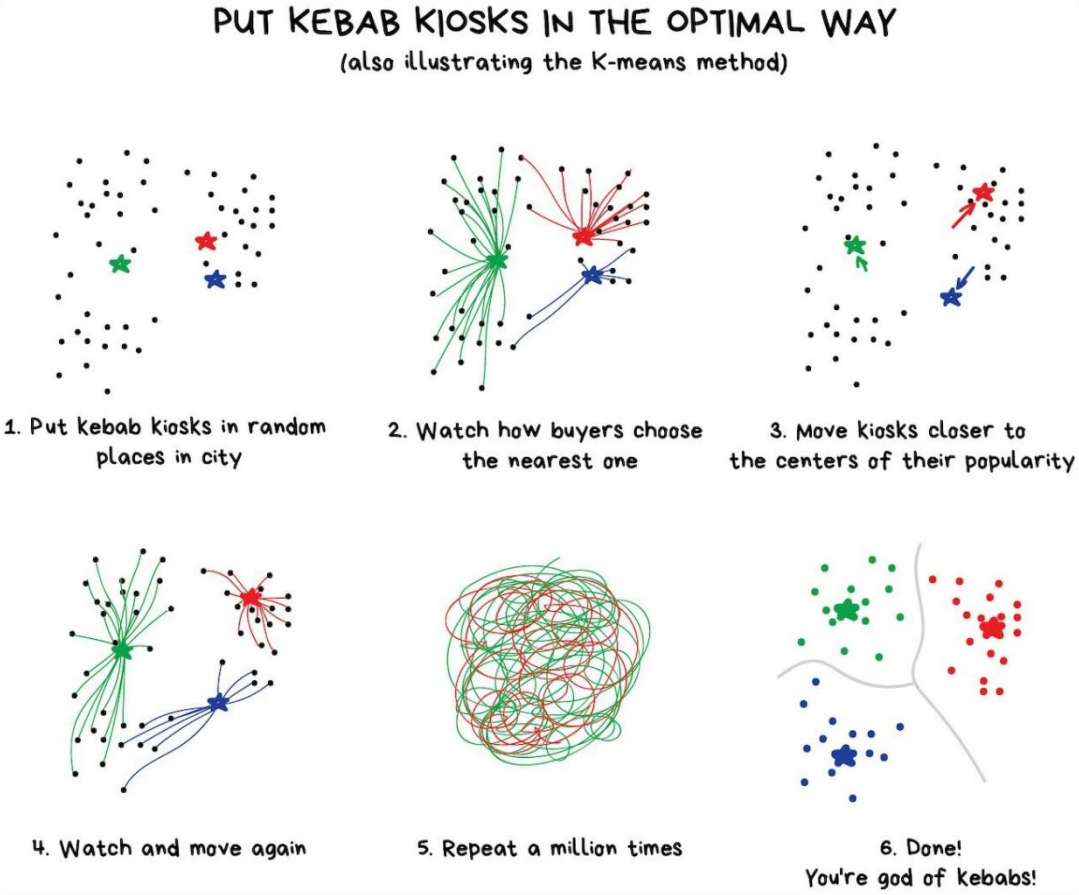
appendix

Classic Example Application for Supervised Learning



Unsupervised Learning / *k*-means





Computer Vision using Machine Learning

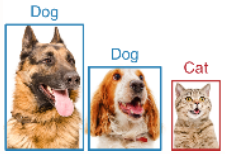
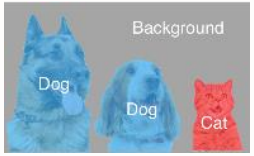
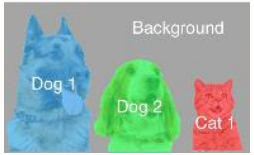


image classification

object detection

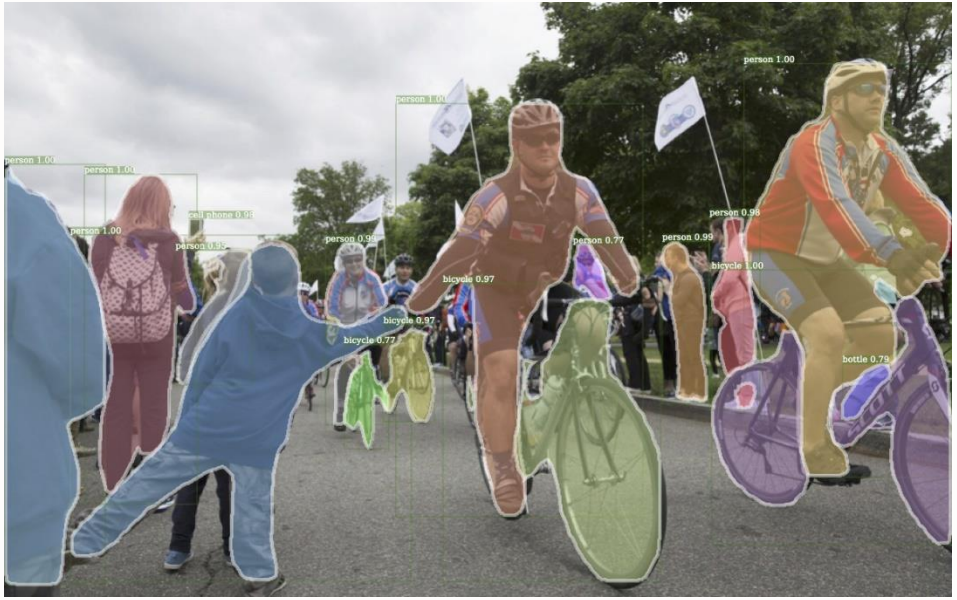


semantic segmentation

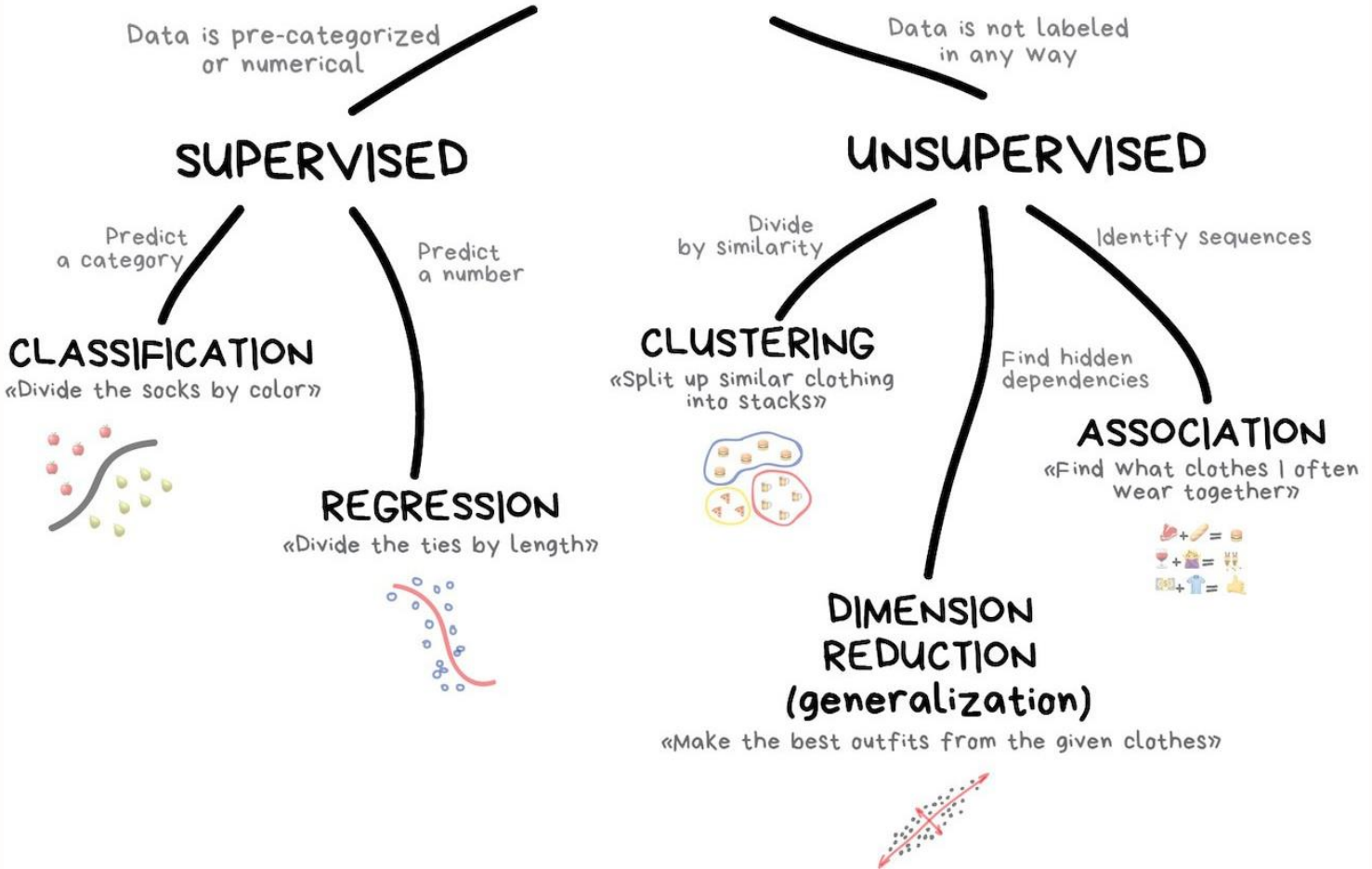


instance segmentation

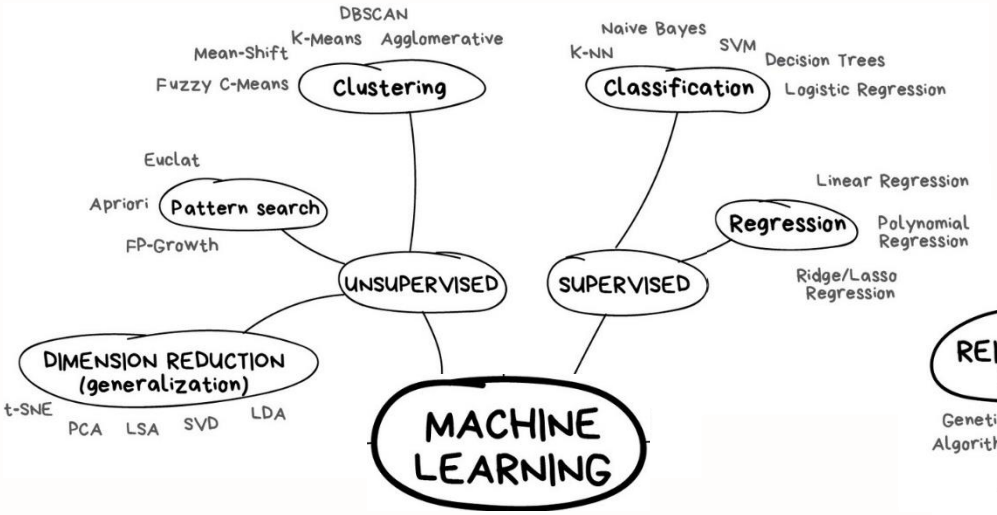
pose estimation

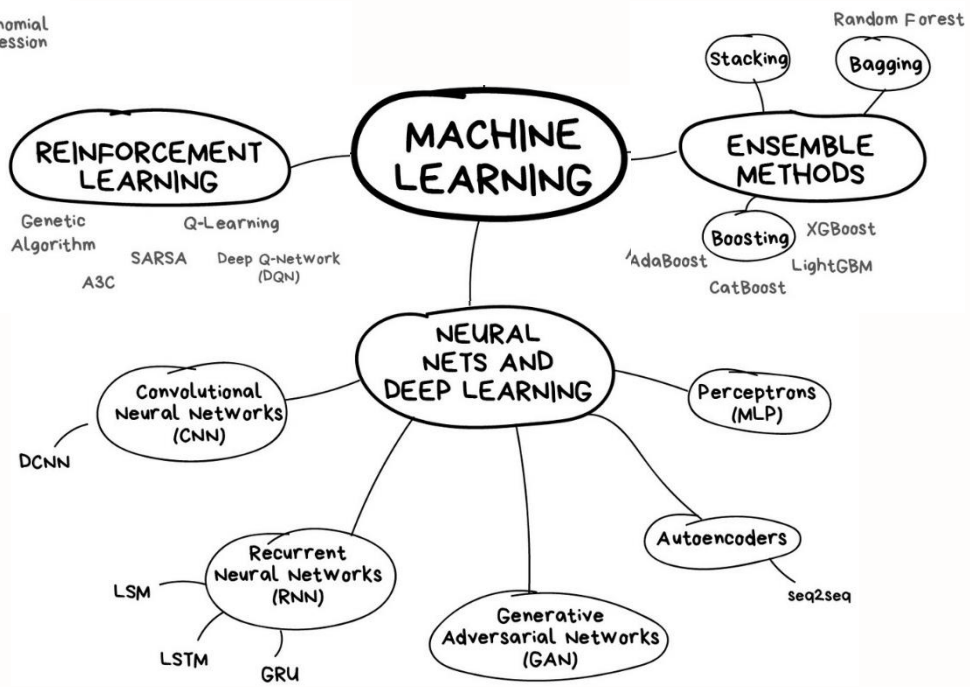


Types of Machine Learning

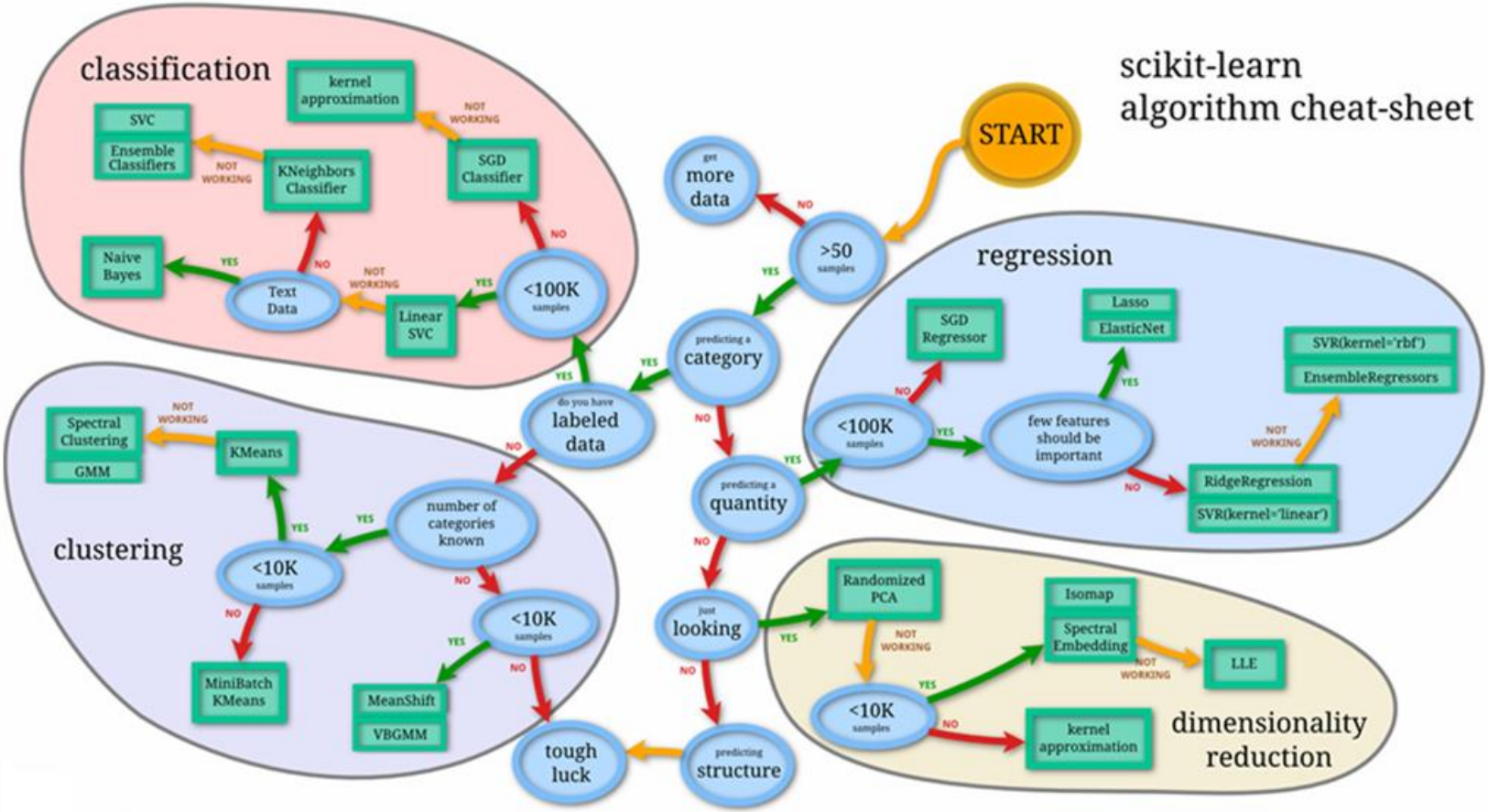


Machine Learning Algorithm Zoo

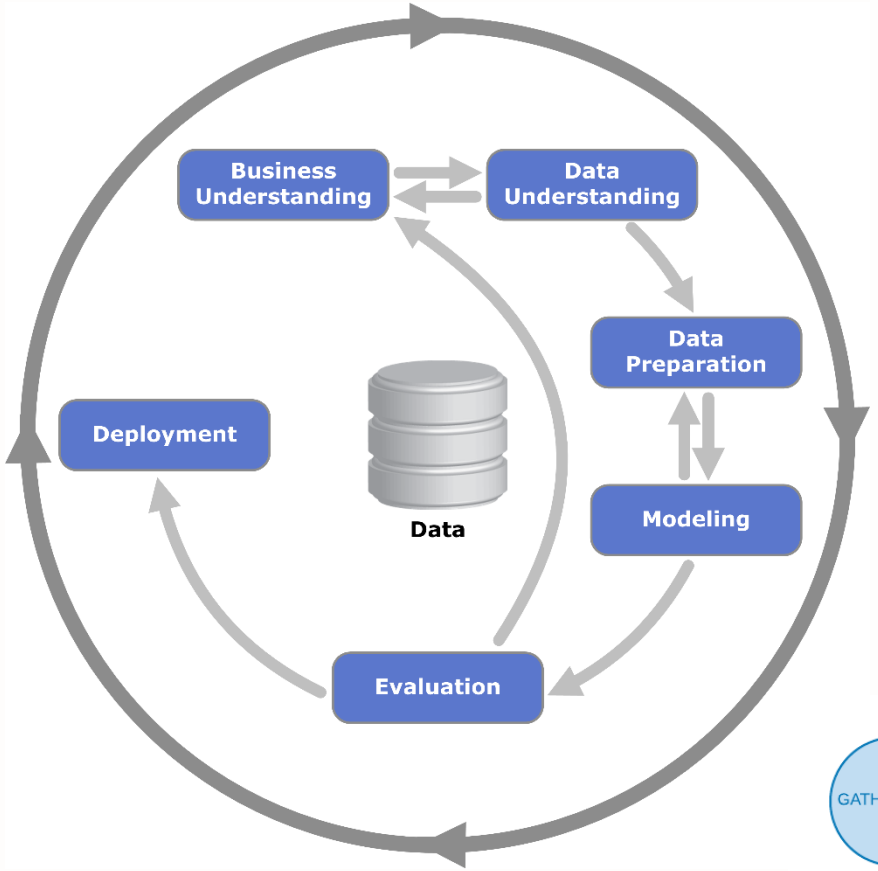


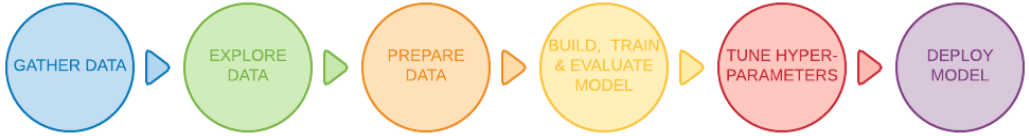


Machine Learning Algorithm Zoo Guide



Machine Learning Process (CRISP-DM vs Microsoft Azure)





the state-of-the-art

Machine Learning Timeline

1763 Thomas Bayes

1805 Adrien-Marie Legendre (LSR)

1812 Pierre-Simon Laplace (Bayes' Theorem)

1913 Andrey Markov (Markov Chains/MCMC)

1950 Alan Turing (test + GA)

1951 Marvin Minsky (NN:SNARC)

1957 Frank Rosenblatt (perceptron)

1967 nearest neighbour

1970 AD/Backpropagation

1989 Chris Watkins (RL:QL)

1995 Tin Kam Ho (RF)

1995 Corinna Cortes and Vladimir Vapnik (SVM)

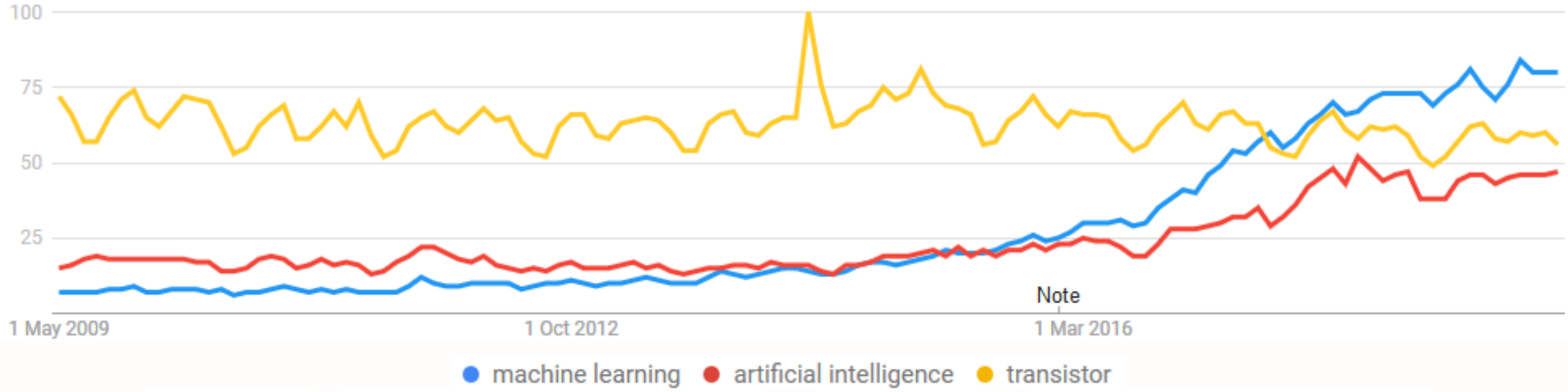
1951 checkers

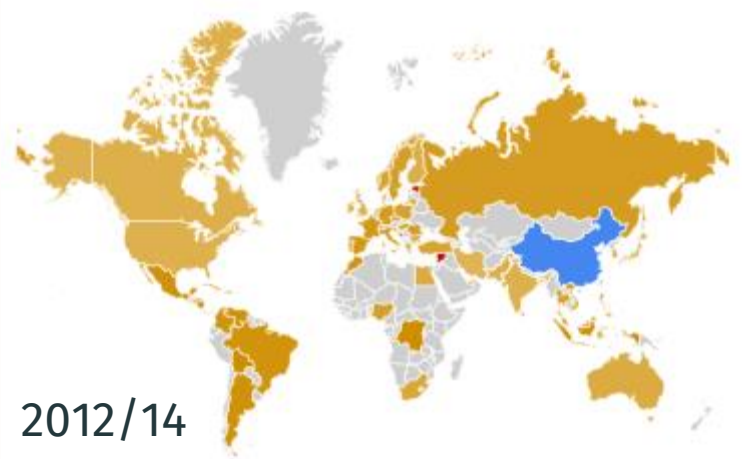
1992 Backgammon, rivalling top human players

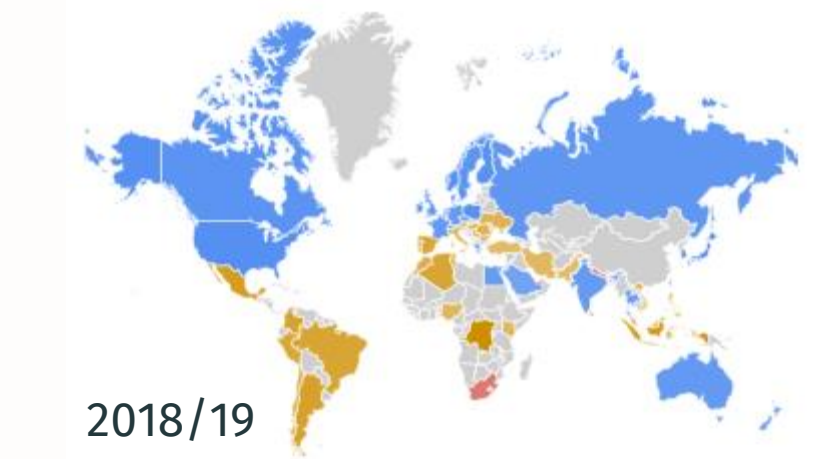
1997 Chess, beating the world champion

2016 Go, beating the world champion

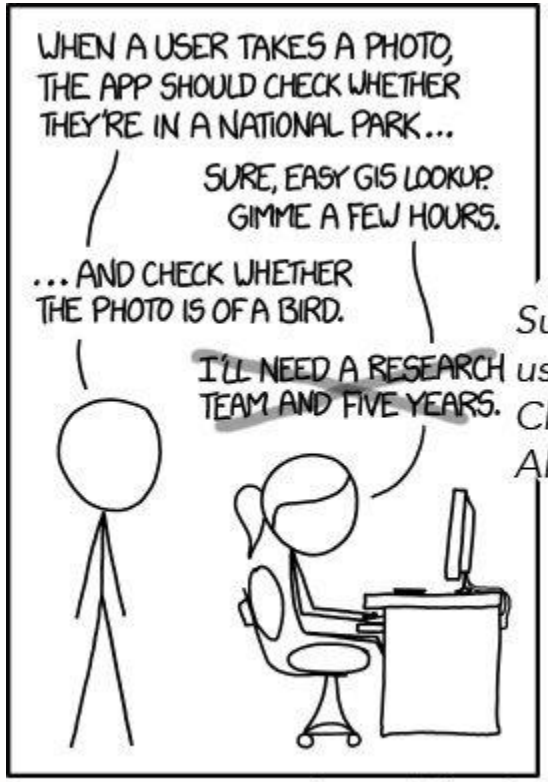
Machine Learning, a global trend







Machine Learning Tools



Sure, I'll just use the Google Cloud Vision API

Source: http://xkcd.com/1425/

Machine Learning and Companies (Sponsoring, 2017)

5 ×	Diamond	×	$80,000
15 ×	Platinum	×	$40,000
15 ×	Gold	×	$20,000
43 ×	Silver	×	$10,000
6 ×	Bronze	×	$5,000

$1.76 M

KDD2017

HUAWEI
Bloomberg
Microsoft
BOSCH
facebook
Alibaba Group
VISA
Visa Research
Bai
rapidminer
IBM Research
Google
sas
SIEMENS
tableau
Disney Research
flipkart
ebay
YAHOO! LABS
HUAWEI
collective
Stanford
technicolor
leidos
ZEMENTIS
KNIME
dstillery
ORACLE LABS
kaggle
MIT
NYU STERN
KDnuggets
AMR
NSF
Donors Choose.org
SALFORD SYSTEMS
DataRobot

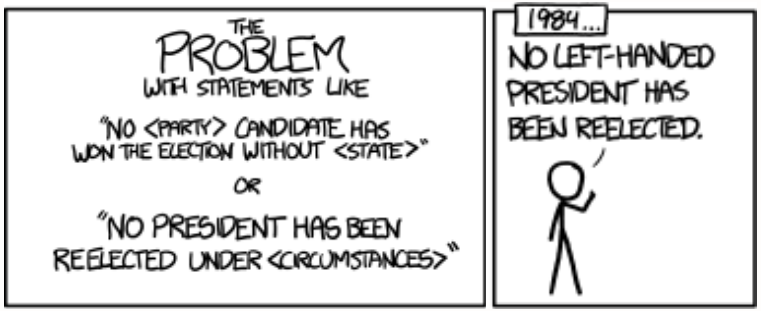
overfitting

THE
PROBLEM
WITH STATEMENTS LIKE

"NO <PARTY> CANDIDATE HAS
WON THE ELECTION WITHOUT <STATE>"

OR

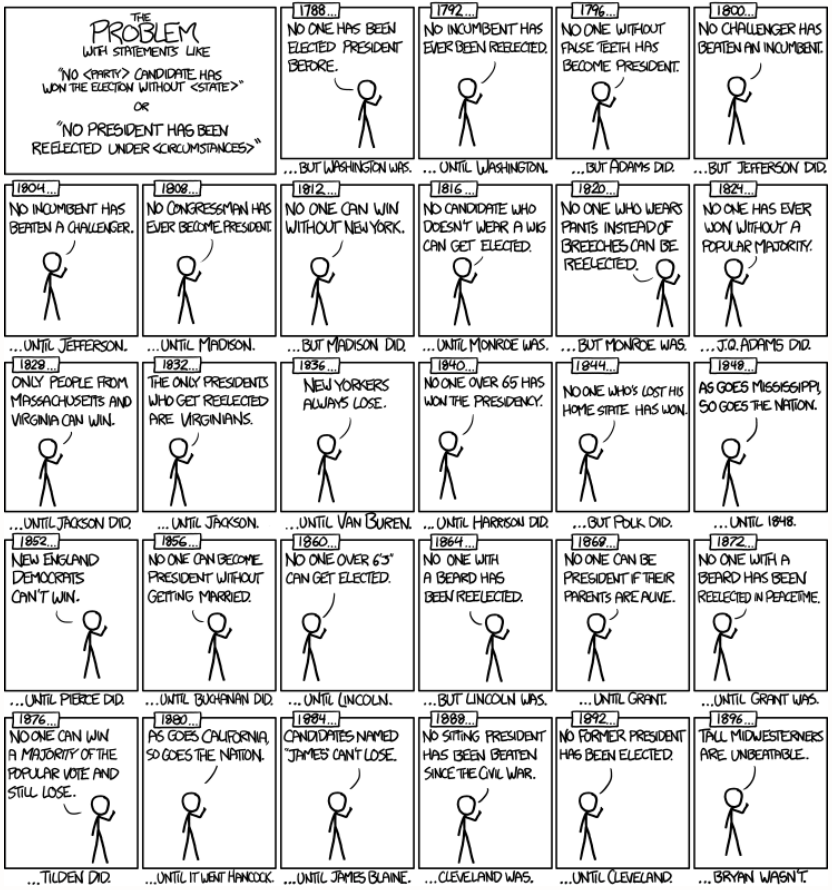
"NO PRESIDENT HAS BEEN
REELECTED UNDER <CIRCUMSTANCES>"

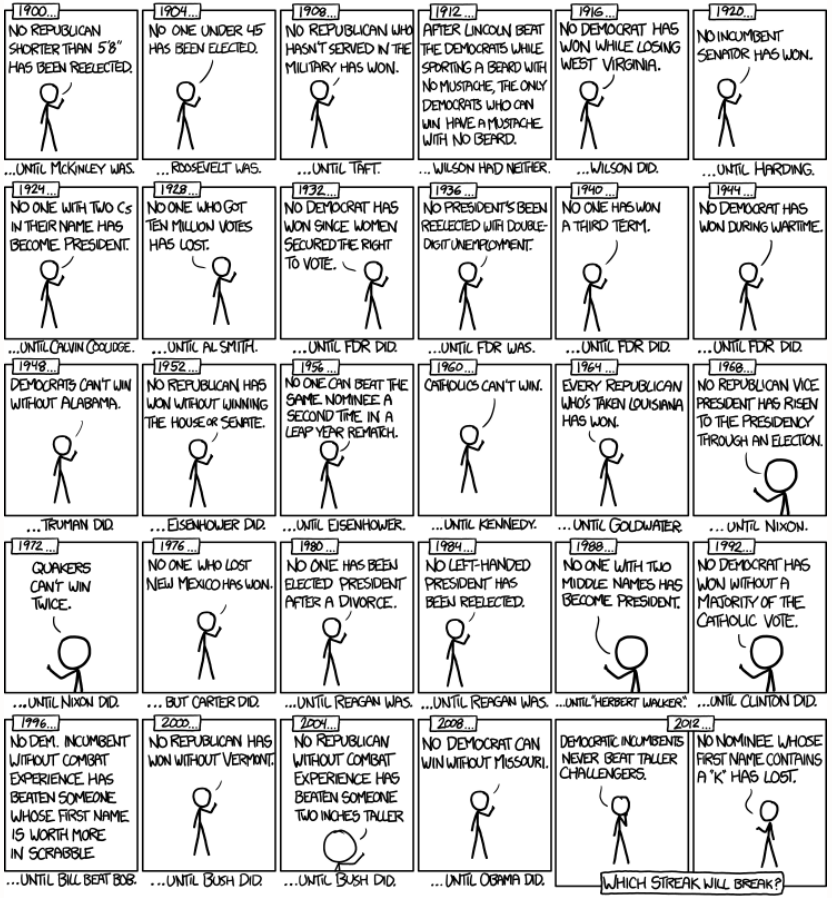

THE PROBLEM WITH STATEMENTS LIKE
"NO <PARTY> CANDIDATE HAS WON THE ELECTION WITHOUT <STATE>"
OR
"NO PRESIDENT HAS BEEN REELECTED UNDER <CIRCUMSTANCES>"
1984...
NO LEFT-HANDED PRESIDENT HAS BEEN REELECTED.

THE PROBLEM WITH STATEMENTS LIKE
"NO <PARTY> CANDIDATE HAS WON THE ELECTION WITHOUT <STATE>"
OR
"NO PRESIDENT HAS BEEN REELECTED UNDER <CIRCUMSTANCES>"
1984...
NO LEFT-HANDED PRESIDENT HAS BEEN REELECTED.
2008...
NO DEMOCRAT CAN WIN WITHOUT MISSOURI.
...UNTIL REAGAN WAS.

THE PROBLEM WITH STATEMENTS LIKE
"NO <PARTY> CANDIDATE HAS WON THE ELECTION WITHOUT <STATE>"
OR
"NO PRESIDENT HAS BEEN REELECTED UNDER <CIRCUMSTANCES>"
1984...
NO LEFT-HANDED PRESIDENT HAS BEEN REELECTED.
...UNTIL REAGAN WAS.
2008...
NO DEMOCRAT CAN WIN WITHOUT MISSOURI.
...UNTIL OBAMA DID.

Illustration

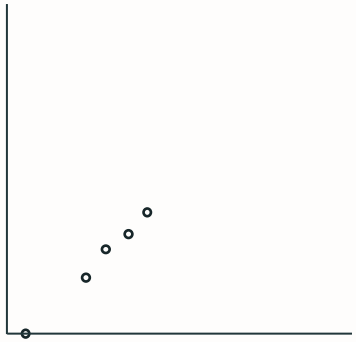


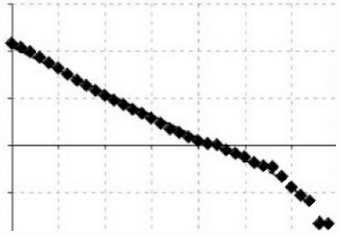


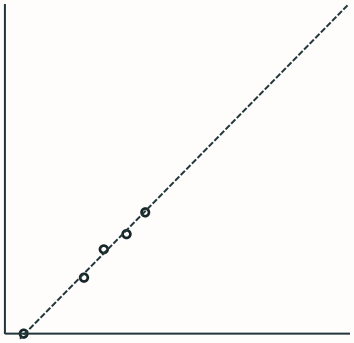
[xkcd, 2012]

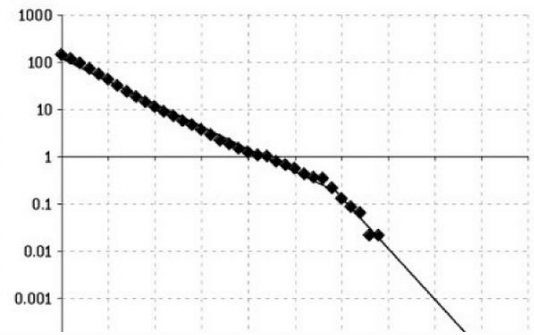
Real-World Examples

“Borussia Dortmund has never lost a Champions League home game to a Spanish opponent when they have lost the previous Bundesliga away game by more than two goals, having scored at least once themselves.”








1000
100
10
1
0.1
0.01
0.001

Cramming more components onto integrated circuits

With unit cost falling as the number of components per circuit rises, by 1975 economics may dictate squeezing as many as 65,000 components on a single silicon chip

By Gordon E. Moore

Director, Research and Development Laboratories, Fairchild Semiconductor division of Fairchild Camera and Instrument Corp.

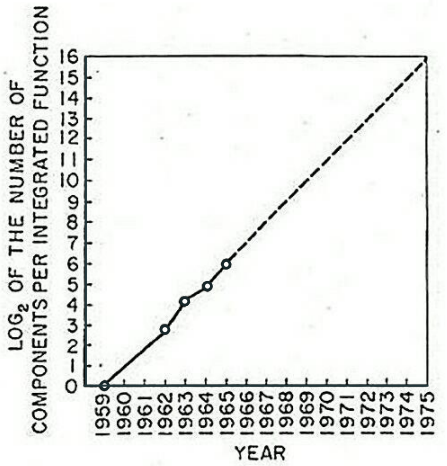
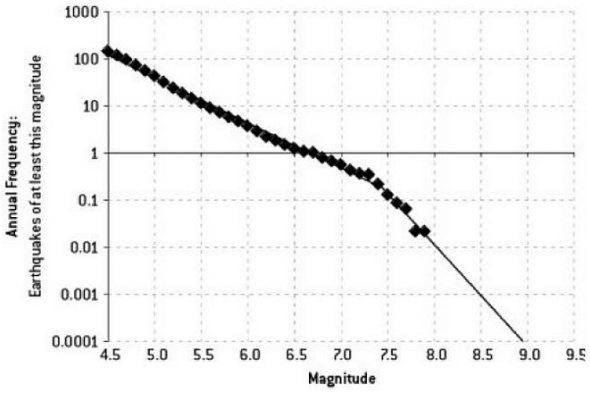


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

FIGURE 5-7C: TŌHOKU, JAPAN EARTHQUAKE FREQUENCIES CHARACTERISTIC FIT



[Gordon Moore, 1965 ; Nate Silver, "The Signal and the Noise"]

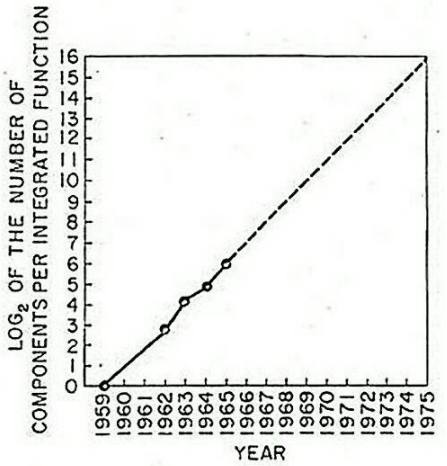
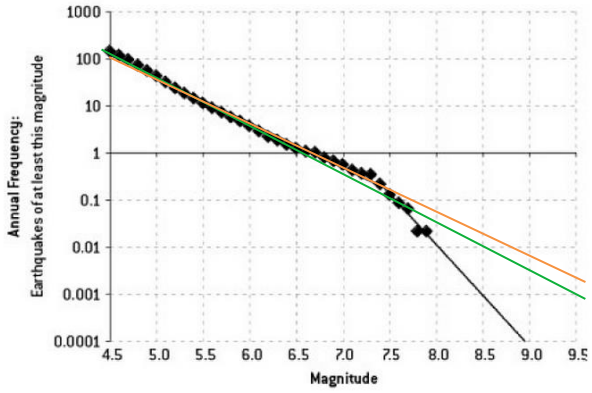
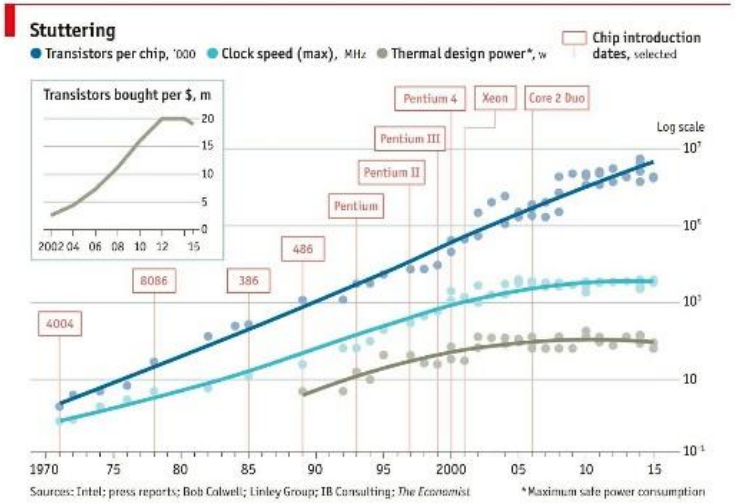
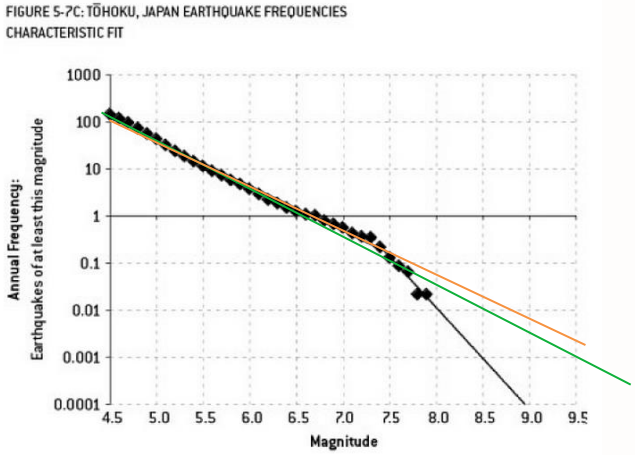


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

FIGURE 5-7C: TŌHOKU, JAPAN EARTHQUAKE FREQUENCIES CHARACTERISTIC FIT

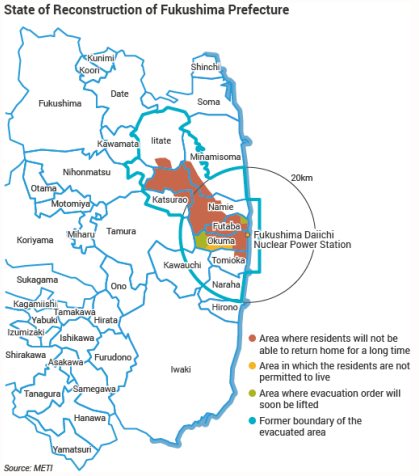






Japan earthquake: Explosion at Fukushima nuclear plant

12 March 2011



NUCLEAR WARNING: Radioactive contamination from Fukushima drifts to America

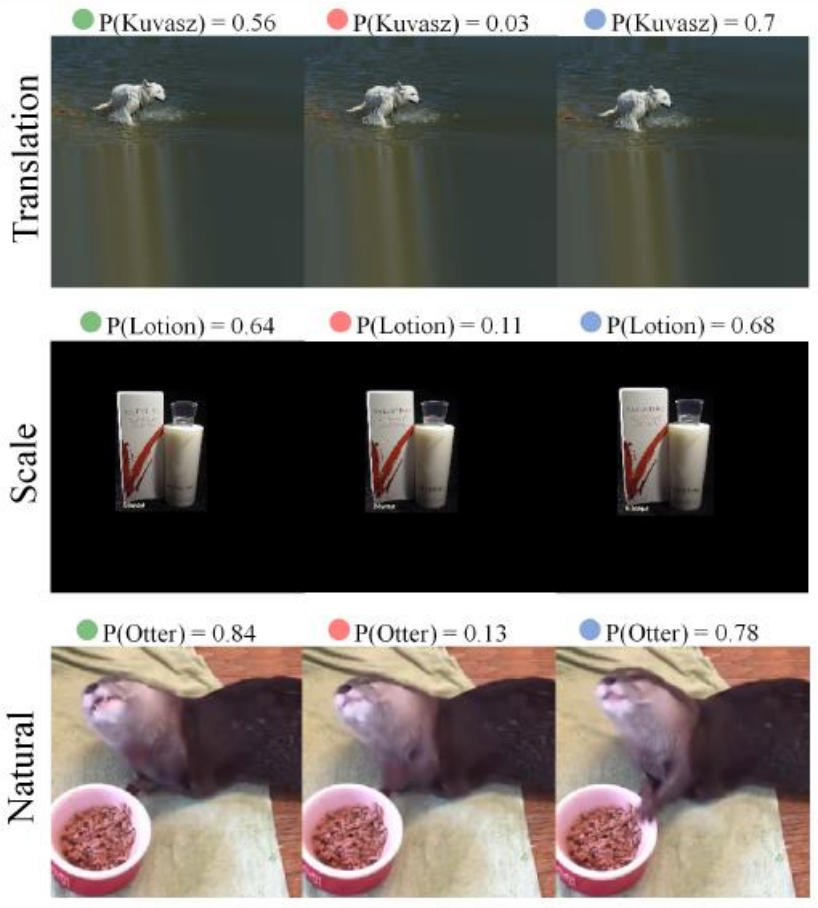
RADIOACTIVE contamination from Japan's Fukushima Daichi nuclear power plant has drifted as far as America, experts claim.

By EMILY FERGUSON

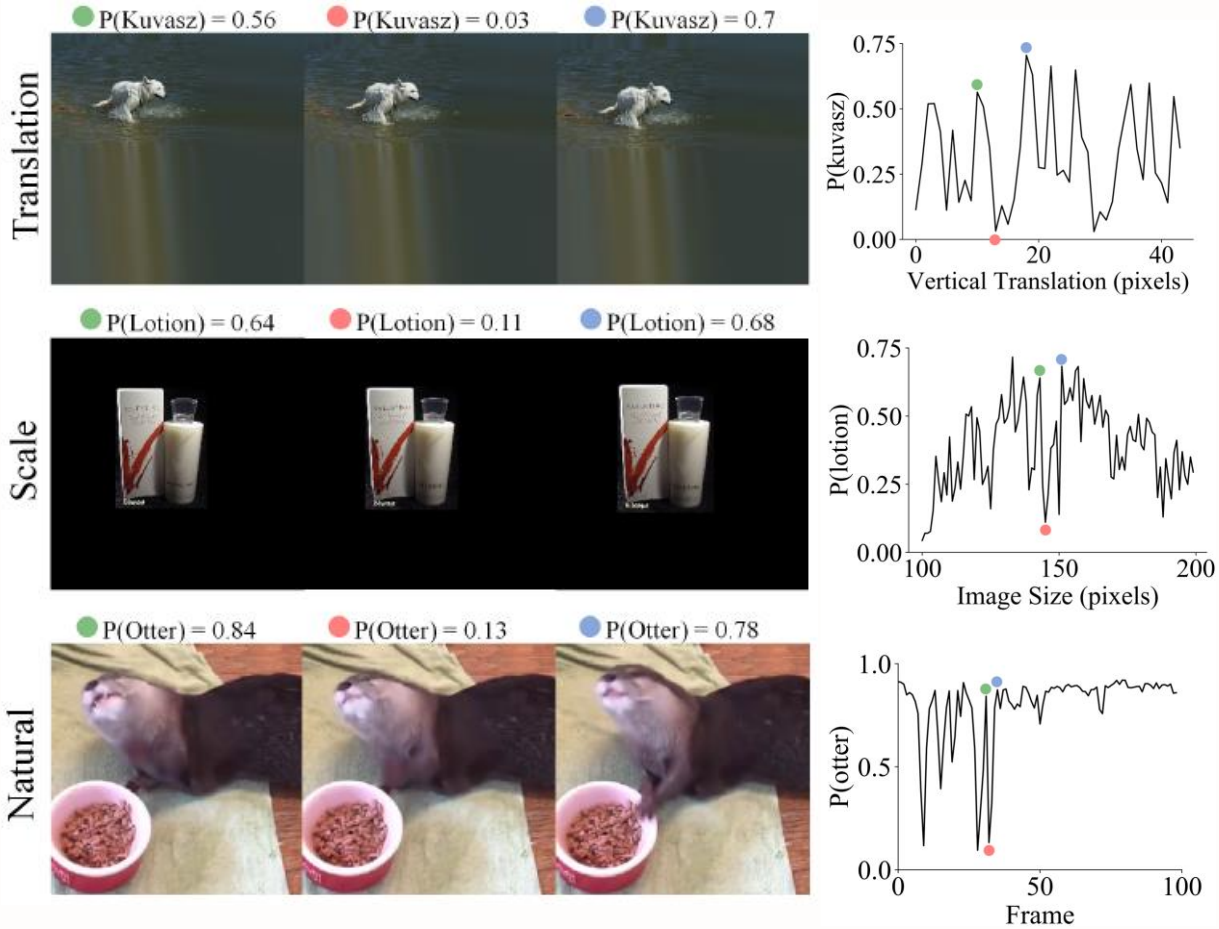
[Economist, 2014; World nuclear association, 2018]

Is this really a problem in machine learning?

Is this really a problem in machine learning?



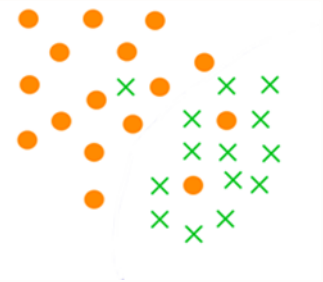
Is this really a problem in machine learning?



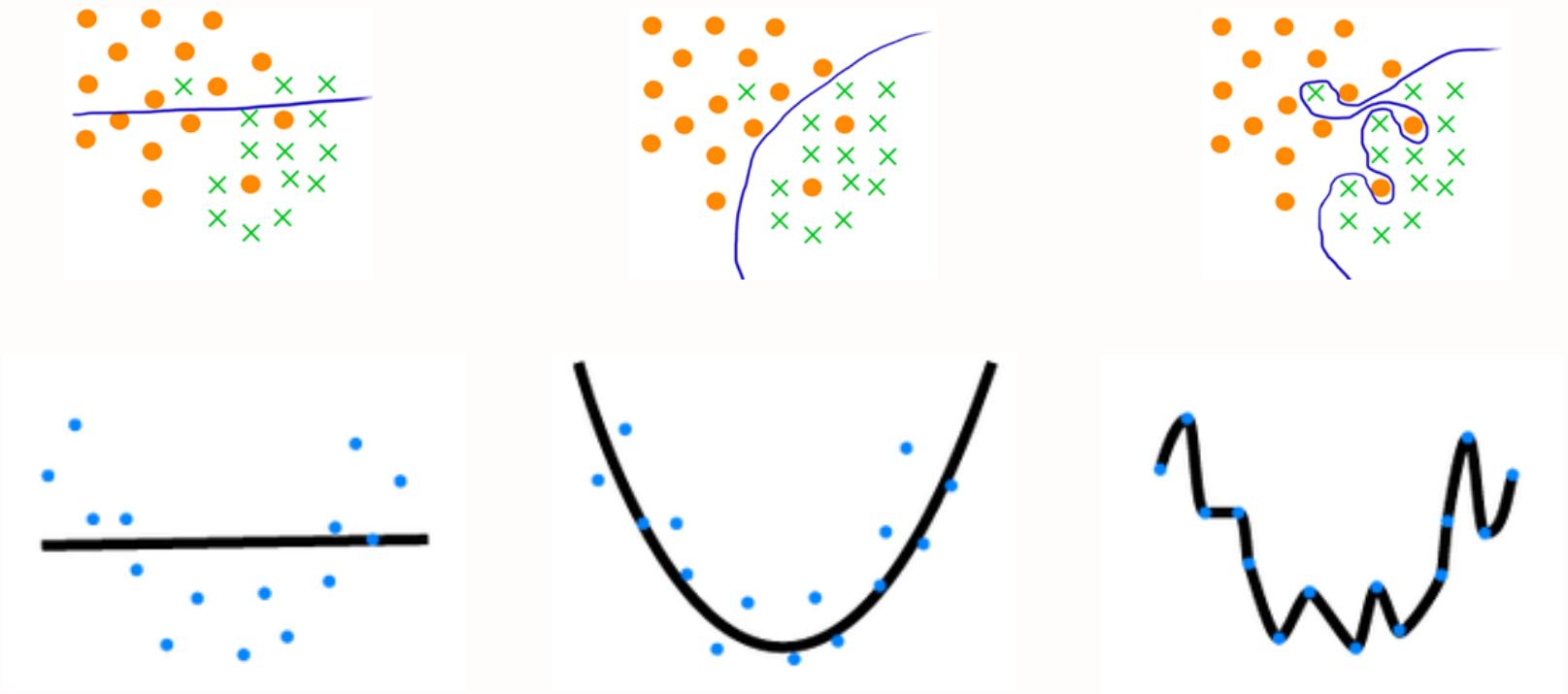
[Azulay & Weiss, 2019; Rosenfeld et al, 2018]

THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Classification and Regression Examples

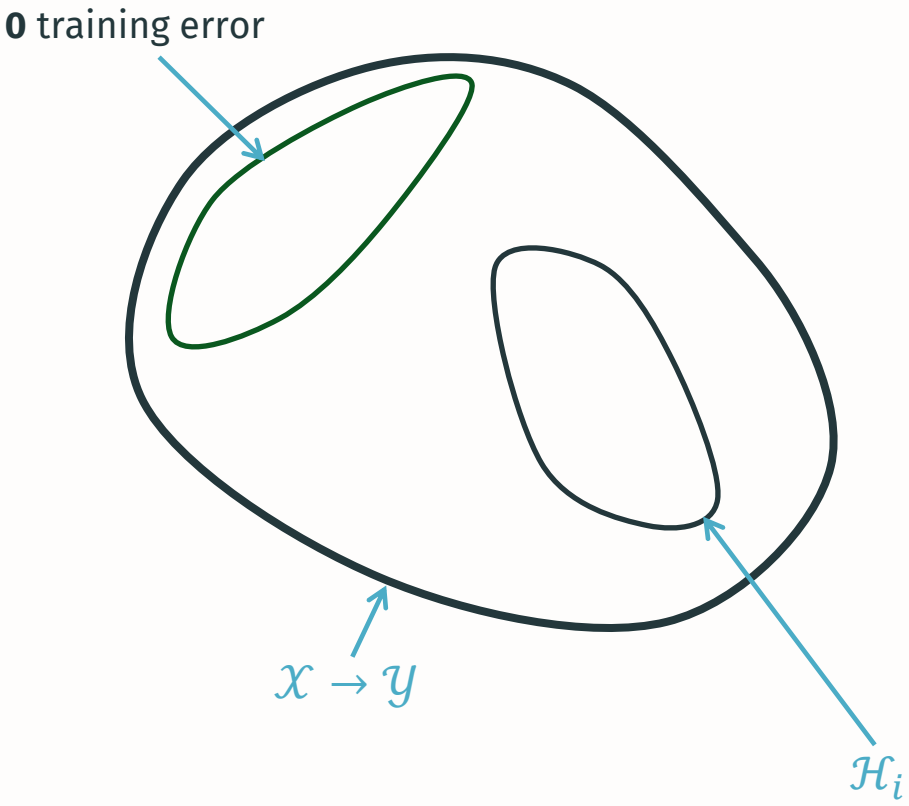


Under-, “Perfect”-, and Over-fitting Examples

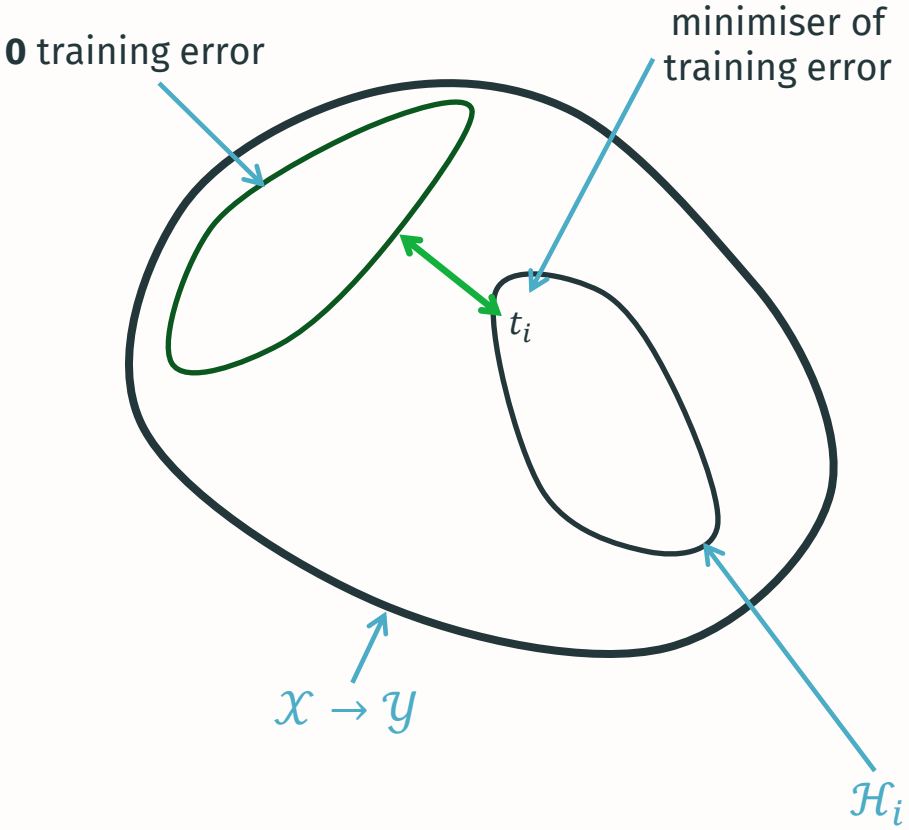


[Joglekar, 2018; Despois, 2018]

Risk Structure

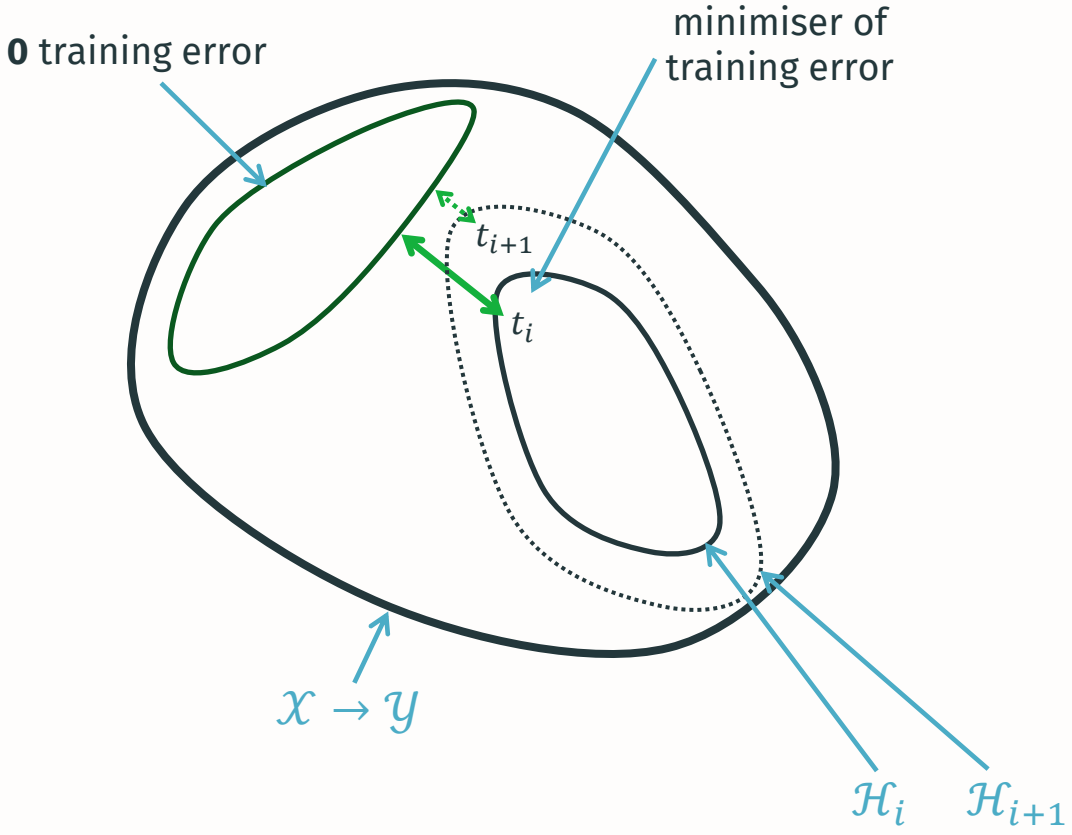


Risk Structure



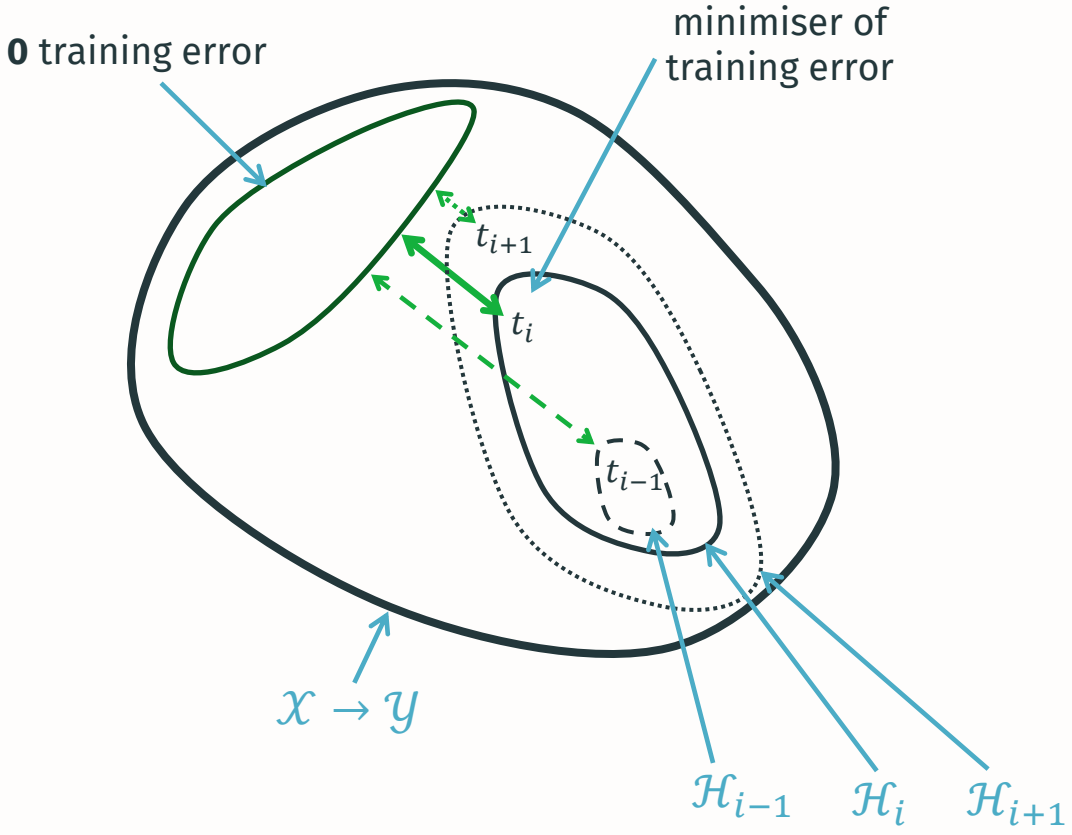
Risk Structure

$\mathcal{H}_i \subseteq \mathcal{H}_{i+1}$



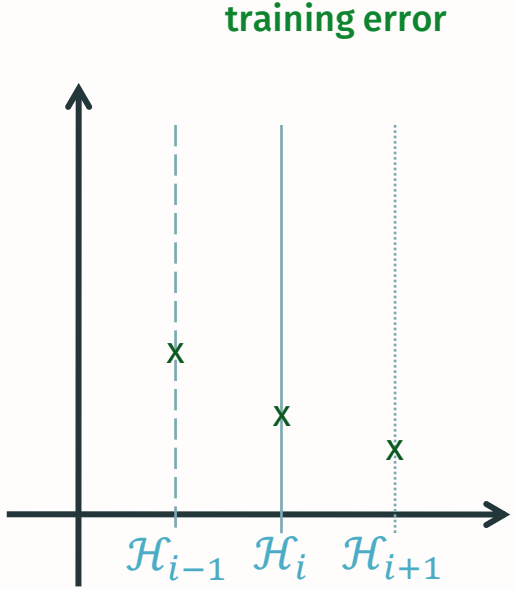
Risk Structure

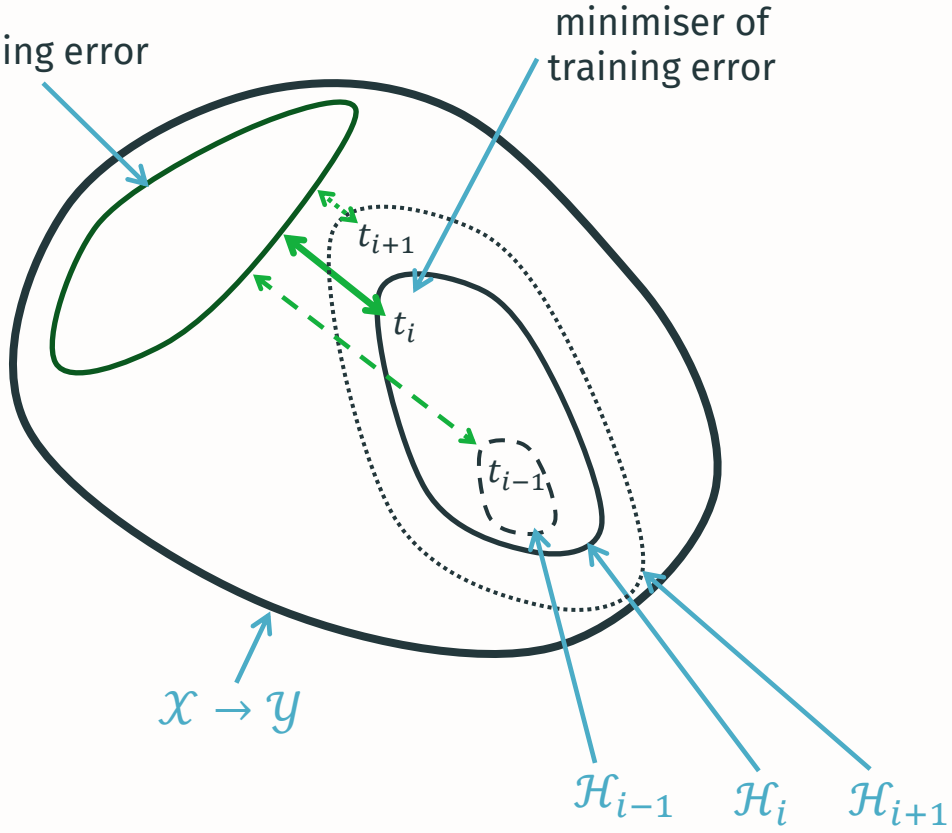
$\mathcal{H}_{i-1} \subseteq \mathcal{H}_i \subseteq \mathcal{H}_{i+1}$



Risk Structure

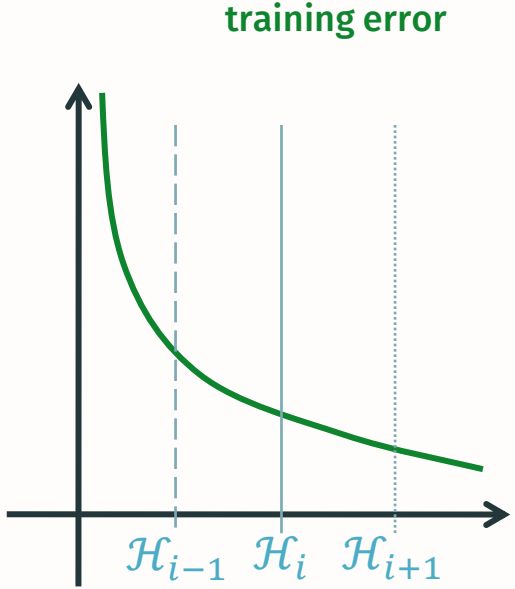
$$\cdots \subseteq \mathcal{H}_{i-1} \subseteq \mathcal{H}_i \subseteq \mathcal{H}_{i+1} \subseteq \cdots \subseteq \mathcal{X} \to \mathcal{Y}$$

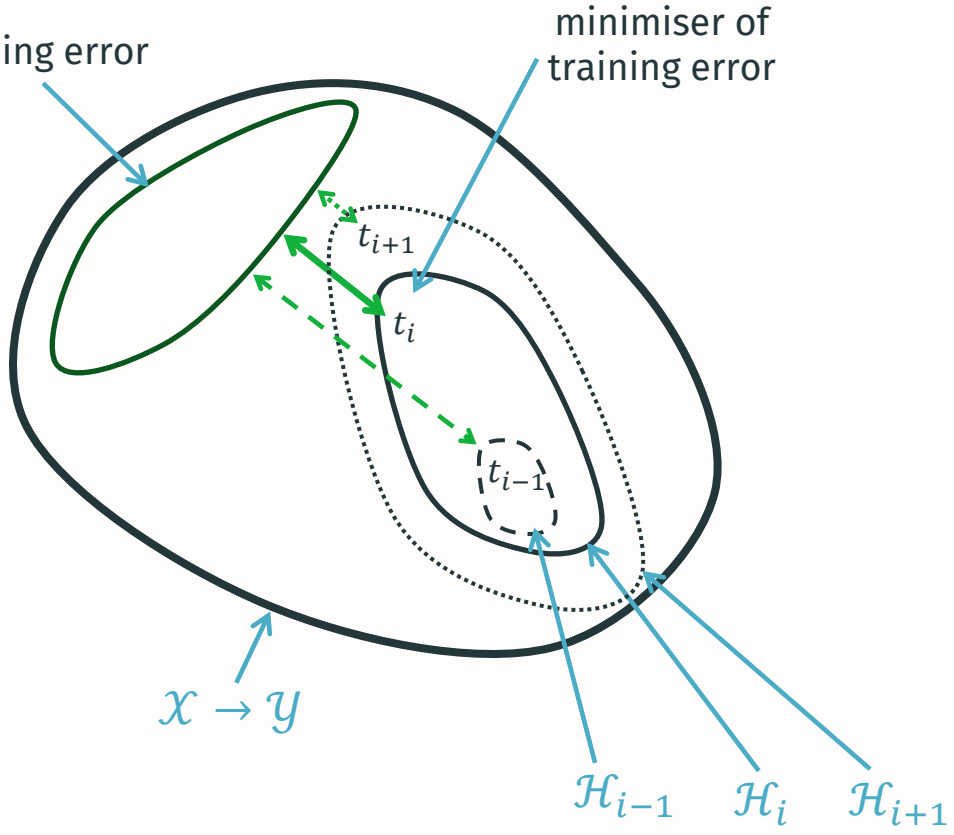




Risk Structure

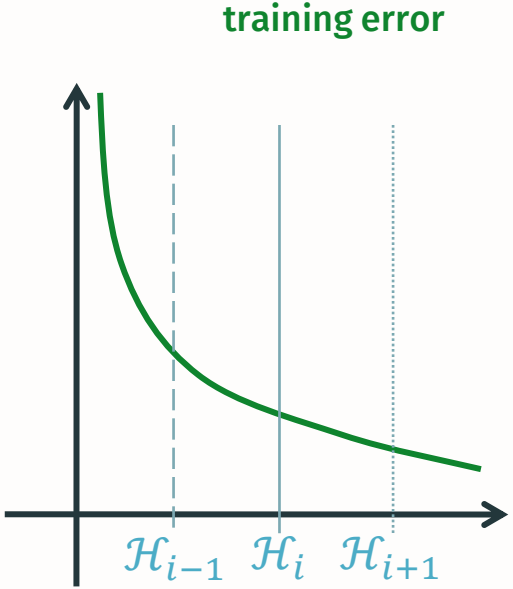
$$\cdots \subseteq \mathcal{H}_{i-1} \subseteq \mathcal{H}_i \subseteq \mathcal{H}_{i+1} \subseteq \cdots \subseteq \mathcal{X} \to \mathcal{Y}$$

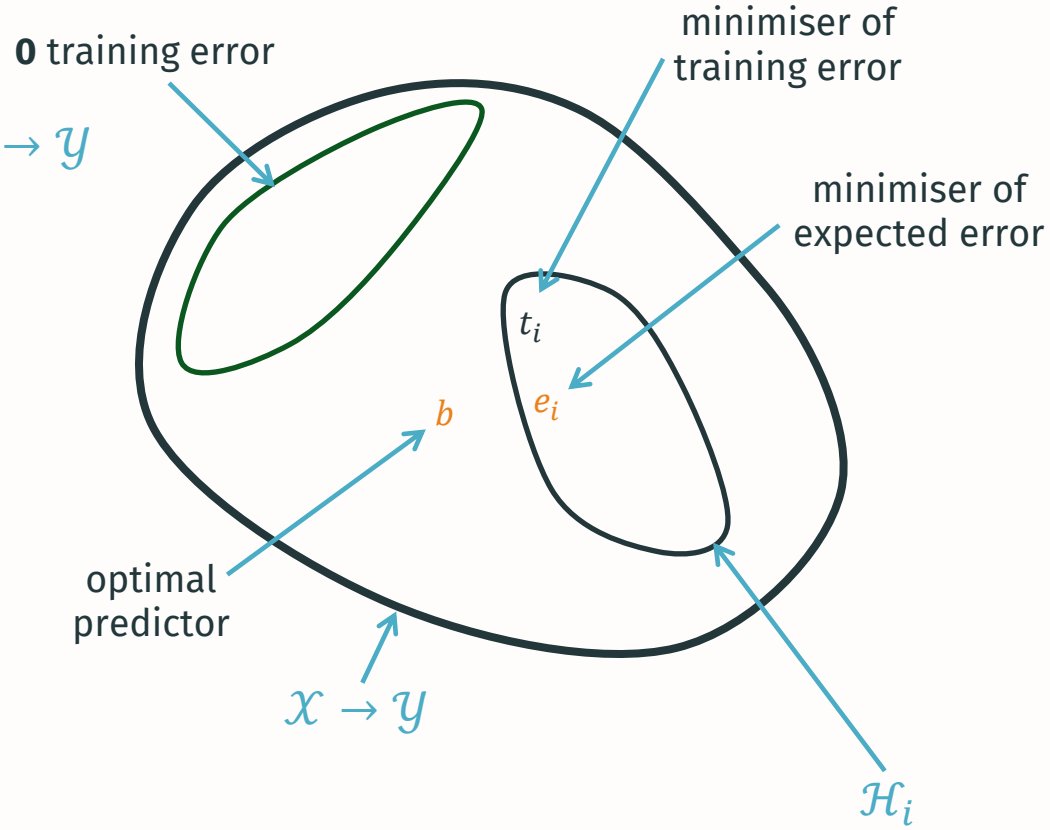




Risk Structure

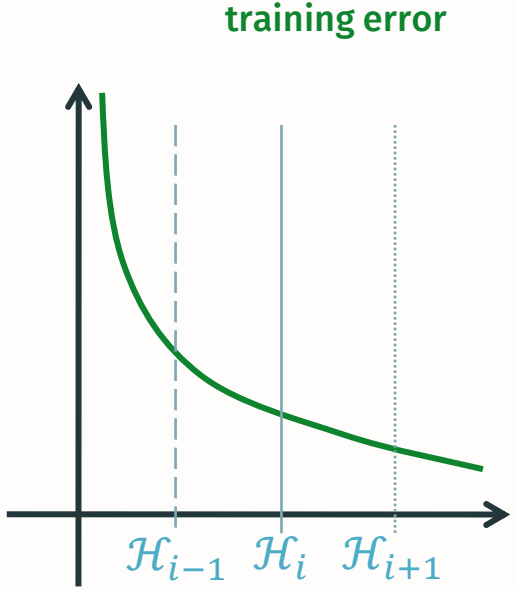
$$\cdots \subseteq \mathcal{H}_{i-1} \subseteq \mathcal{H}_i \subseteq \mathcal{H}_{i+1} \subseteq \cdots \subseteq \mathcal{X} \to \mathcal{Y}$$

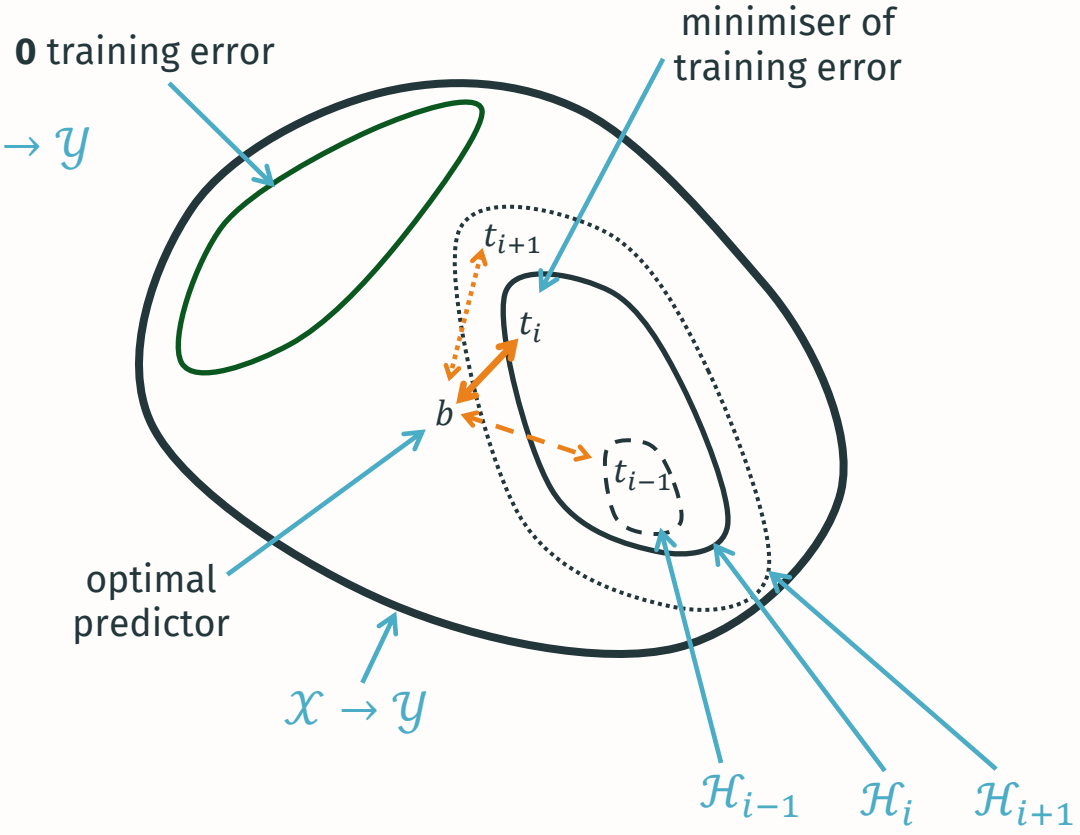




Risk Structure

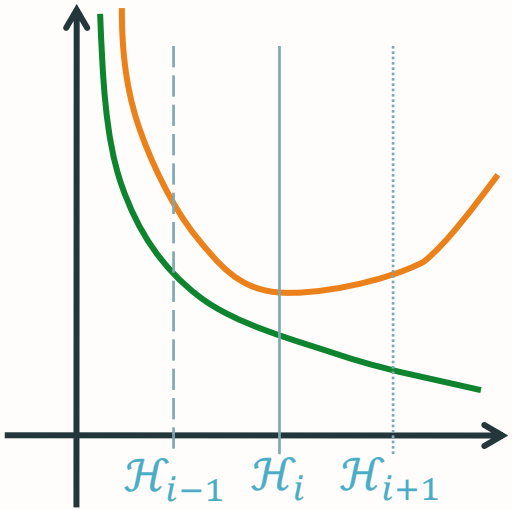
$$\cdots \subseteq \mathcal{H}_{i-1} \subseteq \mathcal{H}_i \subseteq \mathcal{H}_{i+1} \subseteq \cdots \subseteq \mathcal{X} \to \mathcal{Y}$$

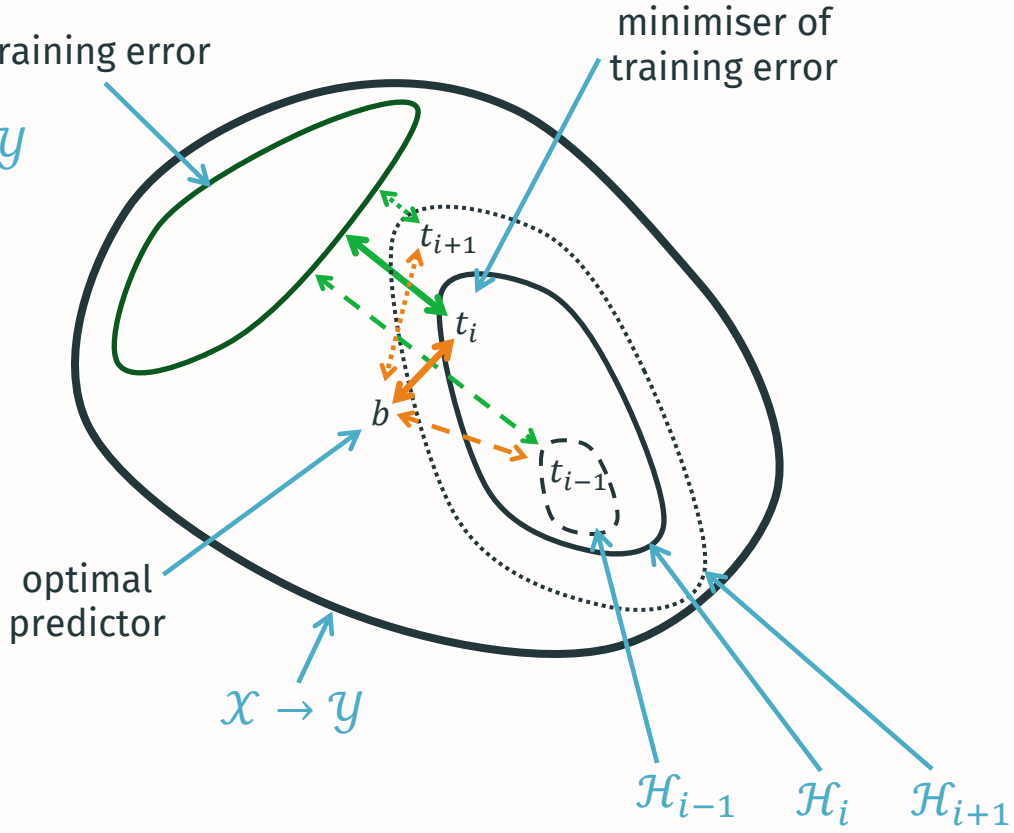




Risk Structure

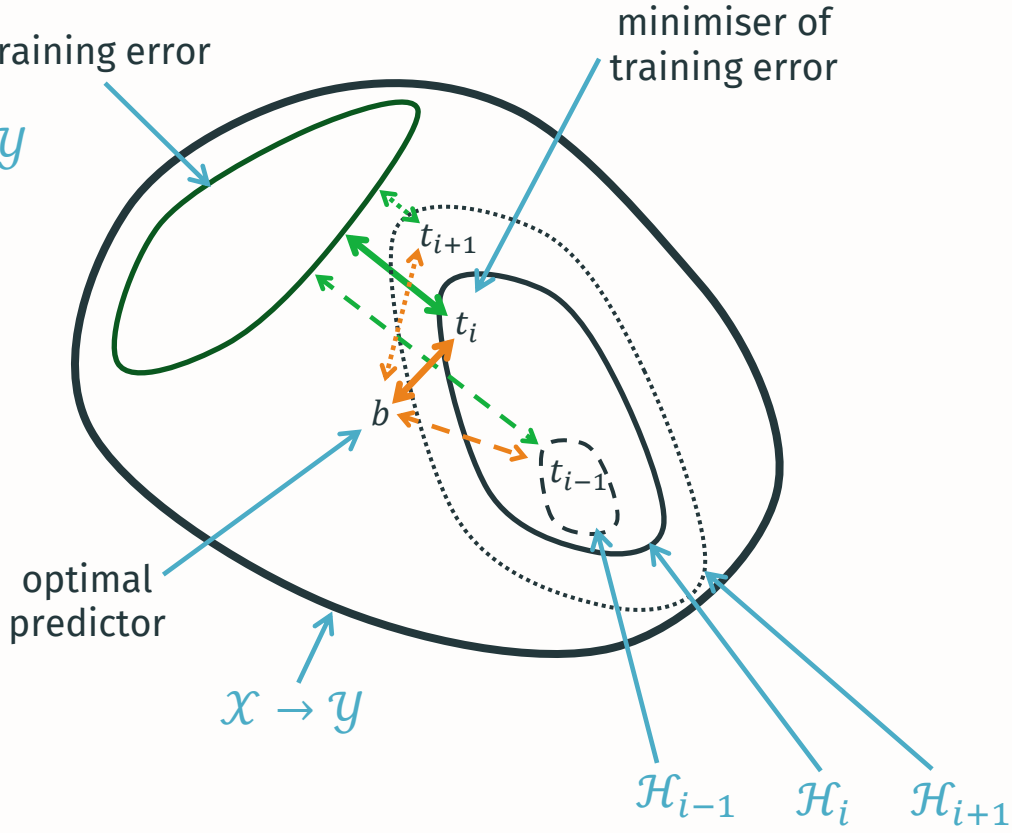
$$\cdots \subseteq \mathcal{H}_{i-1} \subseteq \mathcal{H}_i \subseteq \mathcal{H}_{i+1} \subseteq \cdots \subseteq \mathcal{X} \to \mathcal{Y}$$





Risk Structure

$$\cdots \subseteq \mathcal{H}_{i-1} \subseteq \mathcal{H}_i \subseteq \mathcal{H}_{i+1} \subseteq \cdots \subseteq \mathcal{X} \to \mathcal{Y}$$



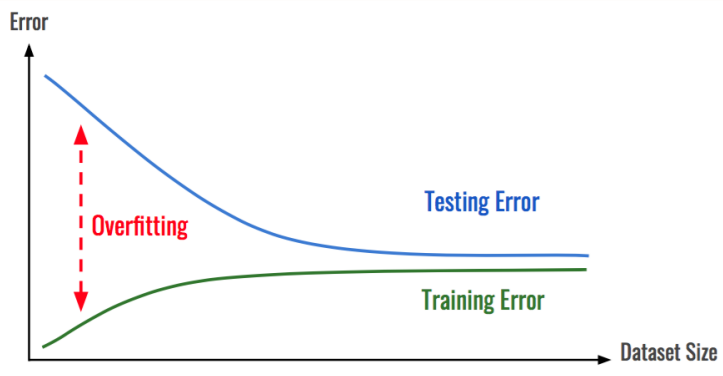
Ways out: Regularisation and/or more data

$$\cdots \subseteq \mathcal{H}_{i-1} \subseteq \mathcal{H}_i \subseteq \mathcal{H}_{i+1} \subseteq \cdots \subseteq \mathcal{X} \to \mathcal{Y}$$

Regularisation (often based on learning theoretic notions of model complexity/capacity)

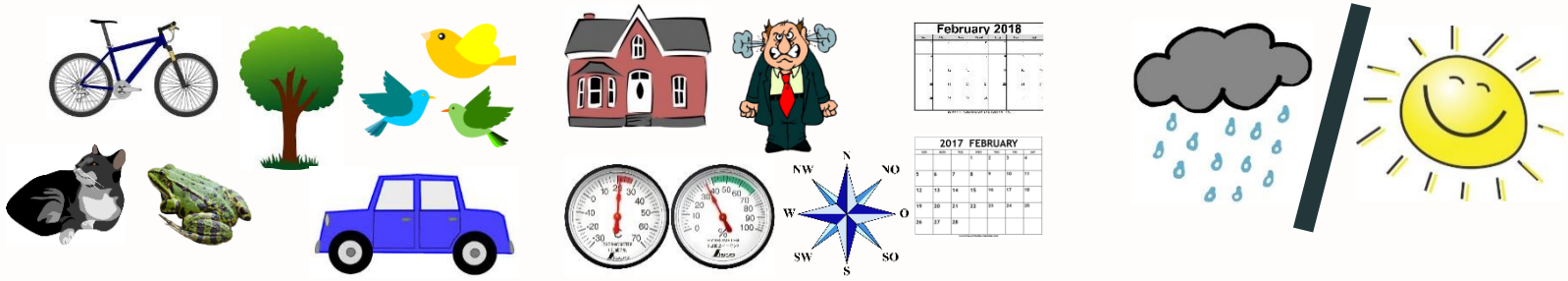
- decision tree pruning
- early stopping
- additive regularisation term
- dropout
- ...

More data

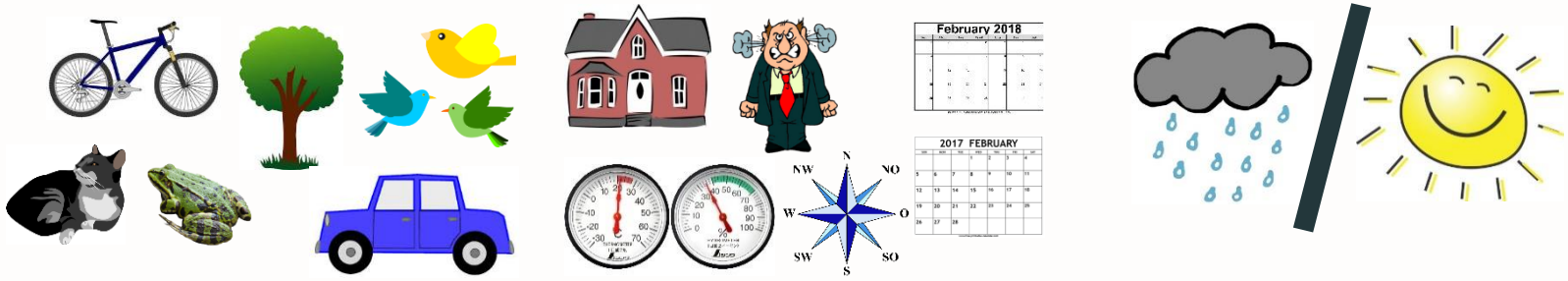


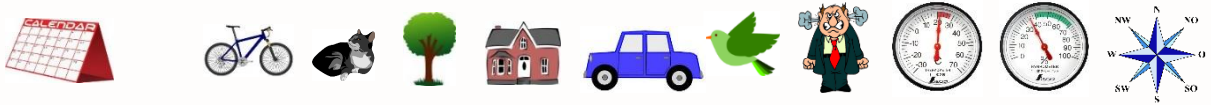
decision trees

Assume we collected a data set ... to predict the weather on our birthday, a week in advance...



Assume we collected a data set ... to predict the weather on our birthday, a week in advance...





thu	2016	✓	✗	✓	✓	✗	✓	✗	14,0	60	N
sat	2017	✗	✓	✓	✓	✓	✗	✗	18,9	40	W
sun	2018	✗	✗	✓	✓	✓	✓	✓	24,1	35	W

Decision Trees form a Hypothesis Class

consist of

- **inner nodes (splitting)**
  - specified by selecting one attribute/feature
  - branch to their children depending on the feature value of an instance
- **outer nodes (decision)**
  - specify the label that we predict for corresponding instances

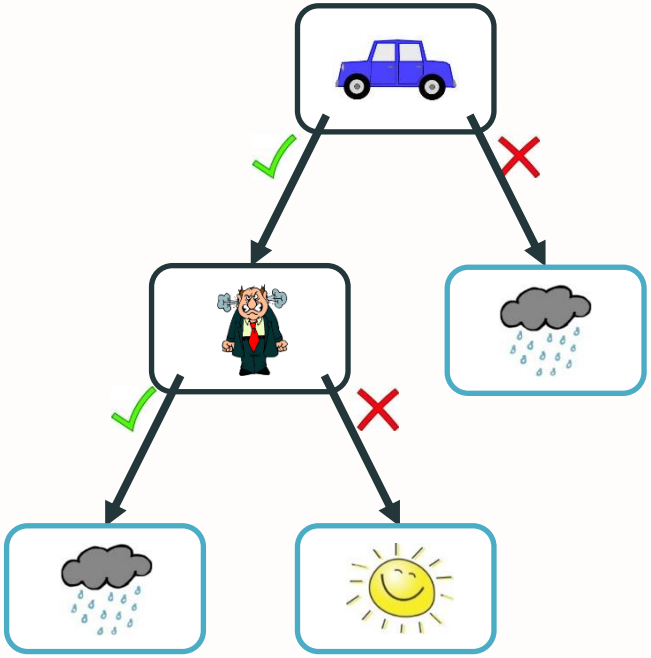
can be used for

- **classification**
- regression
- ranking
- probability

can deal with

- **binary**
- **categorical**
- numerical
- missing
- outliers

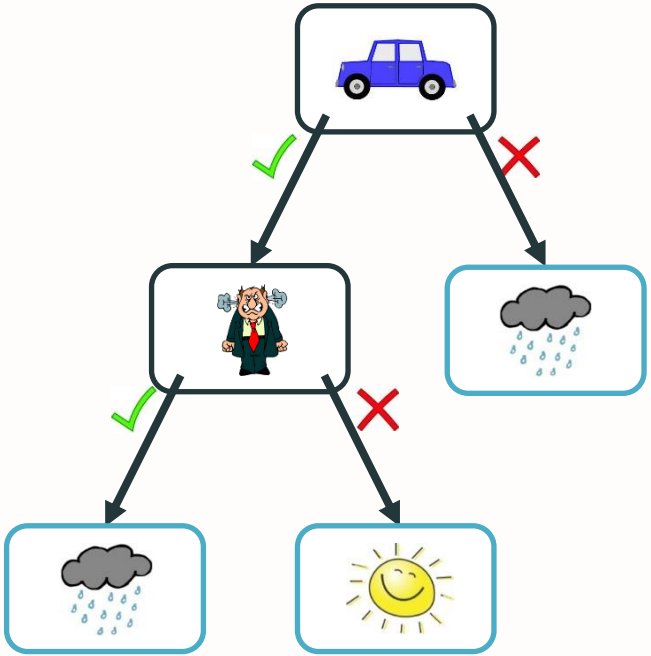
} variables



Decision Trees form a Hypothesis Class

consist of

- **inner nodes (splitting)**
  - specified by selecting one attribute/feature
  - branch to their children depending on the feature value of an instance
- **outer nodes (decision)**
  - specify the label that we predict for corresponding instances



can be used for

- **classification**
- regression
- ranking
- probability

can deal with

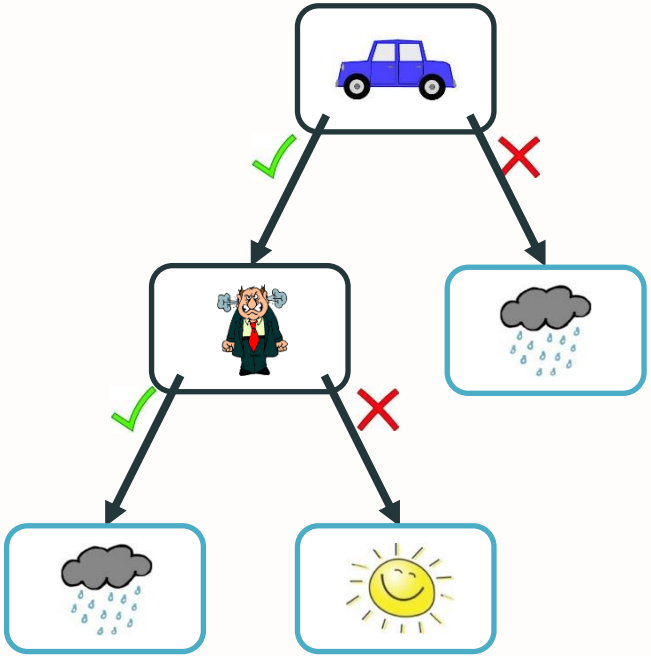
- **binary**
- **categorical**
- numerical
- missing
- outliers

} variables

Decision Trees form a Hypothesis Class

consist of

- **inner nodes (splitting)**
  - specified by selecting one attribute/feature
  - branch to their children depending on the feature value of an instance
- **outer nodes (decision)**
  - specify the label that we predict for corresponding instances



can be used for

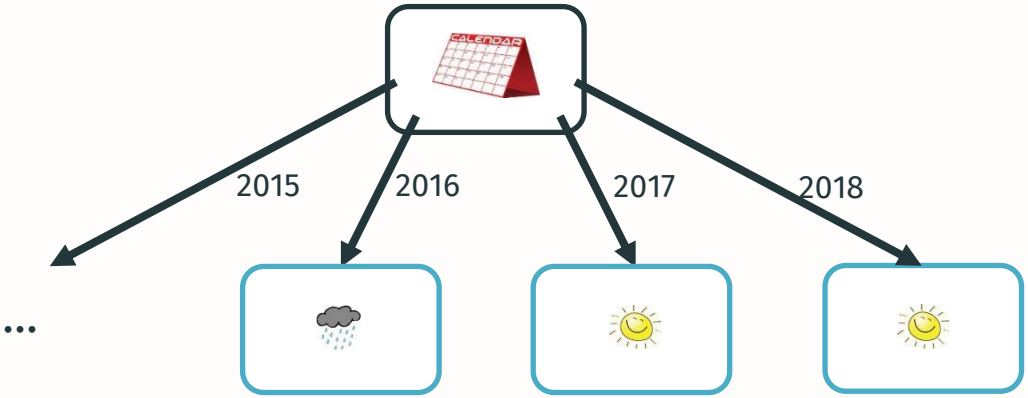
- **classification**
- regression
- ranking
- probability

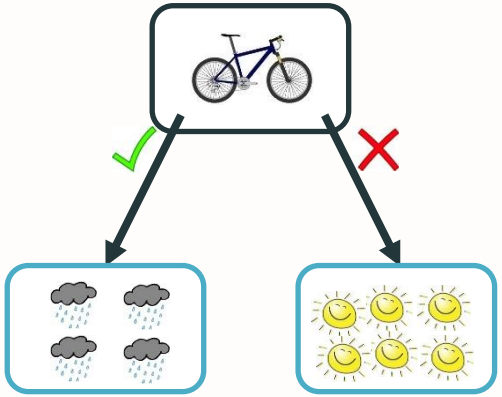
can deal with

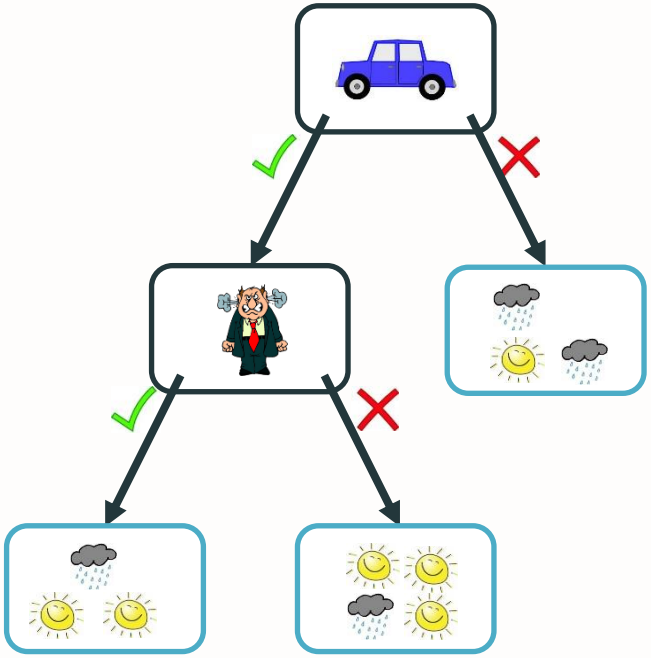
- **binary**
- **categorical**
- numerical
- missing
- outliers

} variables

Decision Trees form a Hypothesis Class







What is a good or even optimal decision tree?
Can we always find a good or even an optimal one?
How long does it take to find a good or even an optimal one?

History of Decision Trees

E.B. Hunt, J. Marin, P.J. Stone, *Experiments in Induction*, 1966

- **top-down induction of decision trees**
- to model human concept learning

A. Fielding. *Binary segmentation: the automatic interaction detector and related techniques for exploring data structure, 1977.*

- for the analysis of survey data

J.R. Quinlan, *Discovering rules by induction from large collections of examples*, 1979.

- information gain heuristic (ID3, later C4.5 J48)

L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and Regression Trees*, 1984.

L. Hyafil and R. Rivest, *Constructing optimal binary decision trees is NP-complete*, 1976.

- considered optimality of prediction time

T. Hancock, T. Jiang, M. Li, and J. Tromp, *Lower bounds on learning decision lists and trees*, 1996.

- finding smallest decision tree is NP-hard

Simple&Generic Top-Down Decision Trees Induction Algorithm

```
grow_tree(D, F):

if stopping_criterion_satisfied(D):
    return new_decision_node(D)

f <- select_feature_for_splitting(D, F)

T <- new_splitting_node()

for each value v of f:
    T.add( grow_tree(D[f==v], F\f)

return T
```

stopping_criterion_satisfied
checks if the tree should be grown further

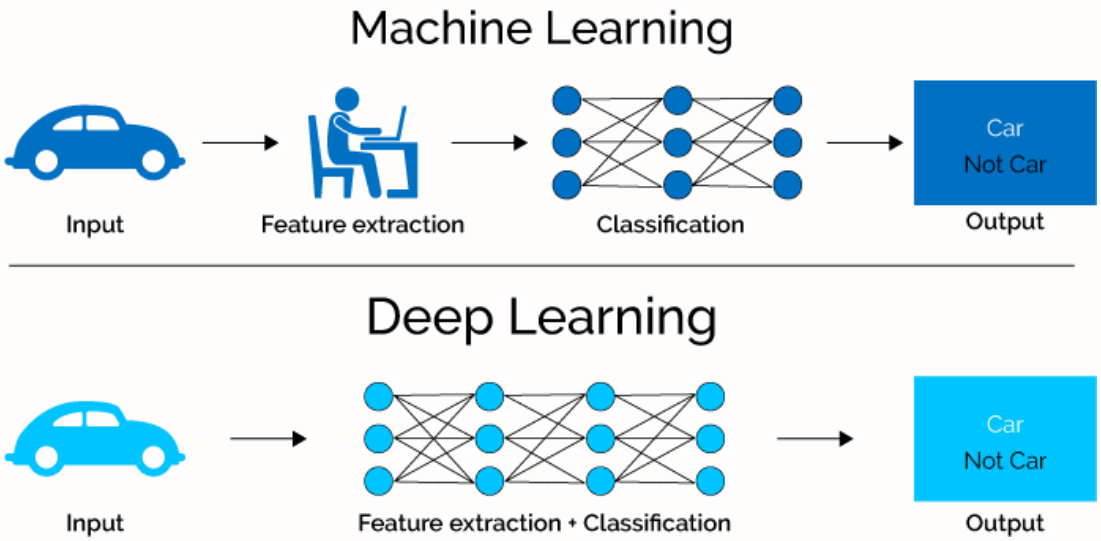
decision_node
analyses what prediction should be made given the remaining data

select_feature_for_splitting
decide which feature to break the dataset down by next

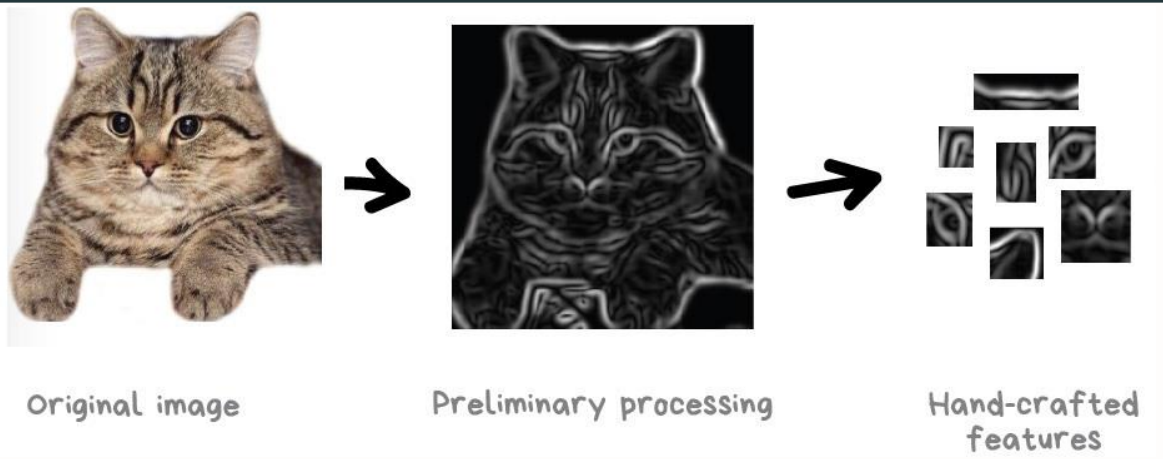
new_splitting_node
creates a new inner node

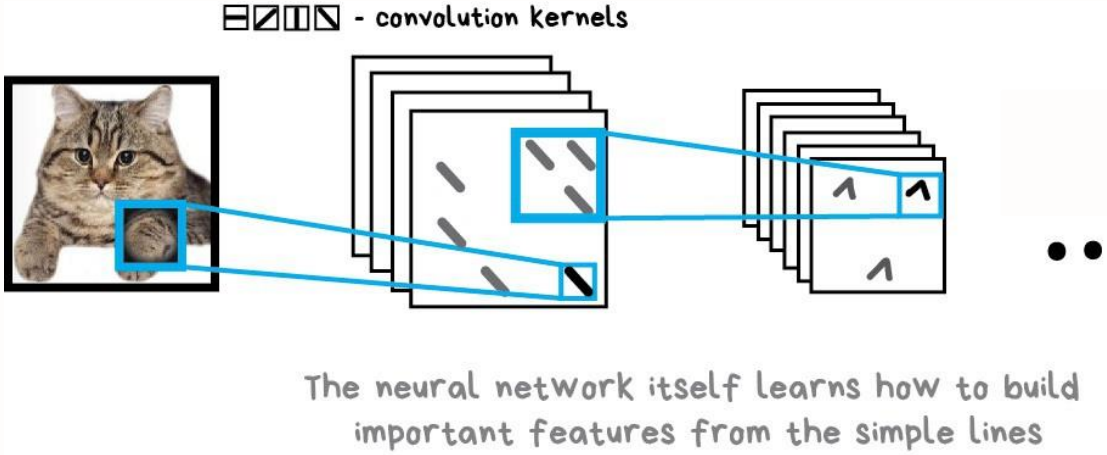
add
grows a child of the current node

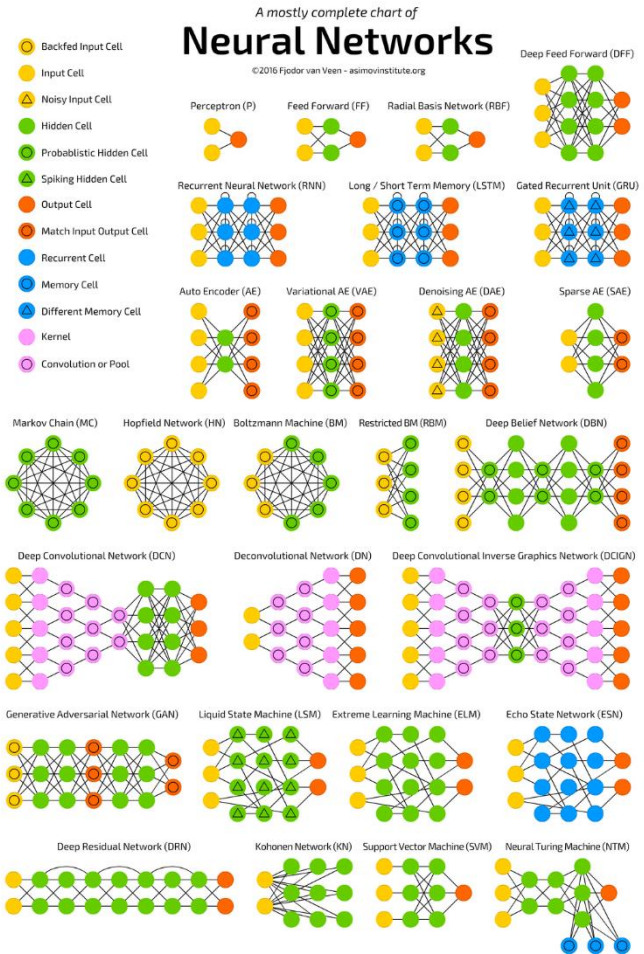
Deep Learning



Feature Generation for Neural Networks



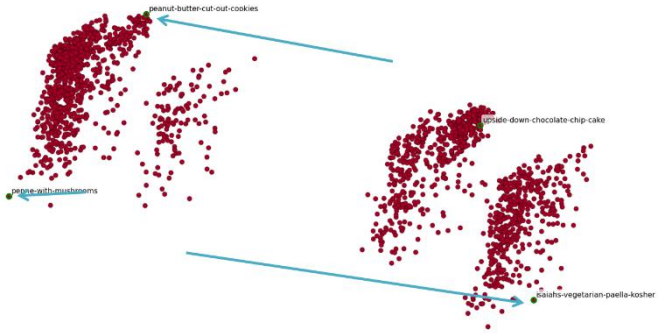




what we do

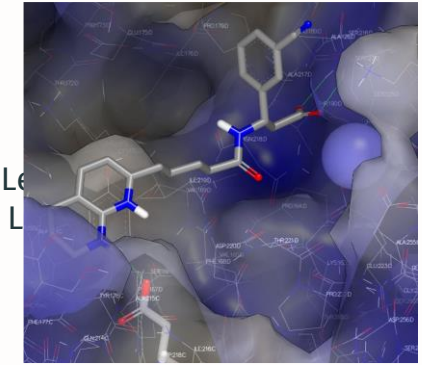
Interactive Machine Learning

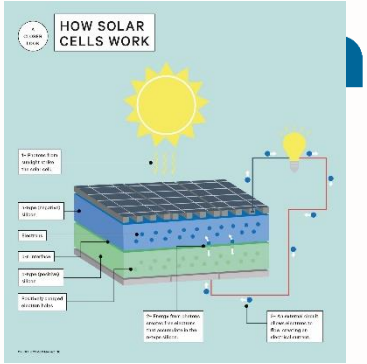
@Technische Universität Wien, 9. 2. 2019

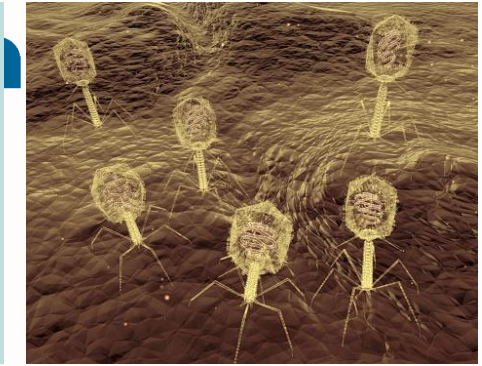


Thomas Gärtner

Universitätsprofessor for Machine Le
Head of the research unit Machine L
TU Wien Informatics







Dynamic Difficulty Adjustment [nips'11, cig'11]

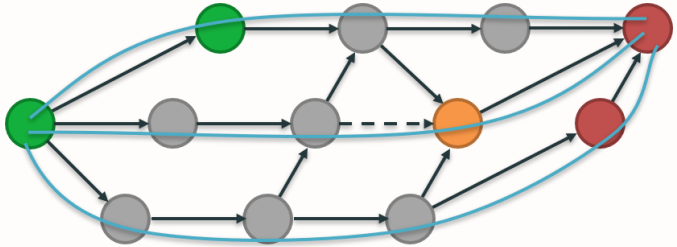
1) **game master predicts** a difficulty setting

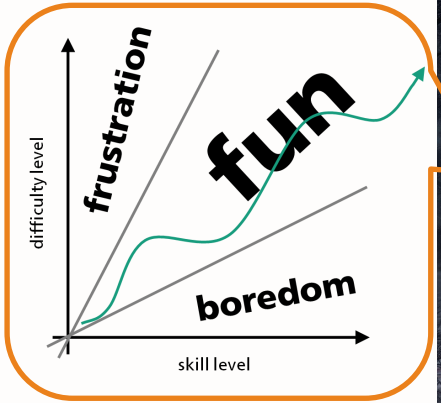
2) **player plays** one round of the game in this setting

3) **game master observes** whether the setting was too easy, just right, or too difficult

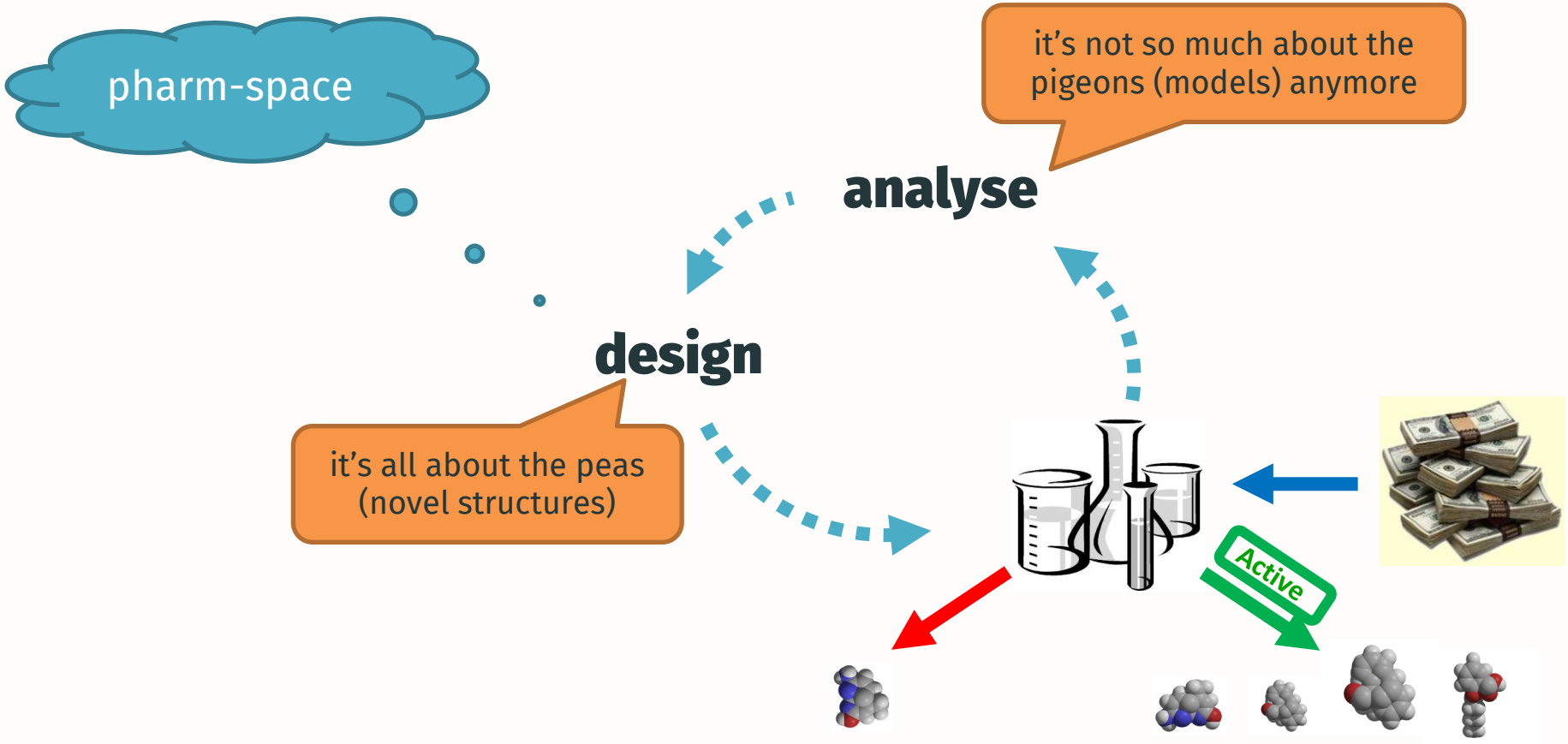
this game is played on a partially ordered set reflecting the **`is more difficult'-relation** the goal of the game master is to predict `just right'-settings as often as possible

we can bound the number of mistakes in this setting





de novo Drug Design



Cyclic Discovery Processes – algorithmic challenges

design novel candidate structures

huge, **intensionally** specified design space of **structures**

design

analyse

build a hypothesis of the properties of all structures from the design space

to improve designs

cope with the cyclic nature

cope with the huge candidate space

aim at general-purpose algorithms

interface with domain experts

make

test

realise designed candidate structures in their context

determine properties of realised structures in their context

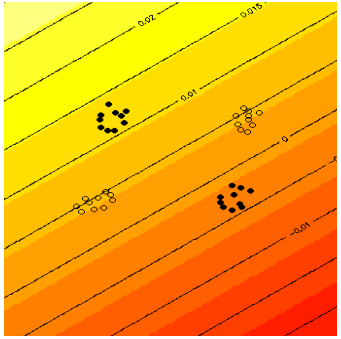
Kernel Methods, e.g., Support Vector Machines (SVM)

find linear separator in Hypothesis space (reproducing kernel Hilbert space)
with small norm (-> maximally far from all labelled instances, robustness against measurement noise)
decision boundary in input space depends on kernel function

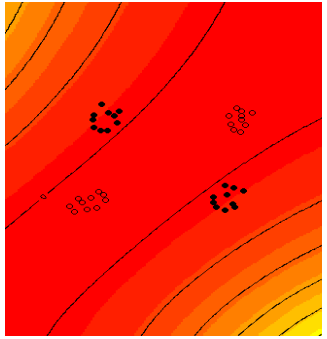
challenges for kernel design are

- **validity** (positive definiteness),
- **efficiency** (polynomial time), and
- **effectiveness** (good generalisation possible)

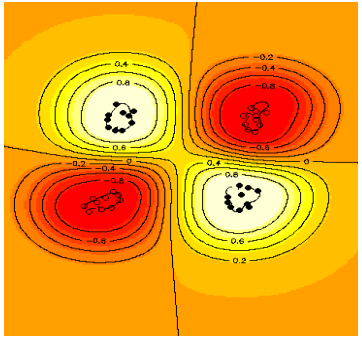
linear



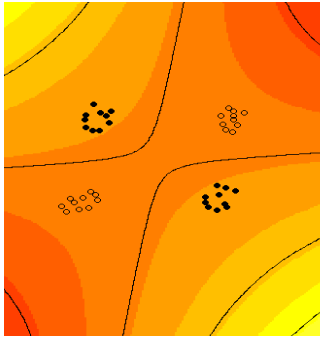
sigmoid



Gaussian



polynomial



Kernels for Structured Data

Kernels for Basic Terms [icml'02, mlj'04]

idea: data as terms in typped higher-order logic; type definition gives kernel % modifiers

theoretical results:

- generally valid (psd) & efficient
- effective for problem classes like MIL

Graph Kernels [colt'03,kdd'04]

idea: define kernel as some function of the **common subgraphs** (cf strings, trees, ...)

but computing graph kernels is hard

for efficient computation consider

- particular graphs-classes
- different embedding operators, e.g., homomorphism

Structured Prediction [mlj'09, uai'09]

previous approaches turn out to be computationally hard for graphs

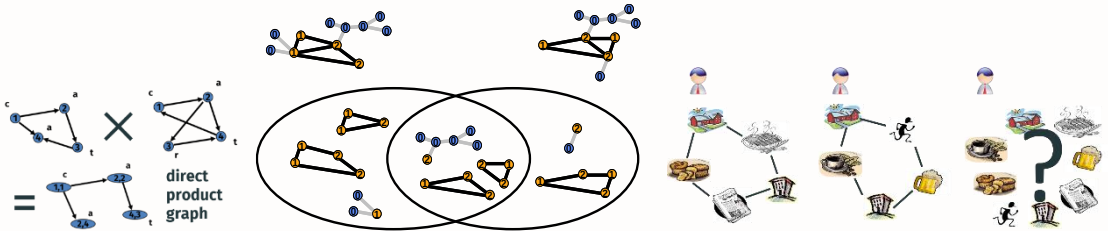
instead

$$\sum_{s \in S} \langle \phi(x,s), h \rangle \quad \text{and} \quad \sum_{s \in S} \langle \phi(x,s), h \rangle^2$$

can often be **computed efficiently** and

$$\sum_{s \in S} exp(\langle \phi(x,s), h \rangle)$$

can often be **approximated efficiently**



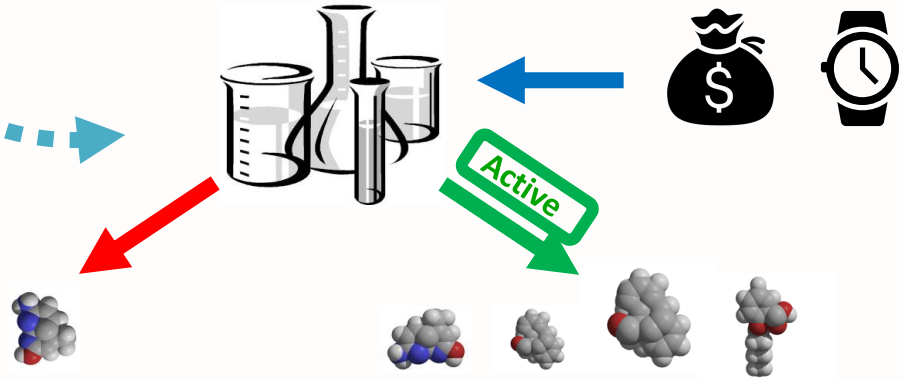
de novo Drug Design [aaai'17]

weighted maximum a posteriori

$$\theta_t = \operatorname*{argmax}_{\theta \in \mathcal{H}} p(Y \mid X, \theta) p(\theta)$$

sample

$$x_t \sim p(x \mid y^*, \theta_t)$$



de novo Drug Design [aaai'17]

pharm-space

sample

$x \sim Q \quad u \sim \mathcal{U}[0,1]$

$$u < \frac{p(y^* \mid x, \theta)}{p(y^* \mid x_t, \theta)}$$

$$\Downarrow$$

$x_t \leftarrow x$

until converged

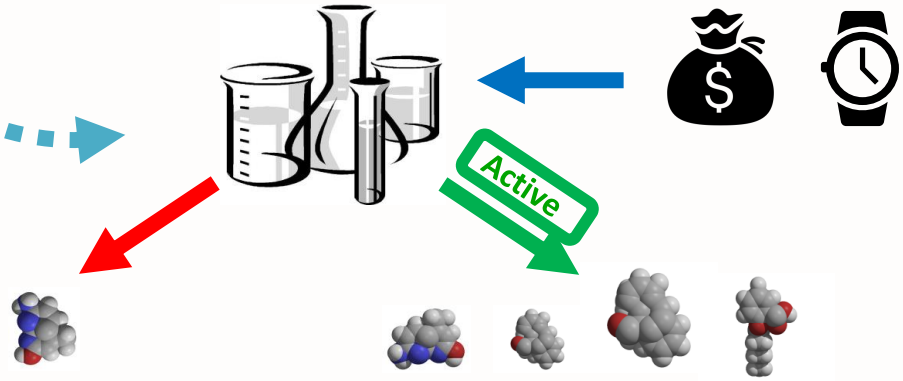
Metropolis-Hastings

sample

$x_t \sim p(x \mid y^*, \theta_t)$

weighted maximum a posteriori

$$\theta_t = \operatorname*{argmax}_{\theta \in \mathcal{H}} p(Y \mid X, \theta) p(\theta)$$

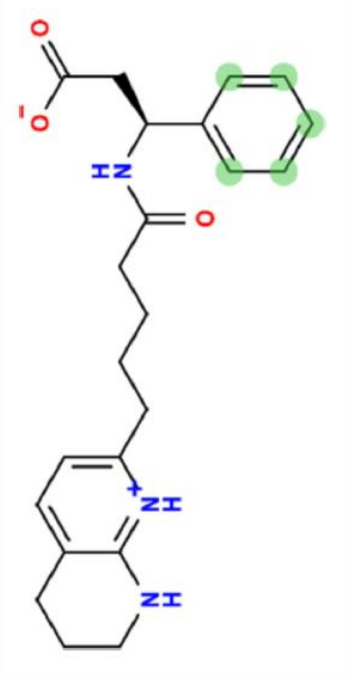


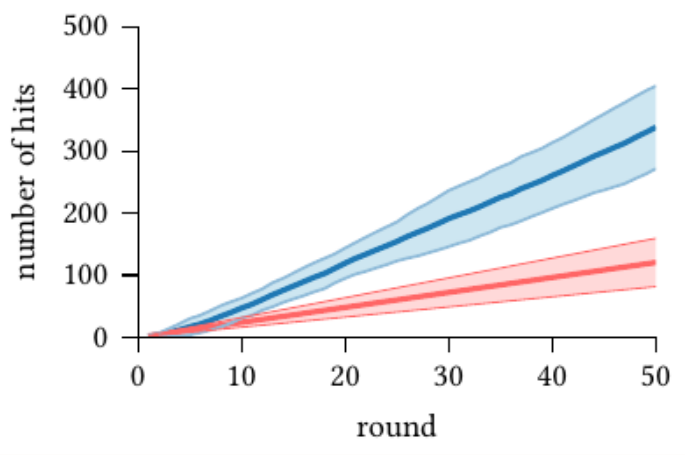
Designing Drugs: Simulations [aaai'17] & Idiopathic Pulmonary Fibrosis [mi'18]

non-Hamiltonian graphs

greedy (baseline)

active search





discovered 19 out of the 24 active compounds which are known to be active from previous biological assays!

Speeding up machine learning

use the structure of the data to speed up optimisation [nips'06]

low-dimensional embeddings:

- Boost the low-dimensional representation for shift-invariant kernels [nips'16]
- Prove the effectiveness of *k*-means++ (sampling) for Nyström landmarks [icml'17]
- Learning with non-positive definite kernels (Kreĭn spaces [icml'18]) [icml'19?]

a logarithmic number of features is often sufficient(!)

parallelise:

- **Guaranteed same performance** as a non-parallel base-learning algorithm in polylogarithmic time using more data and many computational cores [nips'17]
  (cf Nick's Class for decision problems)

$$t \in O\left(\log_2^k(2\mathbf{r}) + h\mathbf{r}^3\right) \qquad T \in O\left(2^{hk}\right)$$

Moore's Law

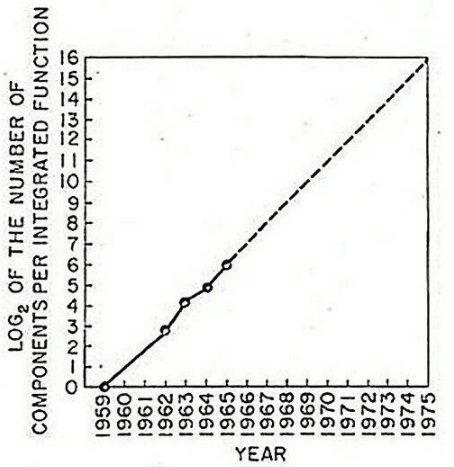
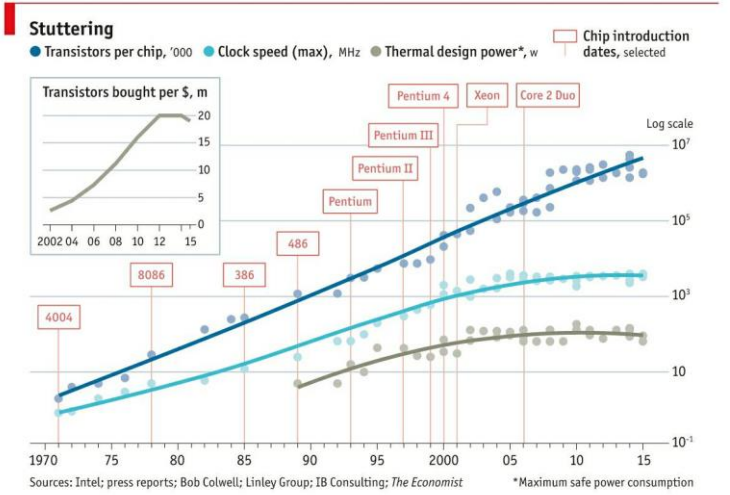
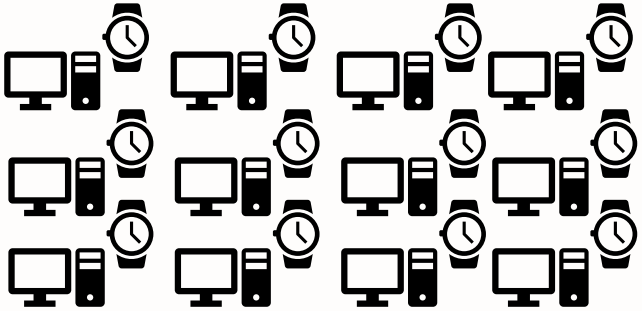


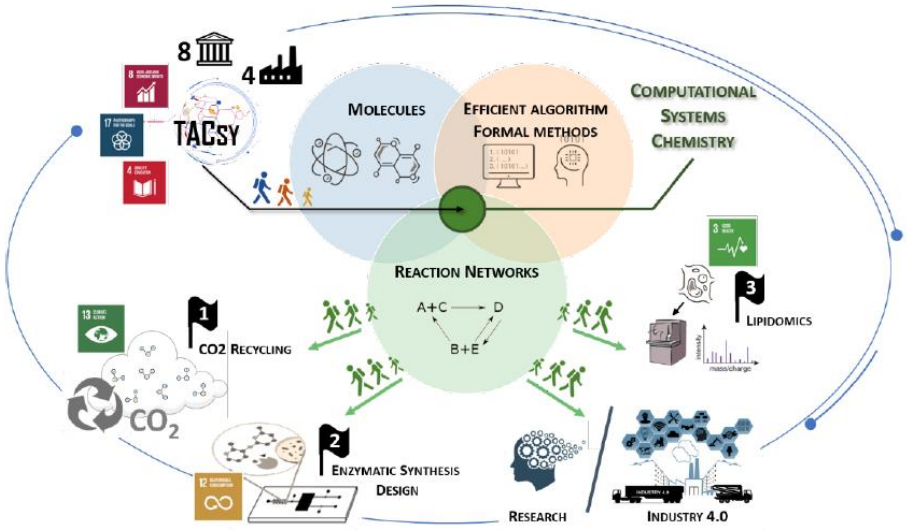
Fig. 2. Number of components per integrated function for minimum cost per component extrapolated vs time.

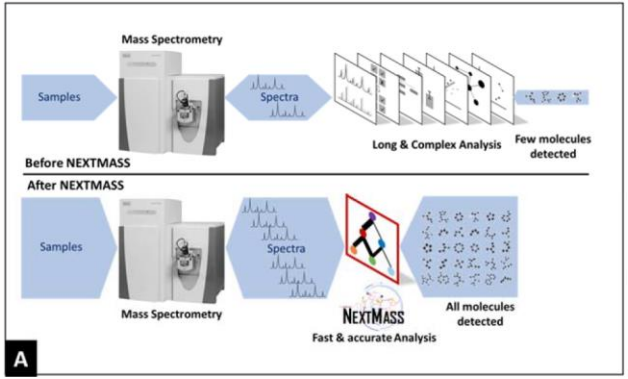


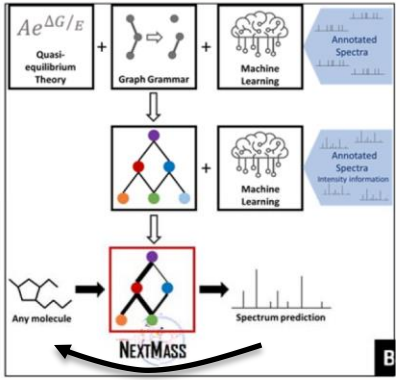


ML for Computational Systems Chemistry / Mass Spectroscopy

under submission







University of Nottingham
UK | CHINA | MALAYSIA

EMBL
European Molecular Biology Laboratory

UNIVERSITÄT LEIPZIG

SDU
UNIVERSITY OF SOUTHERN DENMARK

universität wien

ThermoFisher SCIENTIFIC

BASF
We create chemistry

FLUIGENT
SMART MICROFLUIDICS

FRIEDRICH-SCHILLER-UNIVERSITÄT JENA

The University Of Sheffield.

HARVARD MEDICAL SCHOOL

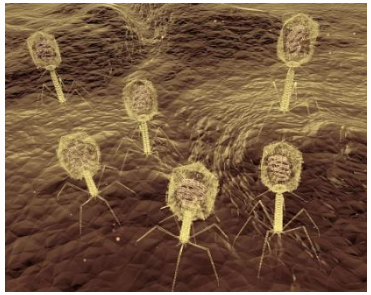
HÖGSKOLAN I BORÅS

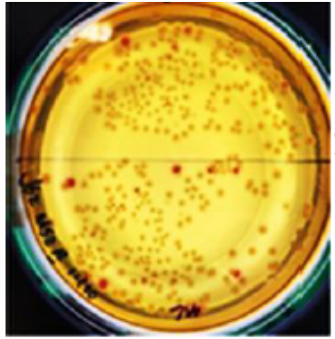
ML for the design of Bacteriophages

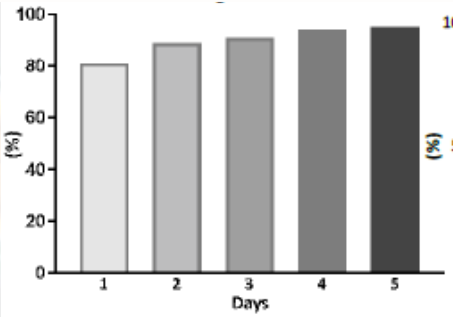
in vivo phage activity

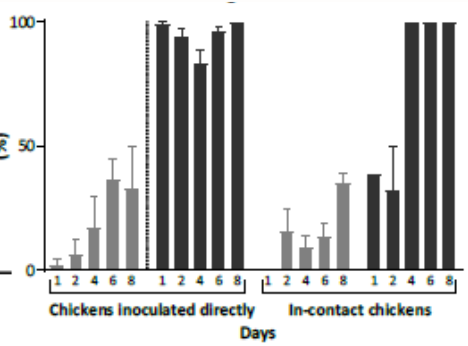
(i) reduces the numbers of bacteria possessing AMR plasmids
(ii) selects for complete AMR plasmid loss and
(iii) selects for loss of AMR plasmid transmissibility

under submission









ML & Simulation for the Active Design of Solar Cells

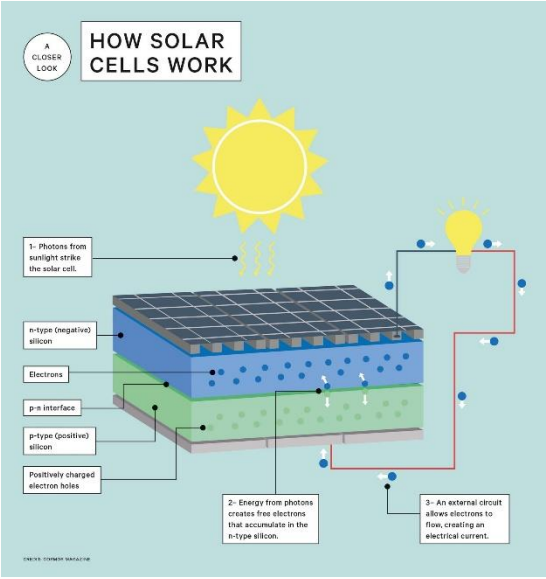
Active search over measurement parameters of device simulator and *real device* to reveal inefficiency of the current real device

Active optimisation over fabrication, structure, and material parameters of *real devices* to find more efficient devices

MERCK

Imperial College London

in development



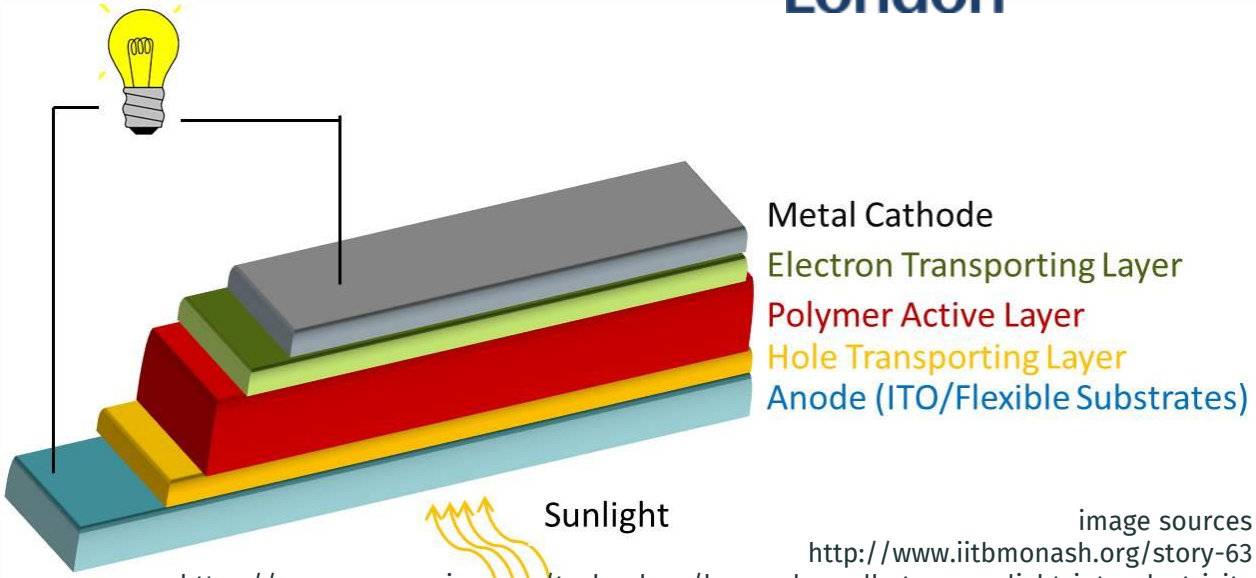
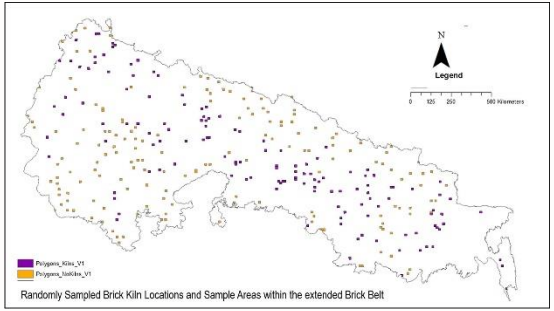
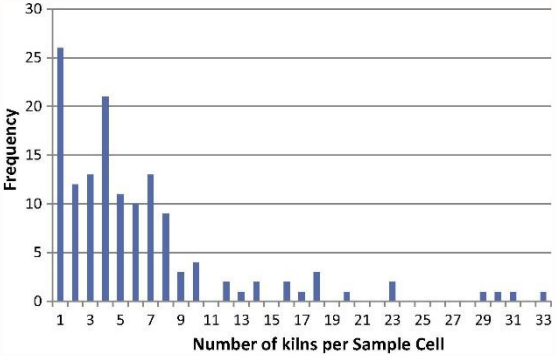


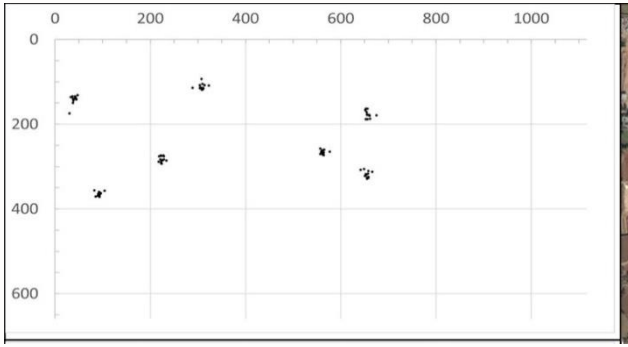
image sources
http://www.iitbmonash.org/story-63
https://cosmosmagazine.com/technology/how-solar-cells-turn-sunlight-into-electricity

Active Search for Modern Slavery Sites (eg Brick Kilns)





Aim: use active search to minimise the amount of manual expert supervision needed



appendix

Radon Points

a **bipartition** of a set is a pair of disjoint subsets

a **Radon partition** of a set is a bipartition whose convex hulls overlap

a **Radon point** of a set is a point in this overlap

the **Radon number** r is the smallest size such that any set of that size has a Radon partition (for Euclidean spaces $\mathbb{R}^d$ Radon's Theorem shows that $r = d + 2$)

Radon Points

a **bipartition** of a set is a pair of disjoint subsets

a **Radon partition** of a set is a bipartition whose convex hulls overlap

a **Radon point** of a set is a point in this overlap

the **Radon number** r is the smallest size such that any set of that size has a Radon partition (for Euclidean spaces $\mathbb{R}^d$ Radon's Theorem shows that $r = d + 2$)

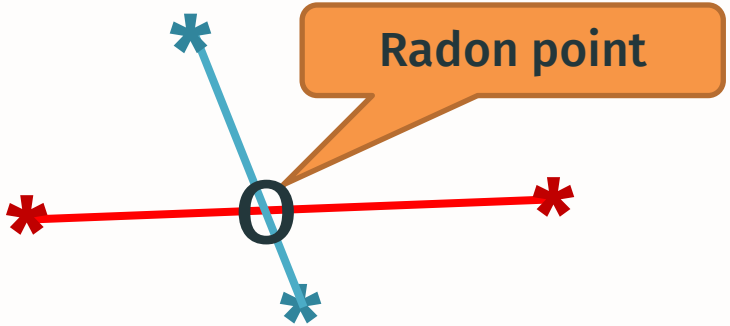
Radon Points

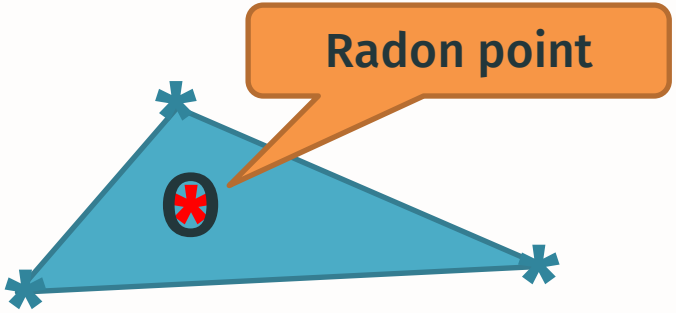
a **bipartition** of a set is a pair of disjoint subsets

a **Radon partition** of a set is a bipartition whose convex hulls overlap

a **Radon point** of a set is a point in this overlap

the **Radon number** r is the smallest size such that any set of that size has a Radon partition (for Euclidean spaces $\mathbb{R}^d$ Radon's Theorem shows that $r = d + 2$)





Radon Points

a **bipartition** of a set is a pair of disjoint subsets

a **Radon partition** of a set is a bipartition whose convex hulls overlap

a **Radon point** of a set is a point in this overlap

the **Radon number** r is the smallest size such that any set of that size has a Radon partition
(for Euclidean spaces $\mathbb{R}^d$ Radon's Theorem shows that $r = d + 2$)

for any (quasi-)convex function q and any set S,
the **Radon point** of S has a value of q that is **no larger** than two points from S

Consistency of Radon Points

$$P\left[q\left(\quad X \quad\right) > \epsilon\right] \quad \leq \delta$$

Consistency of Radon Points

$$P[q(R) > \epsilon] \leq (r\, P[q(X) > \epsilon])^2$$

Consistency of Radon Points

$$P\left[q\left(f^{(0)}\right) > \epsilon\right] \leq \delta$$

$$P\left[q\left(f^{(h)}\right) > \epsilon\right] \leq \left(r\, P\left[q\left(f^{(h-1)}\right) > \epsilon\right]\right)^{2}$$

$$P\left[q\left(f^{(h)}\right) > \epsilon\right] \leq \left(r\, P\left[q\left(f^{(0)}\right) > \epsilon\right]\right)^{2^h} \leq (r\delta)^{2^h} = \Delta$$

what is the sequential runtime complexity of achieving the same bound Δ with the original algorithm?

Time and Sample Complexity

given a fixed confidence target and constant precision parameter

$$\delta = \frac{1}{2\mathbf{r}}; \quad \Delta = (\mathbf{r}\delta)^{2^h} = 2^{-2^h} \qquad m(\epsilon, \delta) = \alpha_\epsilon + \beta_\epsilon \log_2 \frac{1}{\delta} \qquad t(m) \in O\left(m^k\right)$$

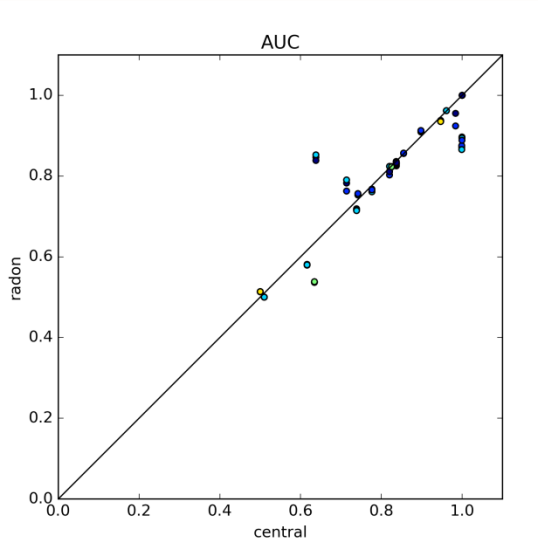
single centralised machine

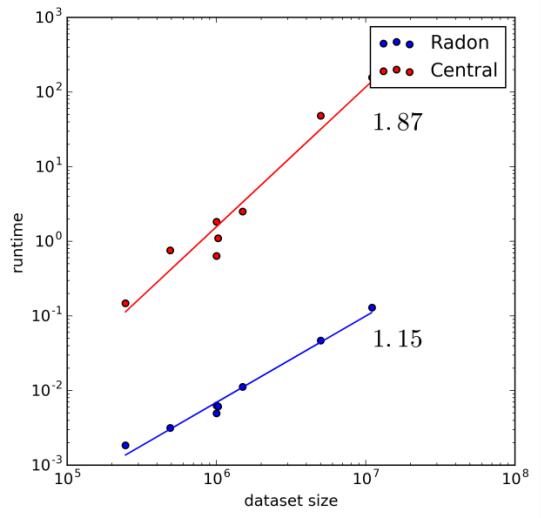
$$M = \alpha_\epsilon + \beta_\epsilon 2^h; \quad T \in O\left(2^{hk}\right)$$

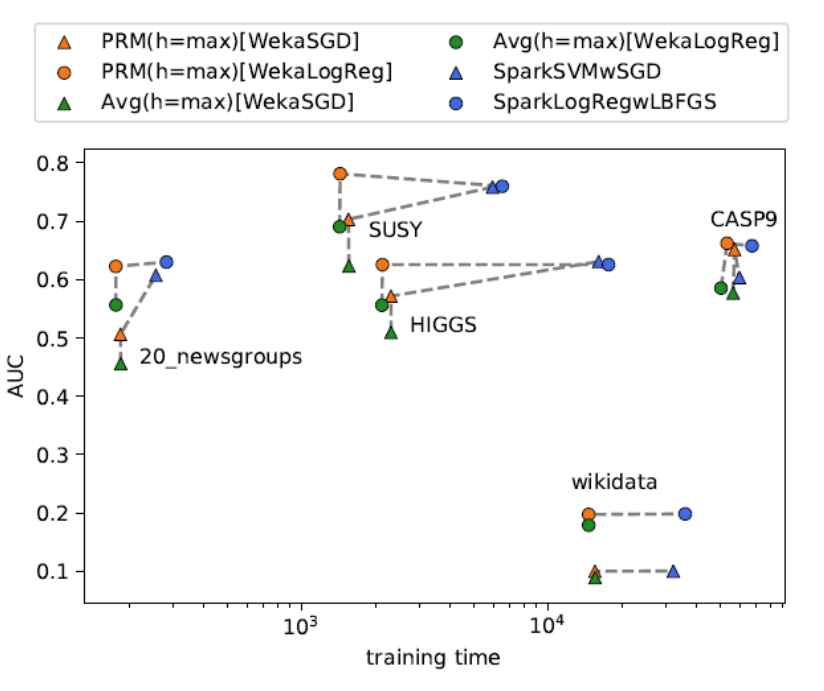
parallel Radon machine

$$m = \alpha_\epsilon + \beta_\epsilon \log_2(2\mathbf{r}); \quad t \in O\left(\log_2^k(2\mathbf{r}) + h\mathbf{r}^3\right)$$

Empirical Simulation Study—Linear SVMs







Thank you

design novel candidate structures

huge, intensionally specified design space of structures

design

analyse

build a hypothesis of the properties of all structures from the design space

to improve designs

cope with the cyclic nature

cope with the huge candidate space

aim at general-purpose algorithms

interface with domain experts

make

test

realise designed candidate structures in their context

determine properties of realised structures in their context

Acknowledgements

collaborators:
Prof. Peter Flach, Prof. John W. Lloyd, Prof. Alex Smola, Prof. Stefan Wrobel, Prof. Tobias Scheffer, Prof. Jürgen Bajorath, Prof. SVN Vishwanathan, Prof. Kristian Kersting, Prof. Stephane Canu, Prof. Jan Ramon, Prof. Gilles Gasso, Prof. Kurt Driessens, Prof. Quoc V. Le, Prof. Jonathan D. Hirst Simon J. F. Macdonald, Thomas Mcinally, Dr. Karina Zapien, Dr. Claudio Lucchese, Dr. Hanna Geppert, Dr. Dagmar Stumpfe, Dr. Yasmin Altun, Dr. Gemma C. Garriga, Dr. Adam Kowalczyk, Dr. Sandy Moens, Simon Burton, Steven A. Oatley,

group & alumni:
Prof. Ulf Brefeld, Prof. Roman Garnett, Dr. Mario Boley, Dr. Tamas Horvath, Dr. Shankar Vembu, Dr. Olana Missura, Dr. Daniel Paurat, Dr. Nicolò Navarin, Michael Kamp, Bo Kang, Dino Oglic, Katrin Ullrich, Jens Humrich, Laurentiu Ilici , Jiojian Wang, Joeseph Redshaw, Alexe Haywood

Emmy Noether-Programm
Deutsche Forschungsgemeinschaft DFG

University of Nottingham
UK | CHINA | MALAYSIA

Fraunhofer IAIS

universitätbonn
RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT

gsk

appendix

Machine Learning Lab Structure

Machine Learning research for applications in Chemistry

Machine Learning research for applications in Materials

basic/fundamental **Machine Learning** research

active learning
online learning
structured data
parallel learning
learning theory
...

Machine Learning research for applications in Humanities

Machine Learning research for applications in _____

Machine Learning as a service

Machine Learning tool development

Further International Collaborations and Standing Indicators

ongoing

previous

Program Co-Chair ECML/PKDD'18

Journal-Track Co-Chair ECML/PKDD'16

Workshop Co-Chair CML'13, '15, '16; MLG '06

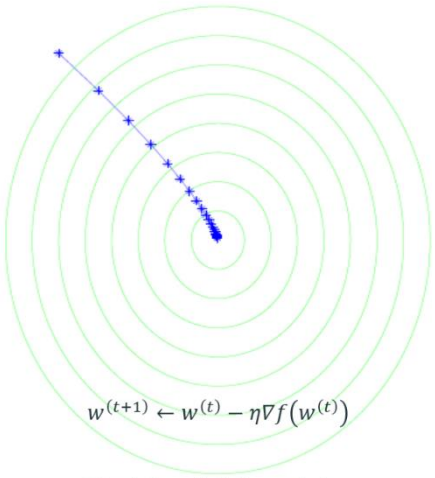
Best Paper/Reviewer Awards

Area Chair/SPC
IJCAI'19
NIPS'19, '18, '17, '15
ECML/PKDD'19, '15, '14, '13

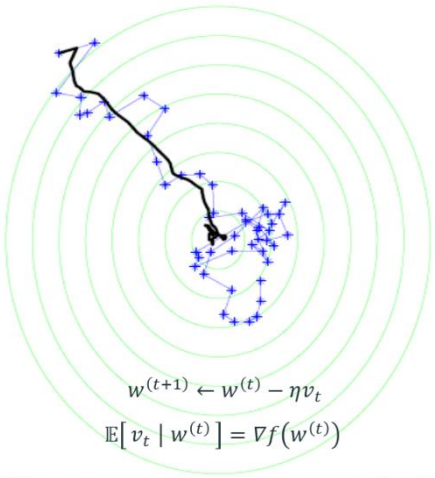
Editor for **Machine Learning**

appendix

Embrace the randomness



GD: deterministic updates
in the direction of the gradient



SGD: random updates whose expectation is
in the direction of the gradient

source: Understanding Machine Learning by Shai Ben-David and Shai Shalev-Shwartz

Funding opportunities

- FWF Einzelprojekte
- FWF Forschungsgruppen
- FWF SFB
- EU FET-open
- ERC AdG

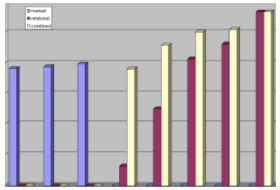
Kernels for Structured Data

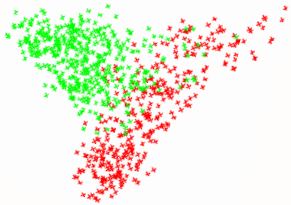
Kernels for Basic Terms [icml'02, mlj'04]

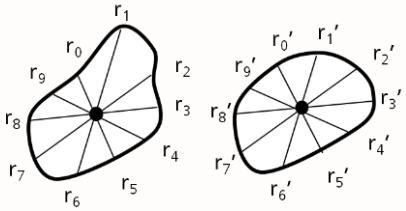
idea: data as terms in typped higher-order logic; type definition gives kernel % modifiers

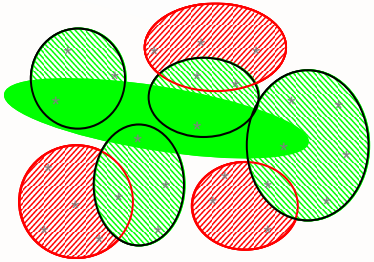
theoretical results:

- generally valid (psd) & efficient
- effective for problem classes like MIL









$(c_I : \mathcal{X} \to \mathcal{Y}) \leftrightarrow (c_{MI} : 2^{\mathcal{X}} \to \mathcal{Y})$ $\quad \mathcal{H}_I \leftrightarrow \mathcal{H}_{MI}$ $\quad k_I \leftrightarrow k_{MI}$

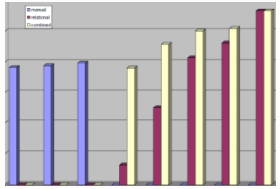
Kernels for Structured Data

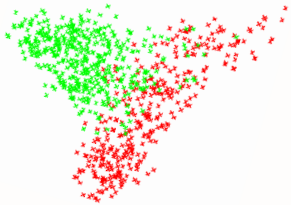
Kernels for Basic Terms [icml'02, mlj'04]

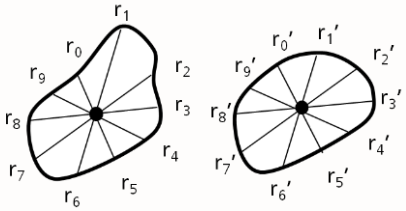
idea: data as terms in typped higher-order logic; type definition gives kernel % modifiers

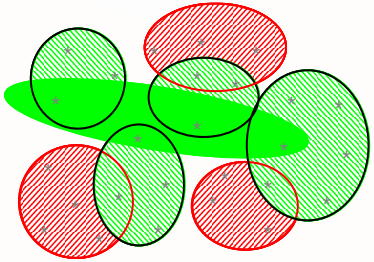
theoretical results:

- generally valid (psd) & efficient
- effective for problem classes like MIL

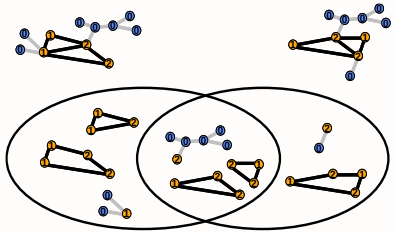








$(c_I : \mathcal{X} \to \mathcal{Y}) \leftrightarrow (c_{MI} : 2^{\mathcal{X}} \to \mathcal{Y})$ $\mathcal{H}_I \leftrightarrow \mathcal{H}_{MI}$ $k_I \leftrightarrow k_{MI}$



Graph Kernels [colt'03,kdd'04]

idea: define kernel as some function of the **common subgraphs** (cf strings, trees, ...)

computing this graph kernel is NP-hard; even for paths, cycles, or connected subgraphs (with subgraph-isomorphism as the embedding)

computing **complete graph kernels** is at least as hard as deciding graph isomorphism

for efficient computation consider

- **particular graphs-classes**
- **different embedding operators, e.g., homomorphism**

Kernels for Basic Terms [icml'02, mlj'04]

principles
represent individuals by typed terms
represent sets, multisets, etc. naturally

logic for learning [Lloyd, 2003]
higher-order logic with polymorphic types,
the terms of the logic are the terms of the
typed λ-calculus,

basic terms
are essentially unambiguous closed terms
of the logic,basic abstractions are used to
represent sets, multisets, etc

valid! **efficient!**

Kernels for Basic Terms [icml'02, mlj'04]

```
type Spectrum     = Frequency -> Multiplicity
type Frequency    = Real with modifier Gaussian 0.6
data Multiplicity = s | d | t | q | 0 default 0
```

principles
represent individuals by typed terms
represent sets, multisets, etc. naturally

logic for learning [Lloyd, 2003]
higher-order logic with polymorphic types, the terms of the logic are the terms of the typed λ-calculus,

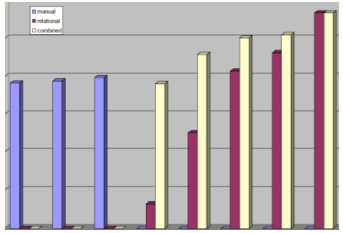
basic terms
are essentially unambiguous closed terms of the logic,basic abstractions are used to represent sets, multisets, etc

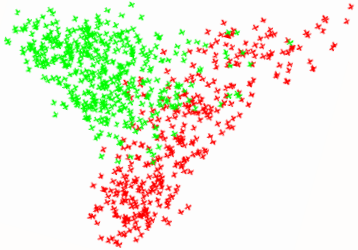
valid! efficient!

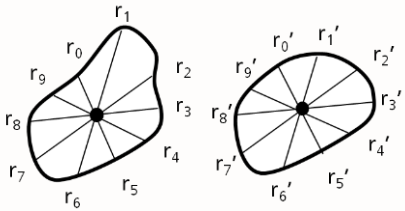
effective?

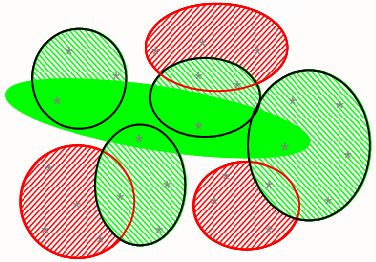
task
given the ^{13}C NMR spectrum of Diterpenes, predict the skeleton class it belongs to

data
1503 spectra classified into 23 skeleton types







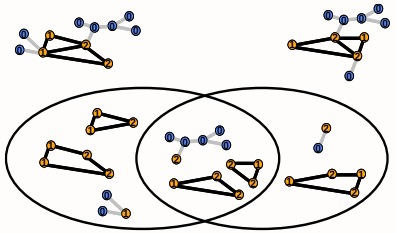


$(c_I : \mathcal{X} \to \mathcal{Y}) \leftrightarrow (c_{MI} : 2^{\mathcal{X}} \to \mathcal{Y})$ $\mathcal{H}_I \leftrightarrow \mathcal{H}_{MI}$ $k_I \leftrightarrow k_{MI}$

Graph Kernels [colt'03,kdd'04]

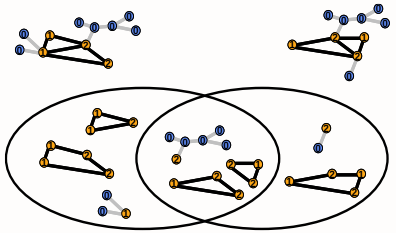
common approach:
define kernel as some function of the
common subgraphs (cf strings, trees, ...)

computing this graph kernel is NP-hard;
even for paths, cycles, or connected subgraphs
(with subgraph-isomorphism as the embedding)



Graph Kernels [colt'03,kdd'04]

common approach:
define kernel as some function of the
common subgraphs (cf strings, trees, ...)



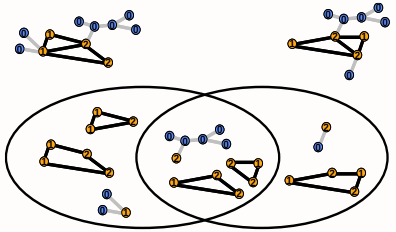
computing this graph kernel is NP-hard;
even for paths, cycles, or connected subgraphs
(with subgraph-isomorphism as the embedding)

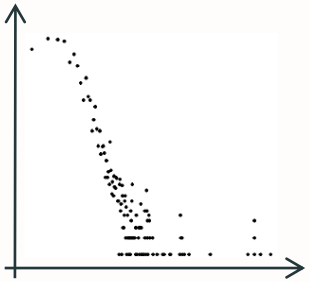
computing **complete graph kernels**
is at least as hard as deciding graph isomorphism

$$k(G, \cdot) = k(G', \cdot) \Leftrightarrow G \simeq G'$$

Graph Kernels [colt'03,kdd'04]

common approach:
define kernel as some function of the
common subgraphs (cf strings, trees, ...)

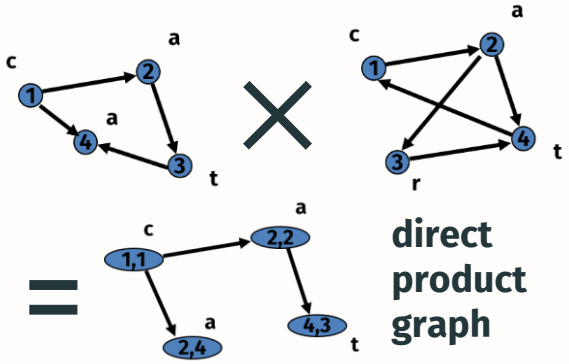




computing this graph kernel is NP-hard;
even for paths, cycles, or connected subgraphs
(with subgraph-isomorphism as the embedding)

computing **complete graph kernels**
is at least as hard as deciding graph isomorphism

$$k(G,\cdot) = k(G',\cdot) \Leftrightarrow G \simeq G'$$



direct product graph

for efficient computation consider
- **particular graphs-classes**
- **different embedding operators, e.g., homomorphism**

$$k_\times(G_1,G_2) = \mathbf{1}^\top \left[\lim_{n\to\infty} \sum_{i=0}^{n} \eta_i E_\times^i \right] \mathbf{1}$$

Structured Prediction [mlj'09, uai'09]

common approach:
assume structures in the data are "***better***"
than structures not in the data

to distinguish between good and bad structures,
we obtain an **optimisation problem** with
exponentially many constraints

it can be solved efficiently under the
assumption that there is an efficient algorithm for
finding a "**better**" structure

Structured Prediction [mlj'09, uai'09]

common approach:
assume structures in the data are "***better***"
than structures not in the data

to distinguish between good and bad structures,
we obtain an **optimisation problem** with
exponentially many constraints

it can be solved efficiently under the
assumption that there is an efficient algorithm for
finding a "**better**" structure

**However, we have shown that this assumption
does not hold for many problems involving graphs**

Structured Prediction [mlj'09, uai'09]

common approach:
assume structures in the data are "***better***" than structures not in the data

to distinguish between good and bad structures, we obtain an **optimisation problem** with exponentially many constraints

it can be solved efficiently under the **assumption** that there is an efficient algorithm for finding a "**better**" structure

However, we have shown that this assumption does not hold for many problems involving graphs

alternative 1:

replace the exponential number of constraints by an (exponential) sum with terms like

$$\sum_{x \in \mathcal{X}} h(x, y) \quad \text{or} \quad \sum_{x \in \mathcal{X}} h^2(x, y)$$

which can "often" be **computed efficiently**

alternative 2:

a fixed-parameter tractable approximation scheme for the **partition function**

$$\sum_{x \in \mathcal{X}} exp(\phi(x, y), h)$$

parameter of the runtime is $\|h\|$, i.e., the "simplicity" of the hypothesis

Time and Sample Complexity

given a fixed confidence target and constant precision parameter

$$\delta = \frac{1}{2\mathbf{r}}; \quad \Delta = (\mathbf{r}\delta)^{2^h} = 2^{-2^h} \qquad m(\epsilon, \delta) = \alpha_\epsilon + \beta_\epsilon \log_2 \frac{1}{\delta} \qquad t(m) \in O\left(m^k\right)$$

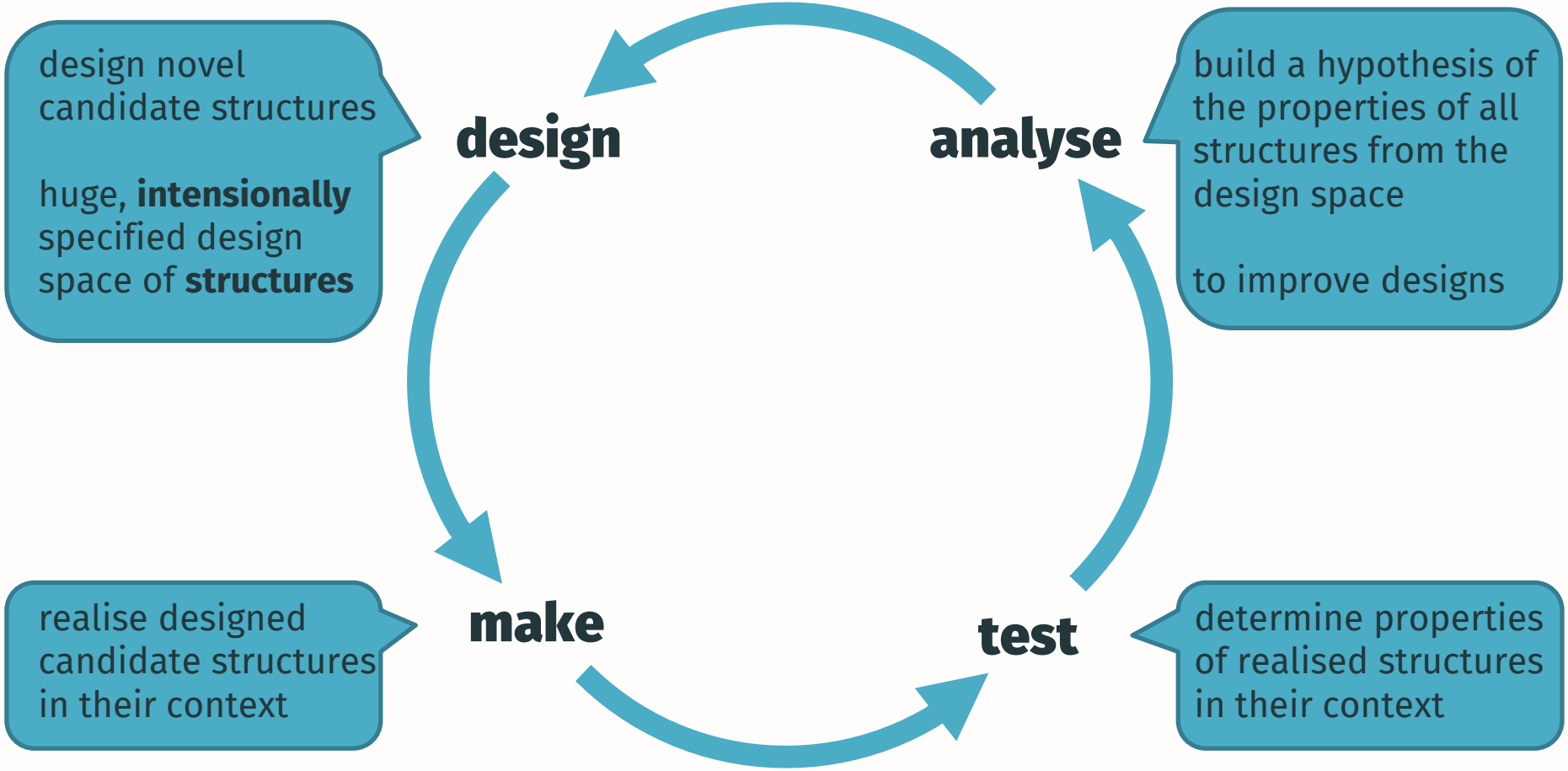
single centralised machine

$$M = \alpha_\epsilon + \beta_\epsilon 2^h; \quad T \in O\left(2^{hk}\right)$$

parallel Radon machine

$$m = \alpha_\epsilon + \beta_\epsilon \log_2(2\mathbf{r}); \quad t \in O\left(\log_2^k(2\mathbf{r}) + h\mathbf{r}^3\right)$$

Cyclic Discovery Processes



appendix

Consistency of Radon Points

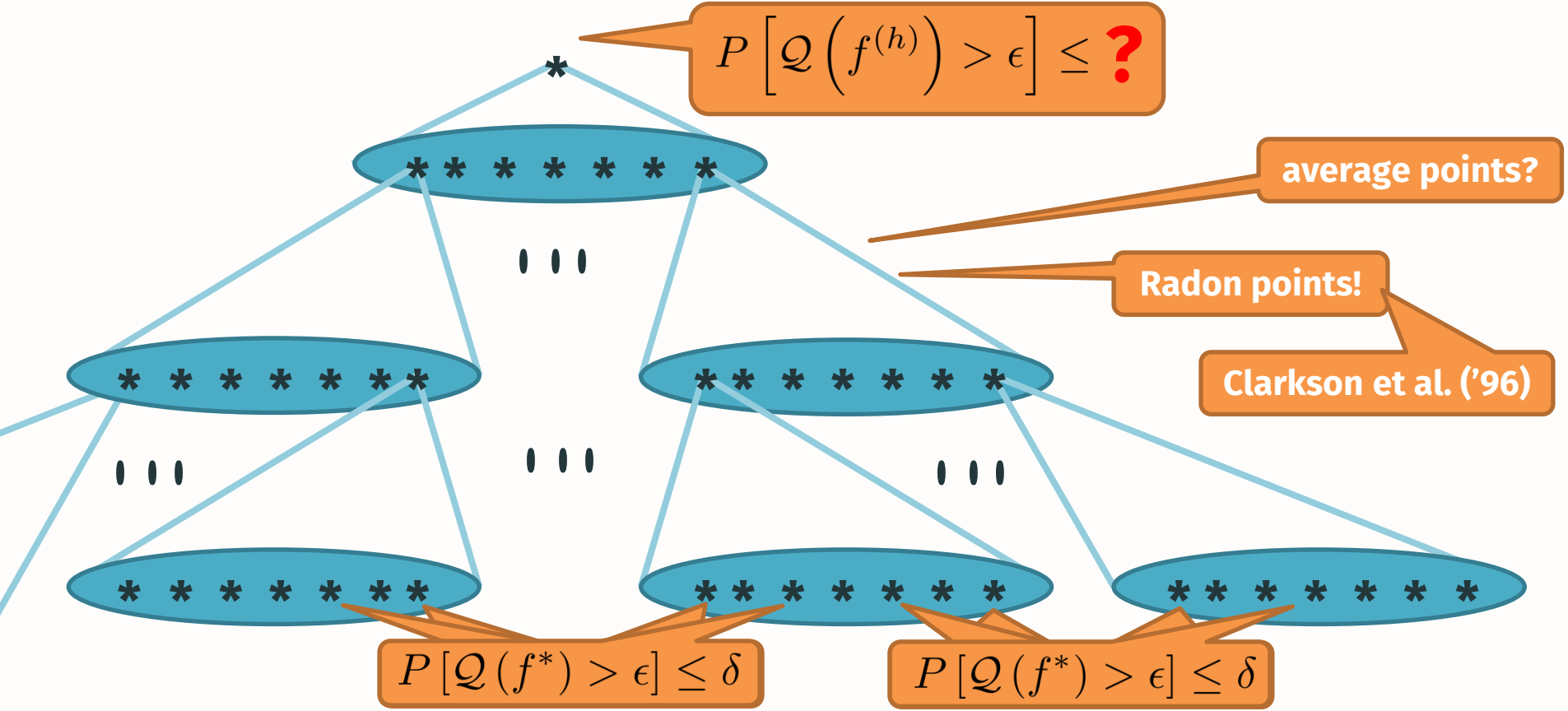
$$P\left[\ \mathcal{Q}\left(f^{(i+1)}\right) > \epsilon\ \right] \leq \left[\ \mathbf{r}\ P\left[\ \mathcal{Q}\left(f^{(i)}\right) > \epsilon\ \right]\ \right]^2$$

Radon number

could we achieve this on a single machine?

$$P\left[\ \mathcal{Q}\left(f^{(h)}\right) > \epsilon\ \right] \leq \left[\ \mathbf{r}\ P\left[\ \mathcal{Q}\left(f^{*}\right) > \epsilon\ \right]\ \right]^{2^h} \leq (\mathbf{r}\delta)^{2^h} = \Delta$$

Parallelisation by Aggregation



Consistency of Radon Points

$$P\left[q\left(f^{(i+1)}\right) < \epsilon\right] \leq \left(P\left[q\left(f^{(i)}\right) < \epsilon\right]\right)^2$$

$$P\left[\mathcal{Q}\left(f^{(i+1)}\right) > \epsilon\right] \leq \left[\mathbf{r}\, P\left[\mathcal{Q}\left(f^{(i)}\right) > \epsilon\right]\right]^2$$

Radon number

could we achieve this on a single machine?

$$P\left[\mathcal{Q}\left(f^{(h)}\right) > \epsilon\right] \leq \left[\mathbf{r}\, P\left[\mathcal{Q}(f^*) > \epsilon\right]\right]^{2^h} \leq (\mathbf{r}\delta)^{2^h} = \Delta$$

$$P\left[q\left(f^{(h)}\right) < \epsilon\right] \leq \left(P\left[q\left(f^{(i)}\right) < \epsilon\right]\right)^{2^h}$$

Consistency of Radon Points

$$P\left[q\left(\quad X \quad\right) > \epsilon\right] \quad \leq \delta$$

Consistency of Radon Points

$$P[q(R) > \epsilon] \leq (r\, P[q(X) > \epsilon])^2$$

Consistency of Radon Points

$$P\left[q\left(f^{(0)}\right) > \epsilon\right] \leq \delta$$

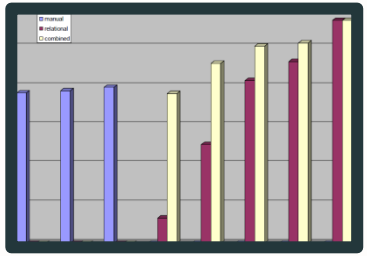
$$P\left[q\left(f^{(h)}\right) > \epsilon\right] \leq \left(r\, P\left[q\left(f^{(h-1)}\right) > \epsilon\right]\right)^{2}$$

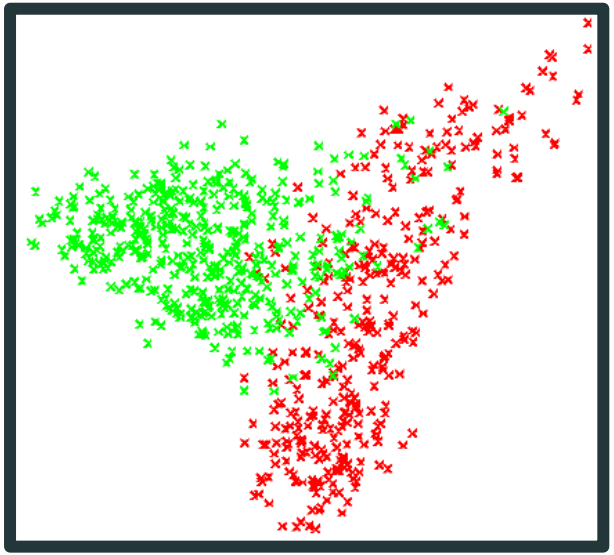
$$P\left[q\left(f^{(h)}\right) > \epsilon\right] \leq \left(r\, P\left[q\left(f^{(0)}\right) > \epsilon\right]\right)^{2^h} \leq (r\delta)^{2^h} = \Delta$$

what is the sequential runtime complexity of achieving the same bound Δ with the original algorithm?

Structure Elucidation [mlj'04]

```
type Spectrum     = Frequency -> Multiplicity
type Frequency    = Real with modifier Gaussian 0.6
data Multiplicity = s | d | t | q | 0 default 0
```

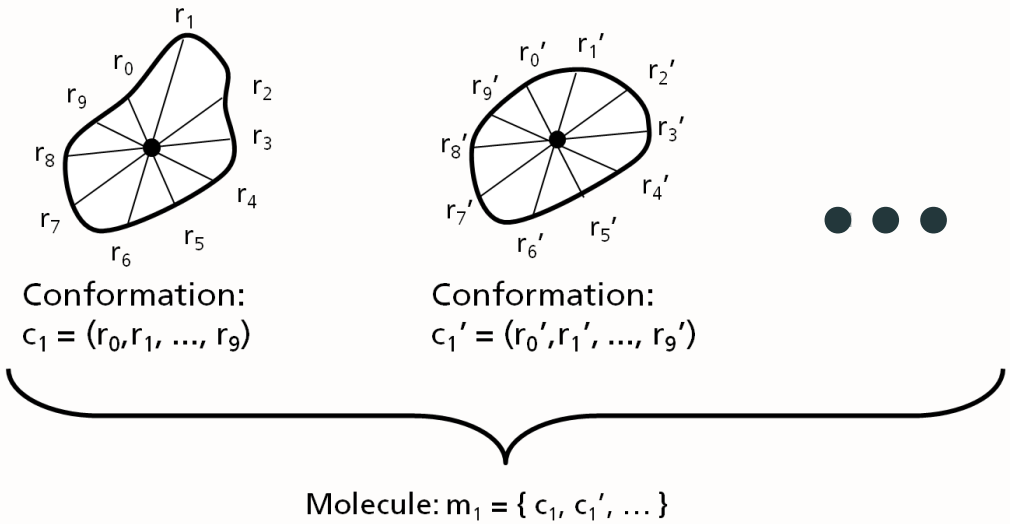


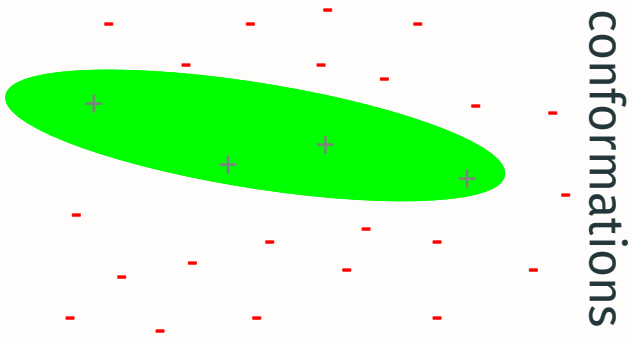


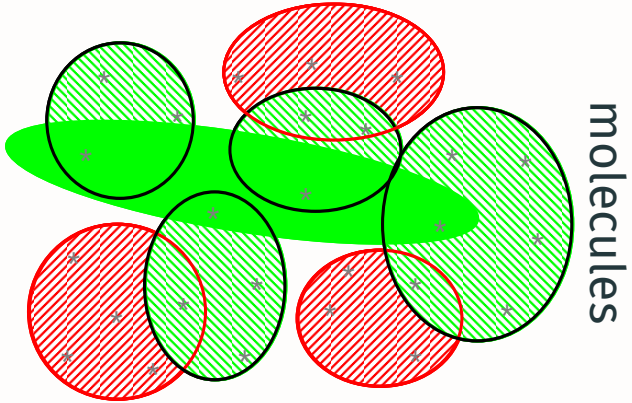
task
given the ^{13}C NMR spectrum of Diterpenes,
predict the skeleton class it belongs to

data
1503 spectra classified into 23 skeleton types

Multi-Instance Kernels [icml'02]







$$f^*_{MI}(m) = +1 \Leftrightarrow \exists c \in m : f^*_I(c) = +1$$

multi-instance concepts can be separated with kernels
for basic terms & the number of mistakes can be bounded.

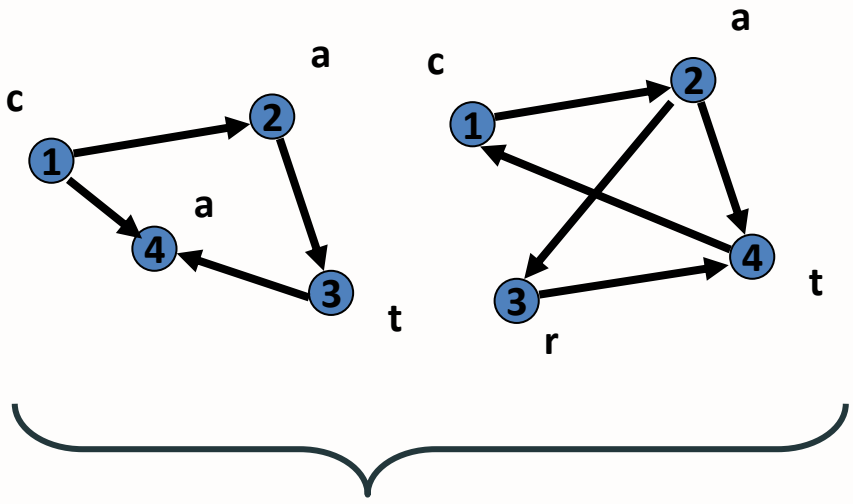
All-Walks Kernels [colt'03]

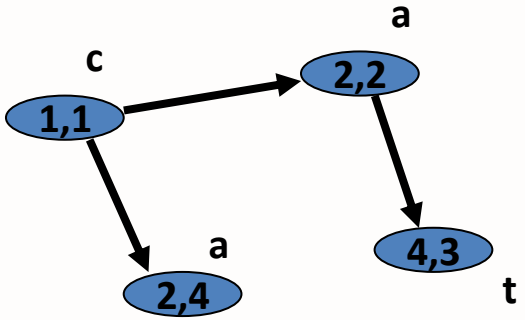
resort to non-complete graph kernels

consider as patterns the (multi-)set of all labelled walks contained in a graph

while there can be infinitely many walks we can measure the common ones efficiently

$$k_\times(G_1, G_2) = \mathbf{1}^\top \left[\lim_{n\to\infty} \sum_{i=0}^{n} \eta_i E_\times^i \right] \mathbf{1}$$

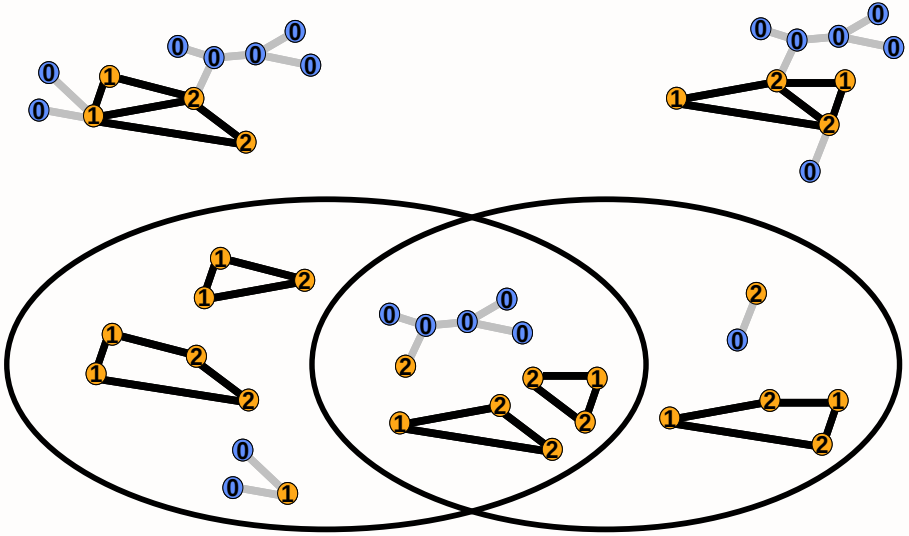


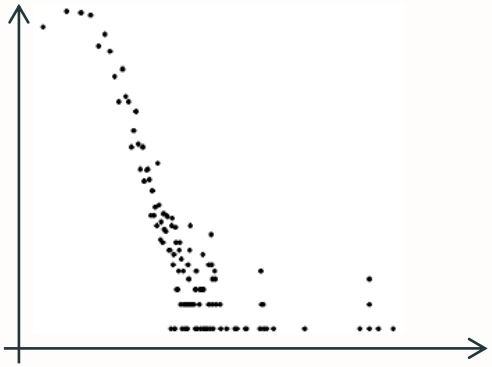


Cyclic Pattern Kernels [kdd'04]

consider as patterns the set of all labelled cycles contained in a graph as well as all trees in the forrest of bridges

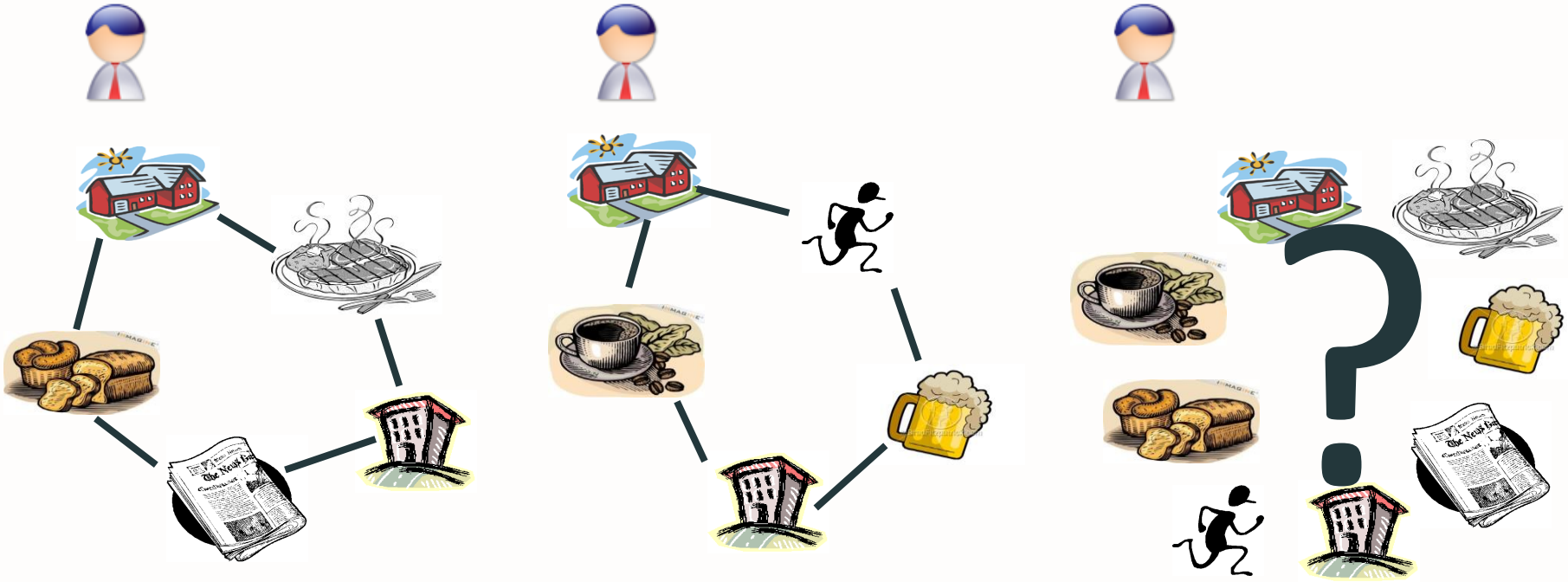
computing CPK is NP-hard but fixed-parameter tractable





log/log plot for HIV dataset
#cycles over #molecules

Travelling John Q Public Problem



SVM-like Structured Prediction

exponentially many constraints

$$\hat{f} = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \quad \nu \|f\|^2 + \|\xi_i\|_1$$
$$\text{subject to} \quad f(x_i, y_i) \;>\; \Delta(y_i, z) + f(x_i, z) - \xi_i \quad (\forall i, \forall z \in \mathcal{Y} \setminus \{y_i\})$$

can be solved efficiently by column-generation/cutting-plane techniques

Finding Violated Constraints

usually used in algorithms

decoding

given *f*, *x* find $z \in \operatorname*{argmax}_{y \in \mathcal{Y}} f(x, y)$

usually sufficient

separation

given *f*, *x*, *y* find any $z : f(x, y) < f(x, z)$ (or prove none exists)

optimality

given *f*, *x*, *y* decide whether $\nexists z \in \mathcal{Y} : f(x, y) < f(x, z)$

"decoding is in P" is the strongest assumption
while **"optimality is in NP"** is the weakest

Optimality vs Non-Optimality

optimality: is there **no** longer cycle?

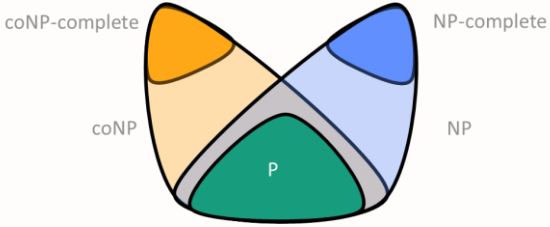
given f, x, y decide $\nexists z \in \mathcal{Y} : f(x,y) < f(x,z)$

what is a short certificate of optimality?

sub-optimality: is there **any** longer cycle?

given f, x, y decide $\exists z \in \mathcal{Y} : f(x,y) < f(x,z)$

certificate of non-optimality is short



sub-optimality is NP-complete

hence **optimality** is coNP-complete and not in NP

Structured Output Ranking [mlj'09, uai'09]

for least squares, the following terms occur in the objective

$$\sum_{y \in \mathcal{Y}} f(x_i, y) = {f_\alpha^i}^\top \sum_{z \in \mathcal{Y}} \phi(z)$$

$$\sum_{y \in \mathcal{Y}} f^2(x_i, y) = {f_\alpha^i}^\top \sum_{z \in \mathcal{Y}} \phi(z)\phi^\top(z) f_\alpha^i$$

often these **exponential sums** can be computed in **polynomial time**

for conditional exponential families, the partition function is

$$\sum_{y \in \mathcal{Y}} \exp(\langle \phi(x, y); \theta \rangle)$$

often this **exponential sum** has a *fixed-parameter tractable* **randomised approximation scheme**

Traditional Difficulty Adjustment

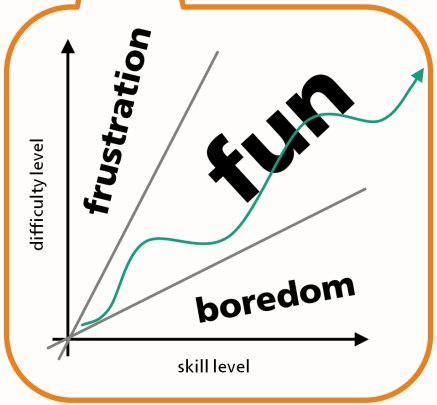
Online Learning in Set Systems

given a **universe** and a **family of subsets** of the universe:
1. **environment** picks a `feasible' set from the family
2. **learner** guesses a point
3. **environment** reveals a `separating' set

make as few as possible infeasible queries (mistakes)
do not to rely on Euclidean geometry (convexity space)
do not make assumptions about the environment (adversary)

conceptually:
maintain belief over universe
multiplicatively update upon mistake
guess points that **enforce large updates**

algorithmically:
never compute belief explicitly / **approximate** guesses



Dynamic Difficulty Adjustment [nips'11]

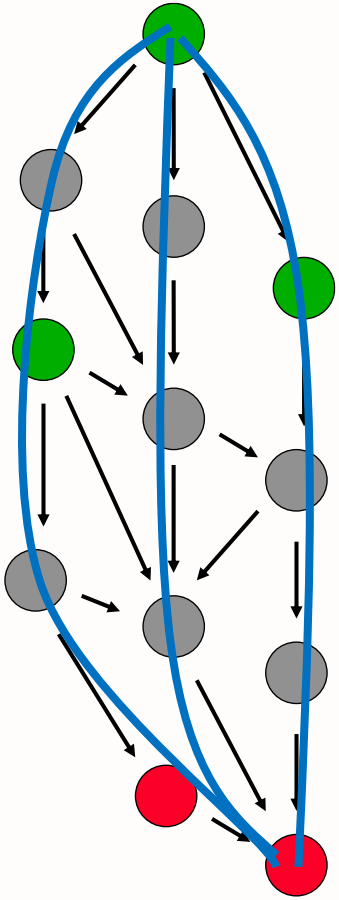
1) **game master predicts** a difficulty setting

2) **player plays** one round of the game in this setting

3) **game master observes** whether the setting was too easy, just right, or too difficult

this game is played on a partially ordered set reflecting the **`is more difficult'-relation**
the goal of the game master is to predict `just right'-settings as often as possible

we can bound the number of mistakes in this setting



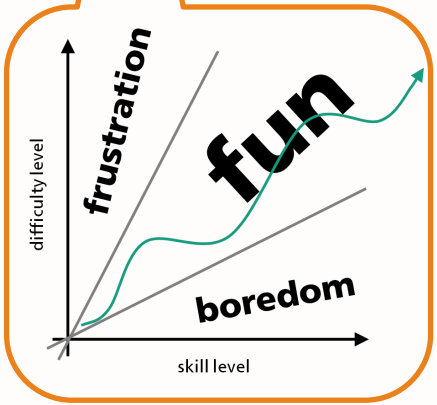
Online Learning in Set Systems

given a **universe** and a **family of subsets** of the universe:
1. **environment** picks a `feasible' set from the family
2. **learner** guesses a point
3. **environment** reveals a `separating' set

make as few as possible infeasible queries (mistakes)
do not to rely on Euclidean geometry (convexity space)
do not make assumptions about the environment (adversary)

conceptually:
maintain belief over universe
multiplicatively update upon mistake
guess points that **enforce large updates**

algorithmically:
never compute belief explicitly / **approximate** guesses



Dynamic Difficulty Adjustment [nips'11, cig'11]

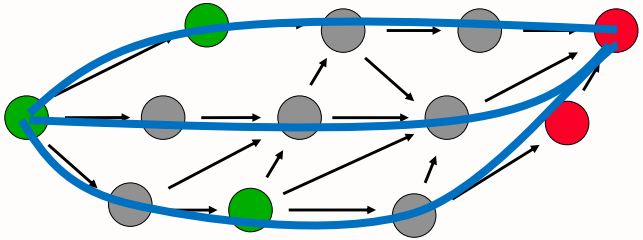
1) **game master predicts** a difficulty setting

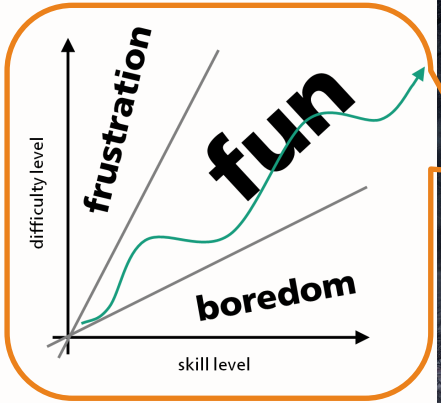
2) **player plays** one round of the game in this setting

3) **game master observes** whether the setting was too easy, just right, or too difficult

this game is played on a partially ordered set reflecting the **`is more difficult'-relation** the goal of the game master is to predict `just right'-settings as often as possible

we can bound the number of mistakes in this setting





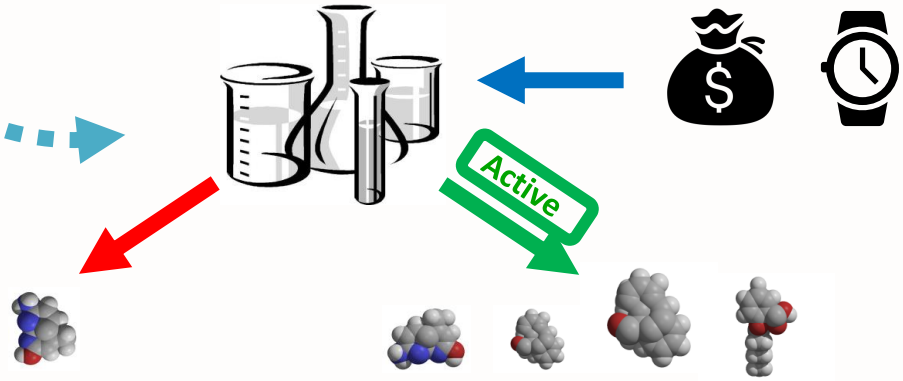
de novo Drug Design [aaai'17]

weighted maximum a posteriori

$$\theta_t = \operatorname*{argmax}_{\theta \in \mathcal{H}} p(Y \mid X, \theta) p(\theta)$$

sample

$$x_t \sim p(x \mid y^*, \theta_t)$$



Moore's Law

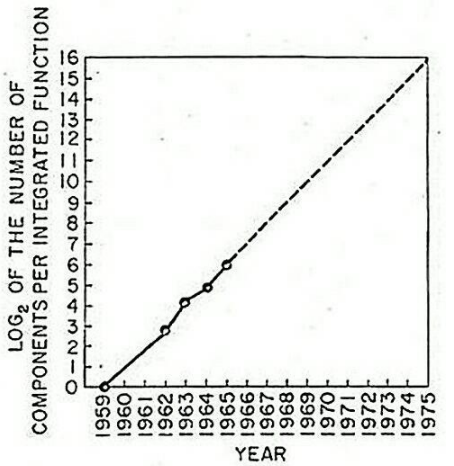
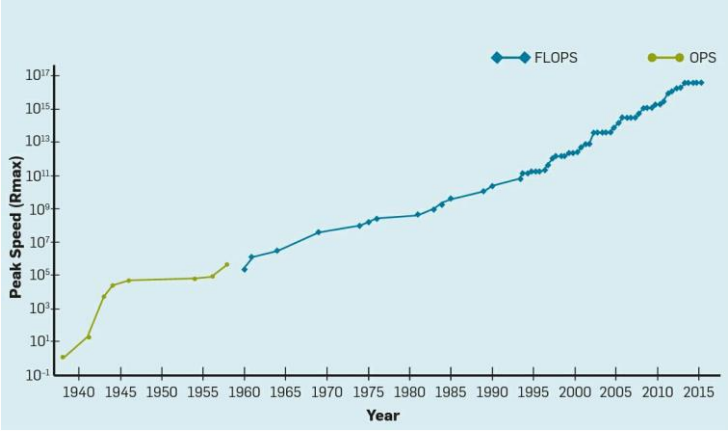
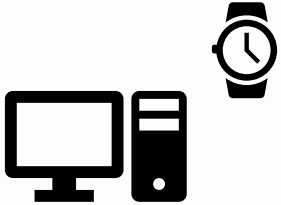
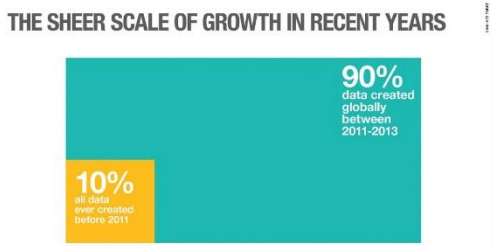
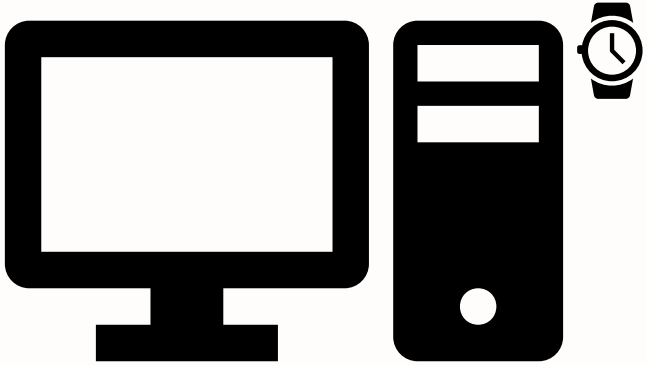


Fig. 2. Number of components per integrated function for minimum cost per component extrapolated vs time.









Learning as Randomised Optimisation

$$D = \{(x_i, y_i)\}_{i=1}^{m} \sim P^m$$

$$\mathcal{L} : D \mapsto f^*$$

convex

$$f^* = \operatorname*{argmin}_{w \in \mathbb{R}^d} \sum_{i=1}^{m} \ell\left(F_w(x_i), y_i\right) + \Omega\left[w\right]$$

quality, convex

such that $P\left[\mathcal{Q}\left(f^*\right) > \epsilon\right] \leq \delta$
whenever $|D| = m > poly\ (1/\varepsilon, \ln 1/\delta)$

for "moderate" m, often interior point methods or special-purpose algorithms are used

how can we scale up?
(stochastic) gradient descent can easily be parallelised

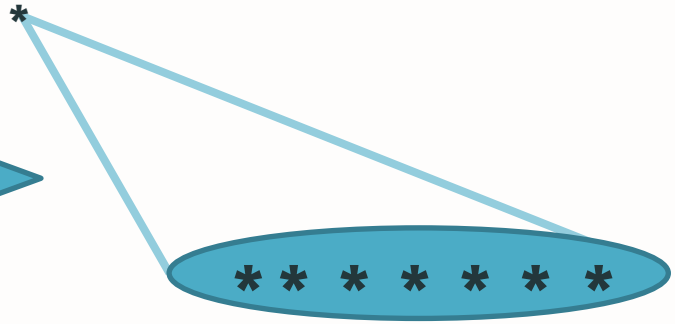
can we scale up independent of the underlying algorithm?

Parallelisation by Aggregation-at-the-end

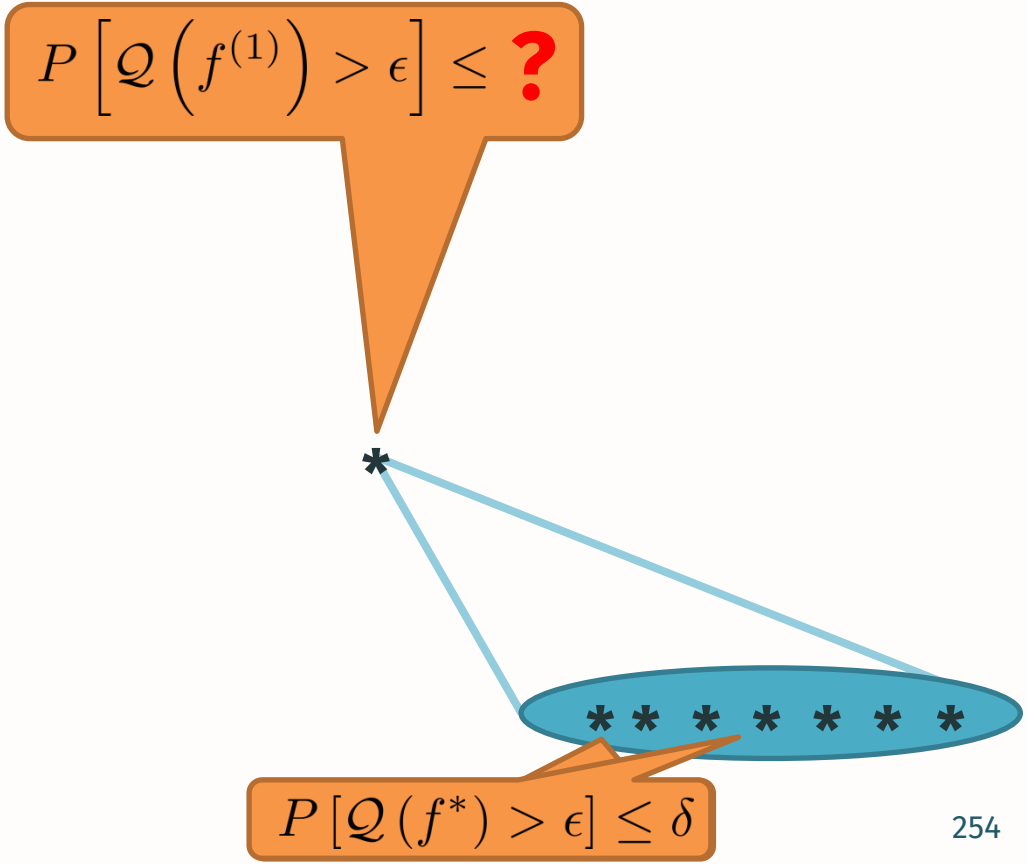
why?
learning algorithms are super-linear
can process chops in parallel
general (“any” base-learner)!

how do we know it’s effective?

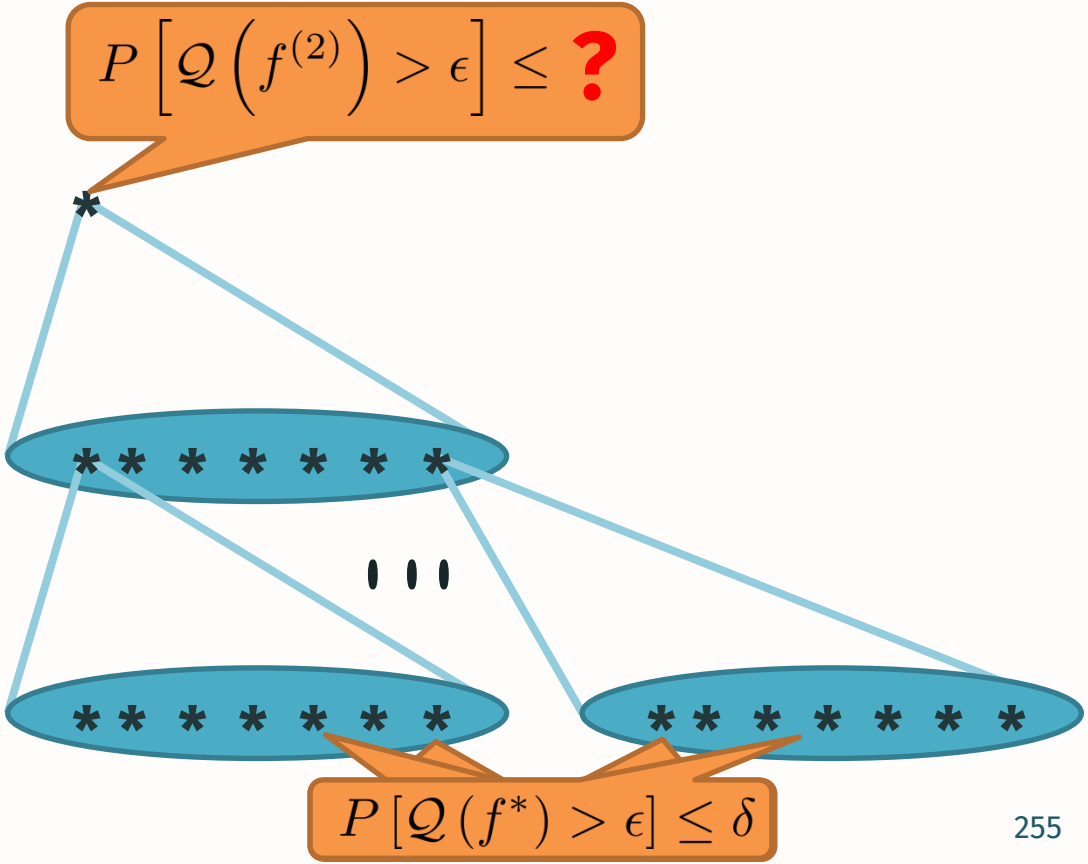
idea:
1. chop up data into small pieces
2. apply a “base” learning algorithm to those
3. aggregate weak hypotheses to stronger ones



Parallelisation by Aggregation



Parallelisation by Aggregation



Radon Points

in two examples

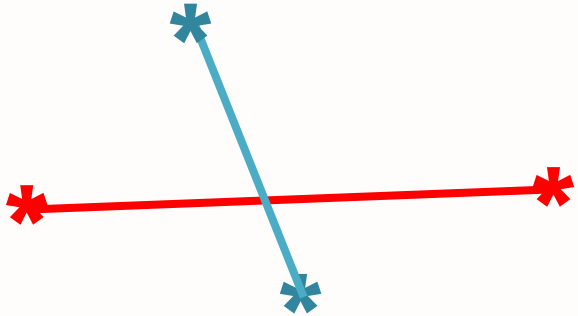
Radon Points

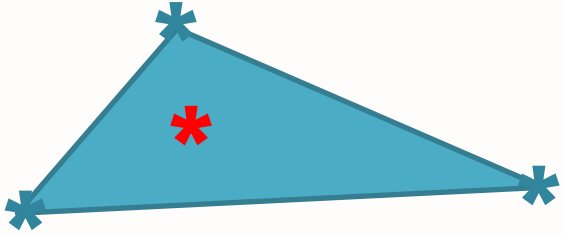
a **bipartition** of a set is a pair of disjoint subsets

Radon Points

a **bipartition** of a set is a pair of disjoint subsets

a **Radon partition** of a set is a bipartition whose convex hulls overlap



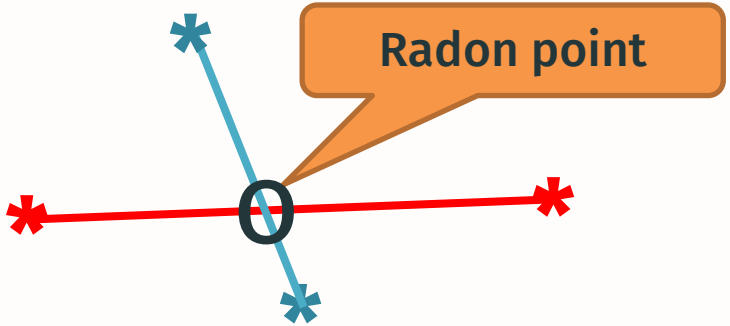


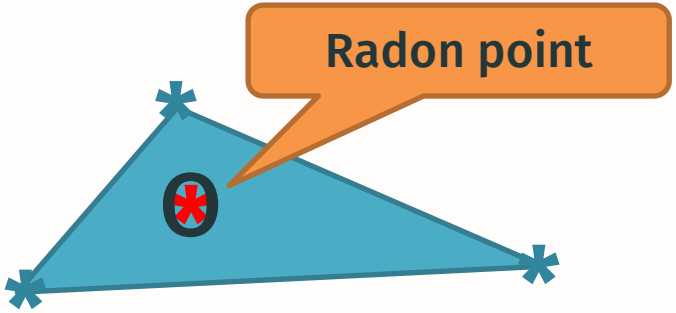
Radon Points

a **bipartition** of a set is a pair of disjoint subsets

a **Radon partition** of a set is a bipartition whose convex hulls overlap

a **Radon point** of a set is a point in this overlap





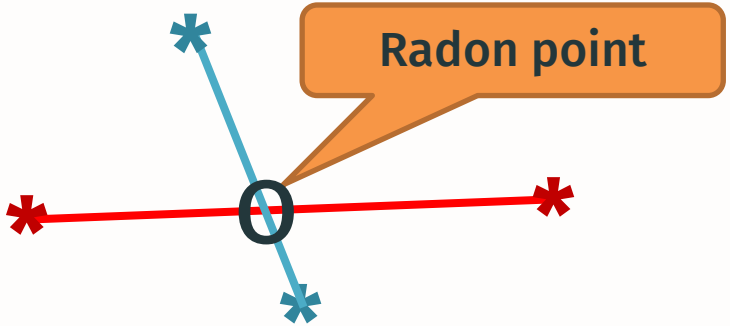
Radon Points

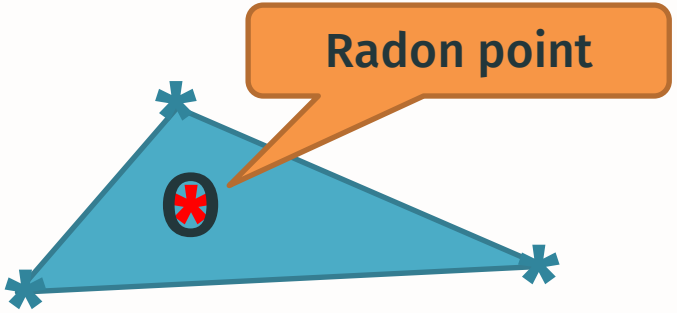
a **bipartition** of a set is a pair of disjoint subsets

a **Radon partition** of a set is a bipartition whose convex hulls overlap

a **Radon point** of a set is a point in this overlap

the **Radon number** r is the smallest size such that any set of that size has a Radon partition (for Euclidean spaces $\mathbb{R}^d$ Radon's Theorem shows that $r = d + 2$)





Radon Points

a **bipartition** of a set is a pair of disjoint subsets

a **Radon partition** of a set is a bipartition whose convex hulls overlap

a **Radon point** of a set is a point in this overlap

the **Radon number** r is the smallest size such that any set of that size has a Radon partition (for Euclidean spaces $\mathbb{R}^d$ Radon's Theorem shows that $r = d + 2$)

Radon Points

a **bipartition** of a set is a pair of disjoint subsets

a **Radon partition** of a set is a bipartition whose convex hulls overlap

a **Radon point** of a set is a point in this overlap

the **Radon number** r is the smallest size such that any set of that size has a Radon partition (for Euclidean spaces $\mathbb{R}^d$ Radon's Theorem shows that $r = d + 2$)

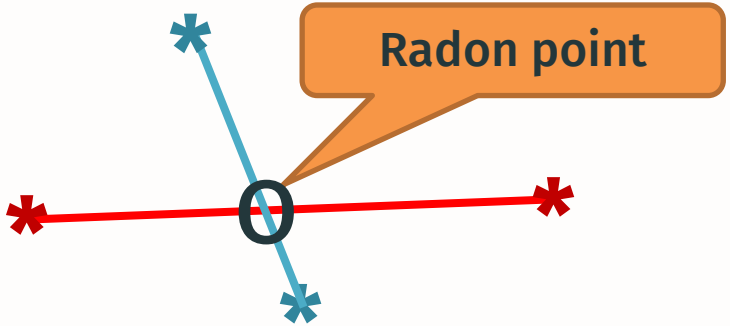
Radon Points

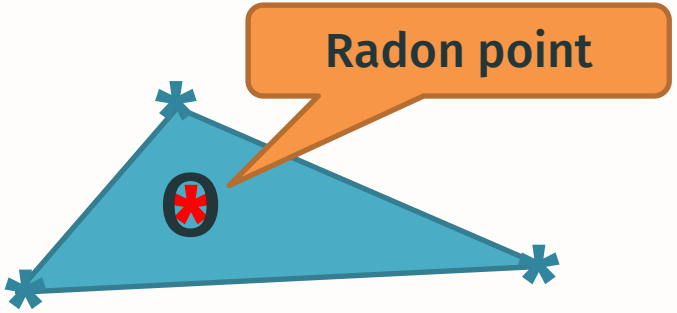
a **bipartition** of a set is a pair of disjoint subsets

a **Radon partition** of a set is a bipartition whose convex hulls overlap

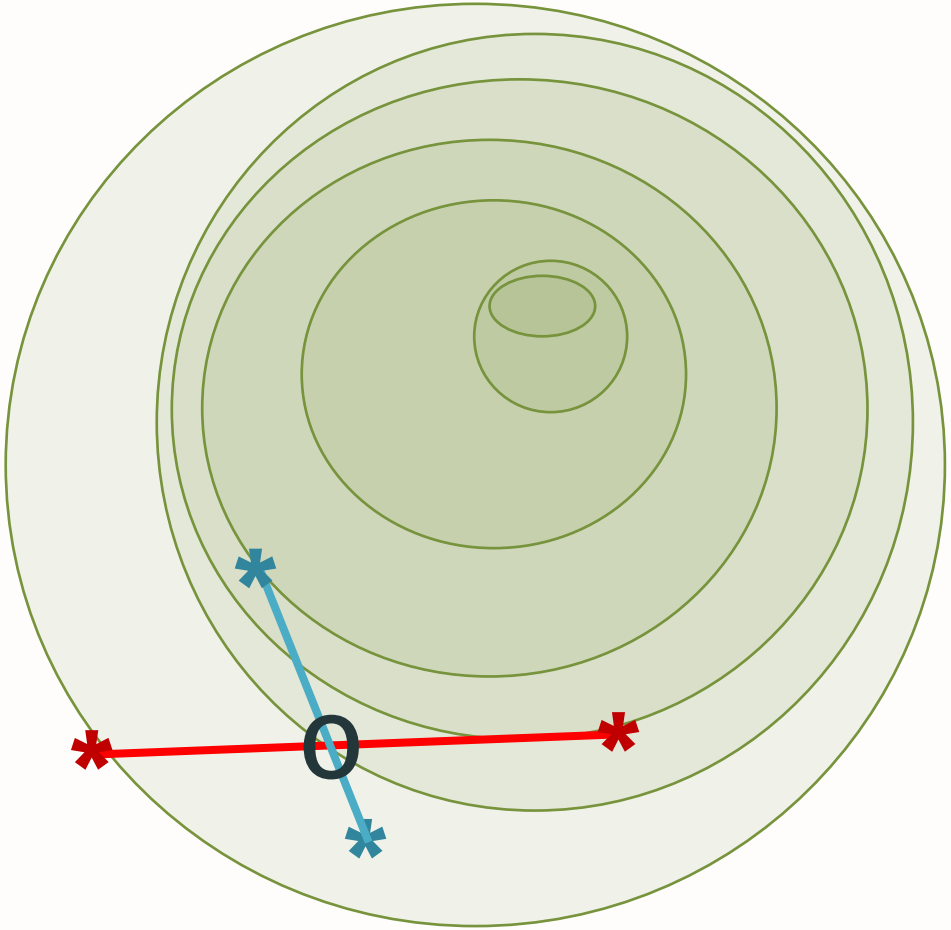
a **Radon point** of a set is a point in this overlap

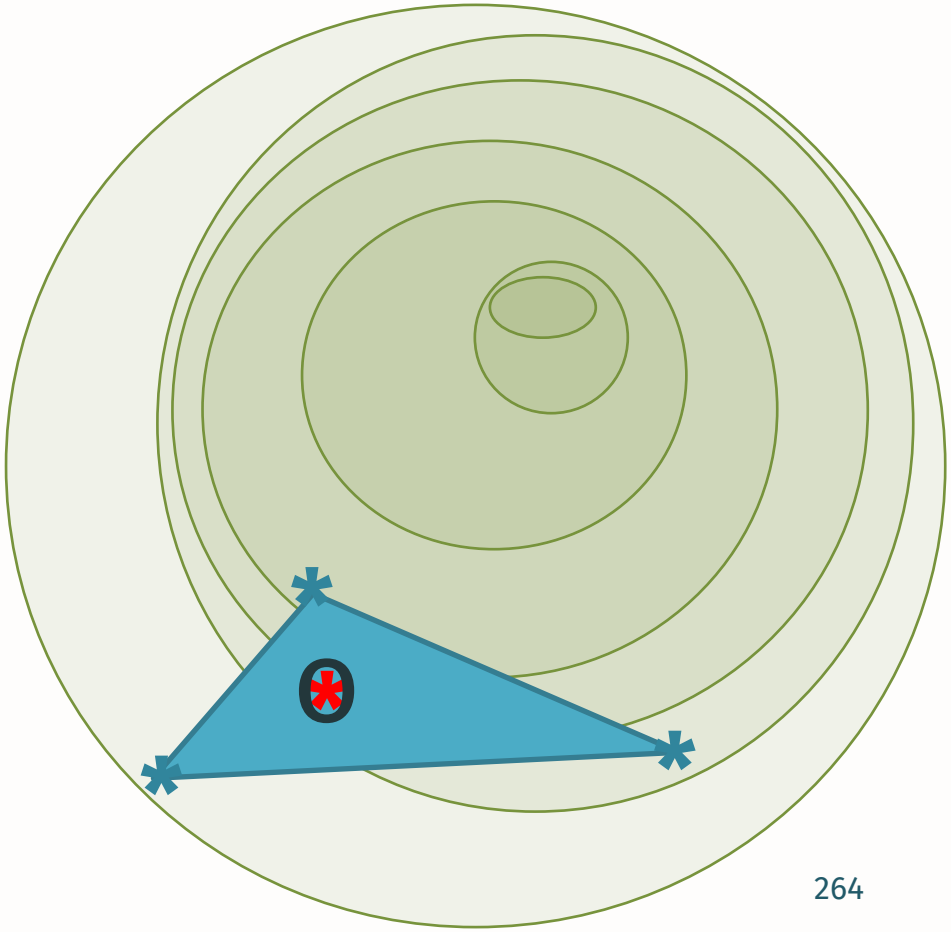
the **Radon number** r is the smallest size such that any set of that size has a Radon partition (for Euclidean spaces $\mathbb{R}^d$ Radon's Theorem shows that $r = d + 2$)





Radon Points and Quasiconvexity

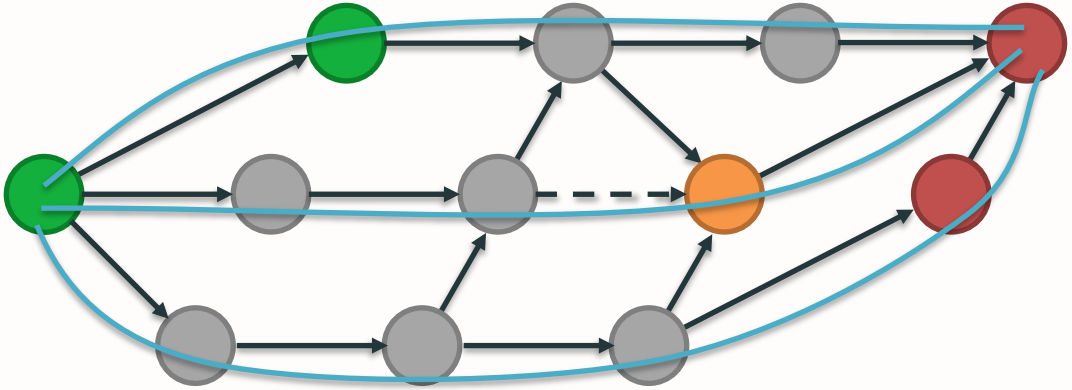


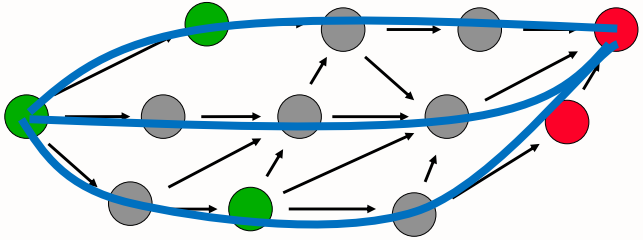


Consistency of Radon Points

$$P\left[\ \mathcal{Q}\left(f^{(i+1)}\right) > \epsilon\ \right] \leq \left[\ \mathbf{r}\ P\left[\ \mathcal{Q}\left(f^{(i)}\right) > \epsilon\ \right]\ \right]^2$$

Radon number





cope with the candidate space for analysis

cope with the candidate space for design

cope with the cyclic nature

interact with domain experts

design novel structures

dealing with too much data

Ta

University of Nottingham
Rights Lab

Almost 46 million people are enslaved around the world today

Ending slavery by 2030 is part of the UN Sustainable Development Goals

Can we detect locations of forced labour from geospatial images?

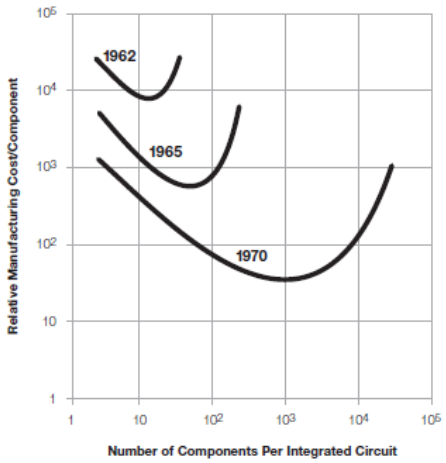
- active learning to reduce the time humans that is needed

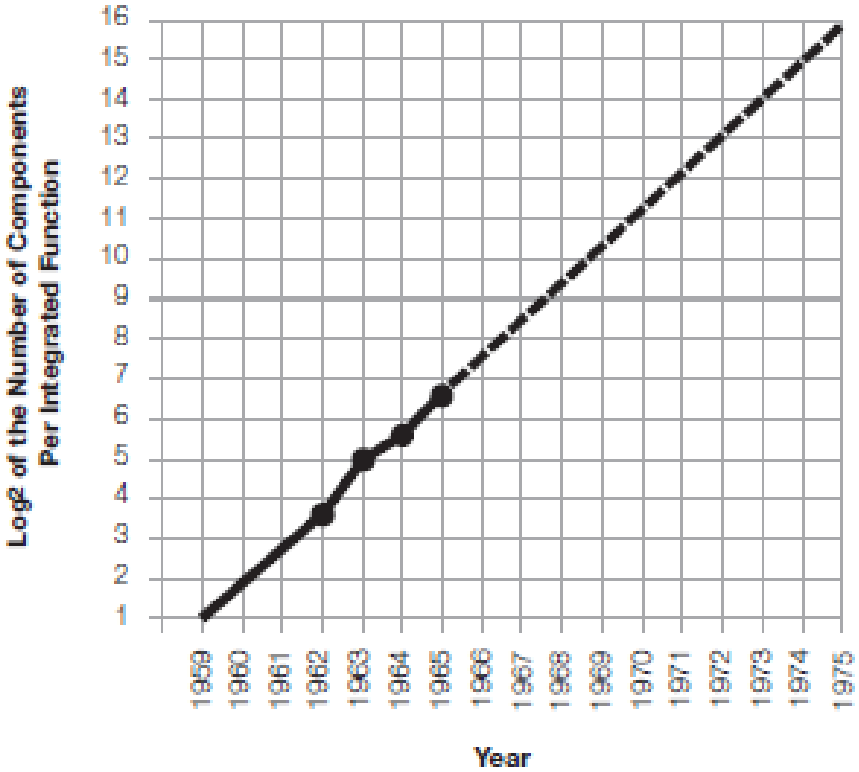
Cramming more components onto integrated circuits

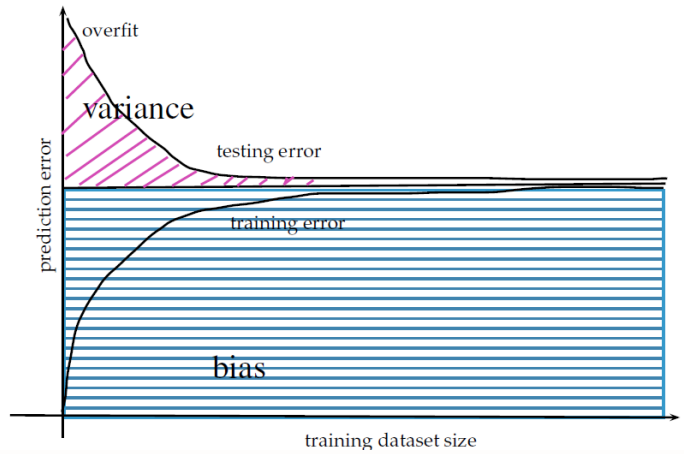
With unit cost falling as the number of components per circuit rises, by 1975 economics may dictate squeezing as many as 65,000 components on a single silicon chip

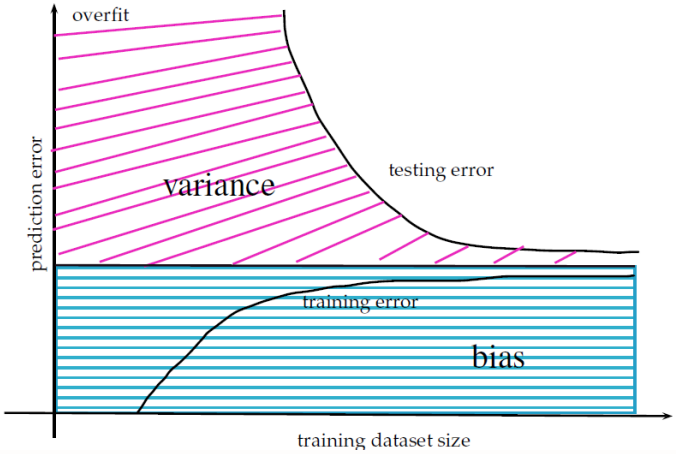
By Gordon E. Moore

Director, Research and Development Laboratories, Fairchild Semiconductor division of Fairchild Camera and Instrument Corp.

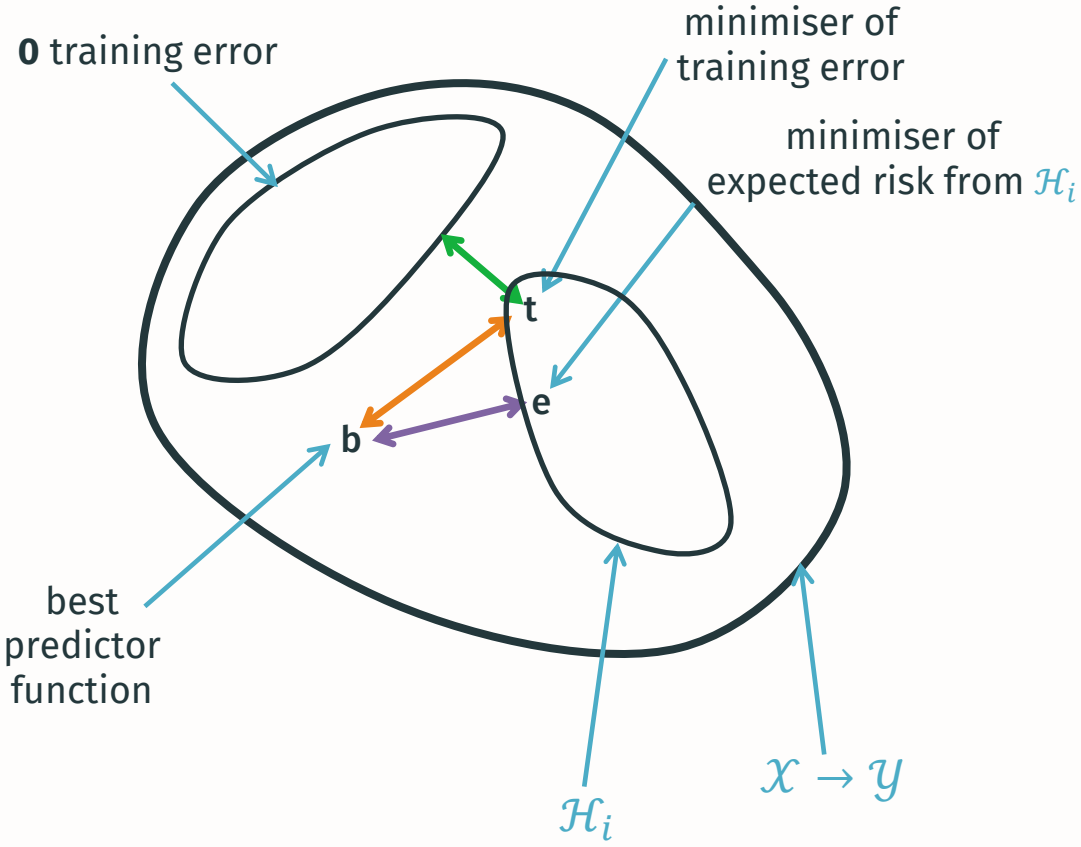




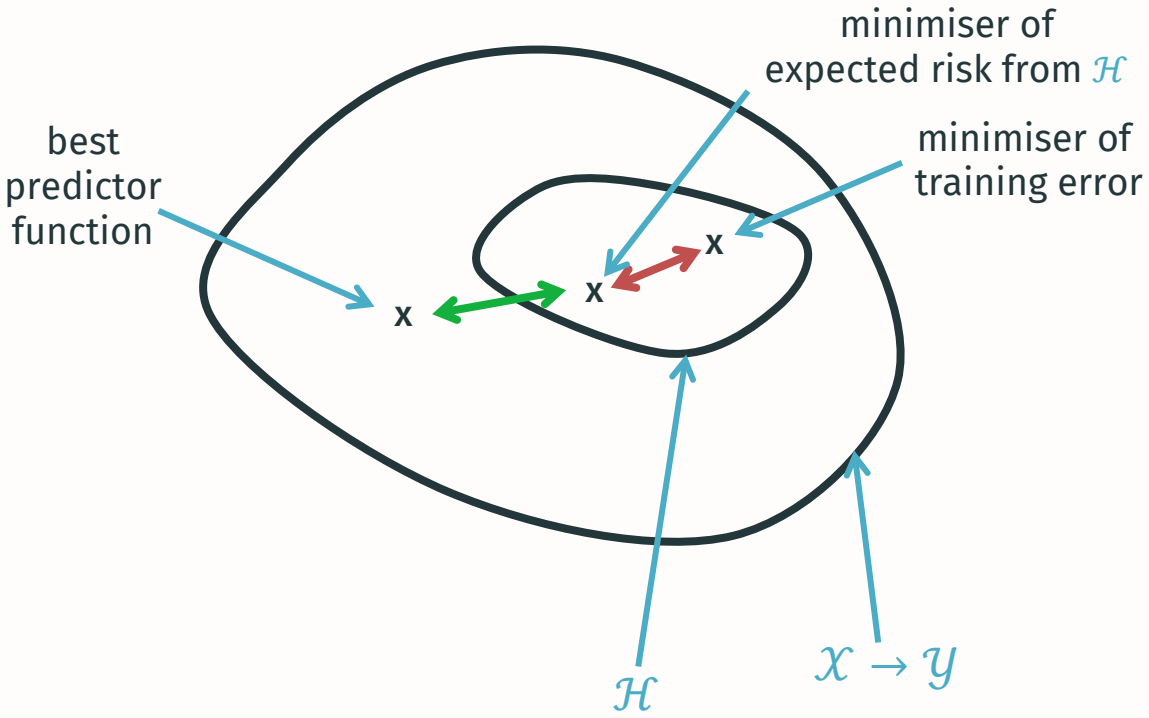

overfit
variance
testing error
training error
bias
prediction error
training dataset size


overfit
variance
testing error
training error
bias
prediction error
training dataset size

Risk Structure



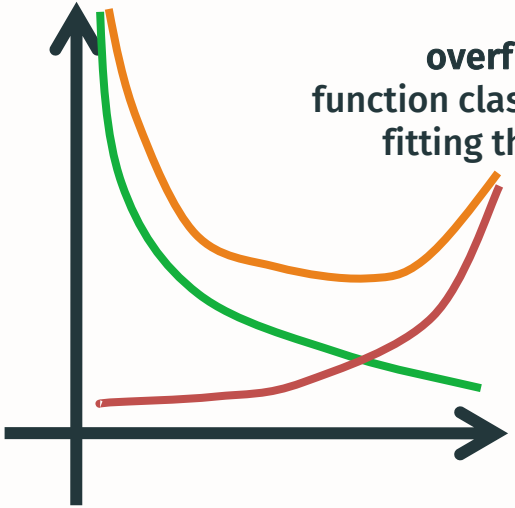
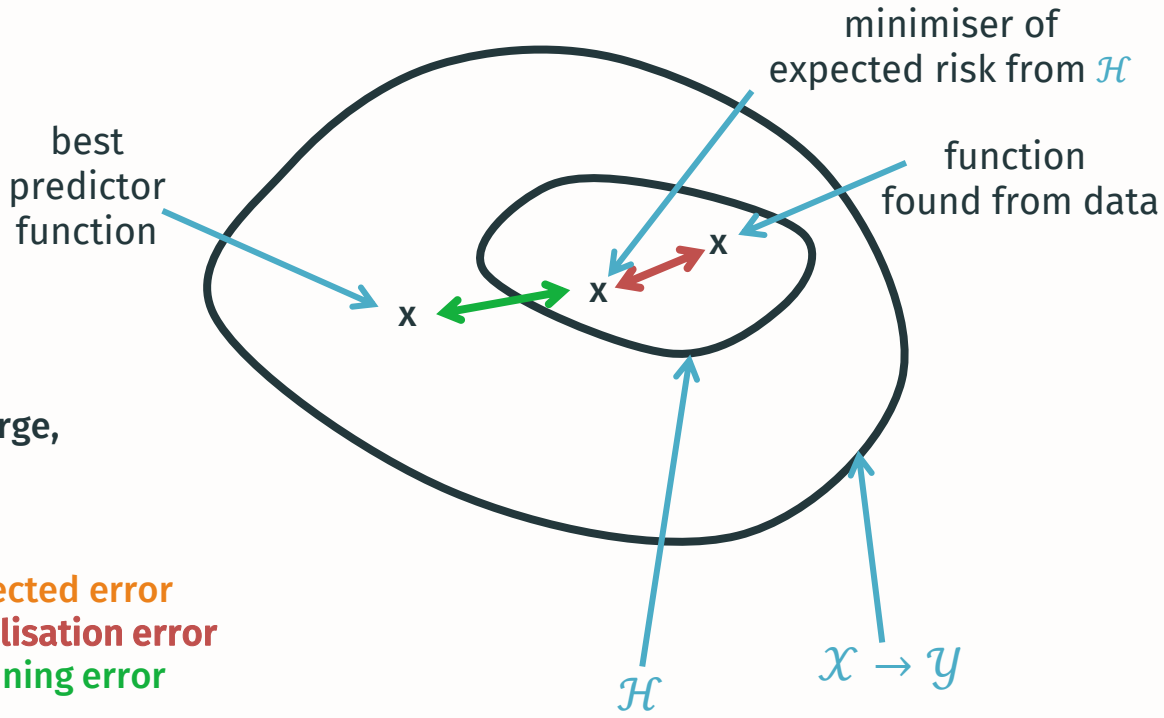
Risk Structure



Risk Structure

underfitting
function class too small,
no good function in there

overfitting
function class too large,
fitting the noise



expected error
generalisation error
training error