UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

FUZZY LOGIC IN MACHINE LEARNING

Eyke Hüllermeier
Department of Computer Science
Paderborn University

eyke@upb.de

MACHINE LEARNING

INTELLIGENT SYSTEMS

"Machine learning is the science and art of algorithms that make sense of data."

Peter Flach, 2012

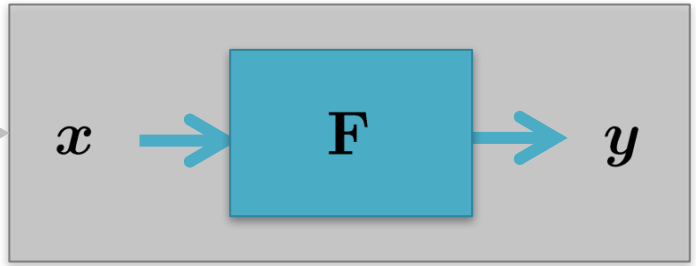
"Machine learning is the science of getting computers to act without being explicitly programmed."

Andrew Ng, 2013

THE ALGORITHMIC APPROACH

INTELLIGENT SYSTEMS

algorithm



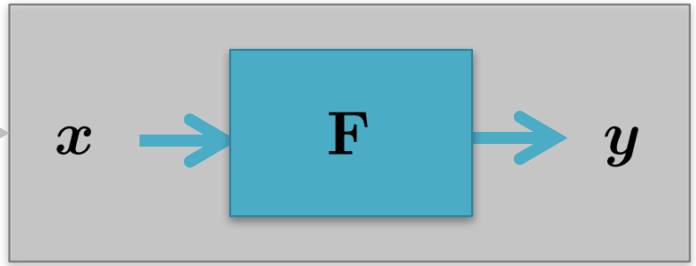
domain expert = programmer

```
ALGORITHM shortest-path(V,T)
 W := {v1}
 ShortDist[v1] :=0
 FOR each u in V - {v1}
     ShortDist[u] := T[v1,u]
 WHILE W /= V
       MinDist := INFINITE
       FOR each v in V - W
            IF ShortDist[v] < MinDist
               MinDist = ShortDist[v]
               w := v
            END {if}
       END {for}
       W := W U {w}
       FOR each u in V - W
            ShortDist[u] := Min(ShorDis[u],ShortDist[w] + T[w,u])
 END {while}
```

THE ALGORITHMIC APPROACH

domain expert = programmer

algorithm

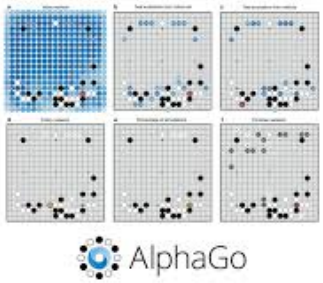


*Requires a **comprehensive understanding** and adequate formalization, not only of the problem, but also **of the solution process**.*

COMPLEX PROBLEMS

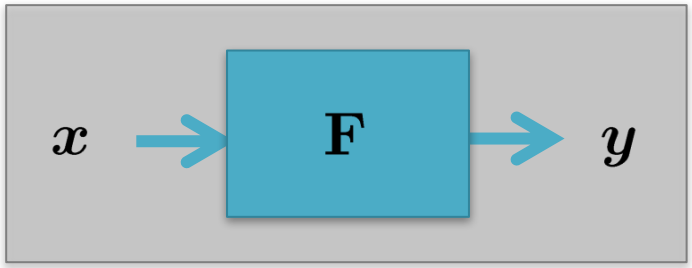
INTELLIGENT SYSTEMS

GAME PLAYING



ROBOT SOCCER

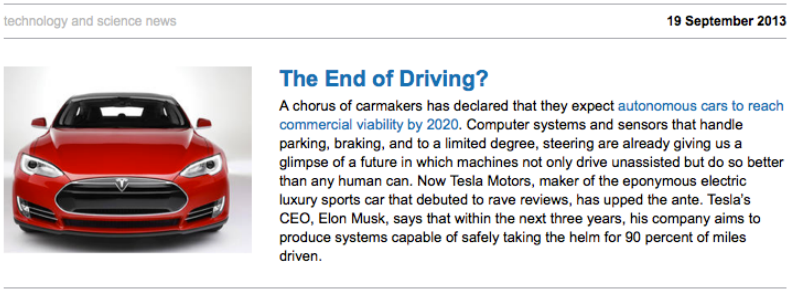
state vector describing the environment



action vector

MALE

IMAGE RECOGNITION



technology and science news 19 September 2013

The End of Driving?

A chorus of carmakers has declared that they expect autonomous cars to reach commercial viability by 2020. Computer systems and sensors that handle parking, braking, and to a limited degree, steering are already giving us a glimpse of a future in which machines not only drive unassisted but do so better than any human can. Now Tesla Motors, maker of the eponymous electric luxury sports car that debuted to rave reviews, has upped the ante. Tesla's CEO, Elon Musk, says that within the next three years, his company aims to produce systems capable of safely taking the helm for 90 percent of miles driven.

AUTONOMOUS CARS

IMPLICIT SKILLS

INTELLIGENT SYSTEMS

Human skills are not always easy to explain!

MALE
OR
FEMALE

$\boldsymbol{x} \in \mathbb{R}^N$

$\boldsymbol{y} \in \{0, 1\}$

IMPLICIT SKILLS

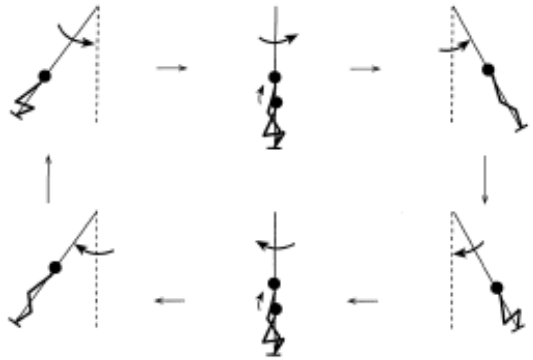
INTELLIGENT SYSTEMS

Human skills are not always easy to explain!

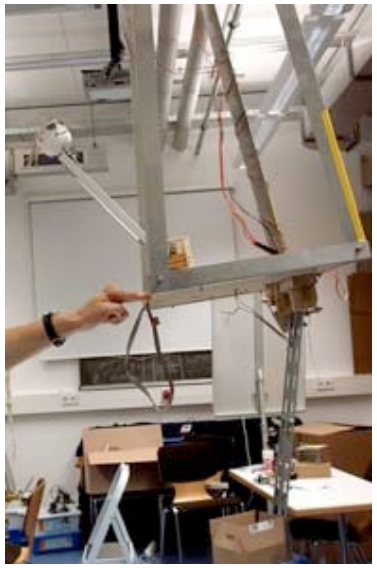
For example, a reduction of the search space does not immediately imply better solutions.

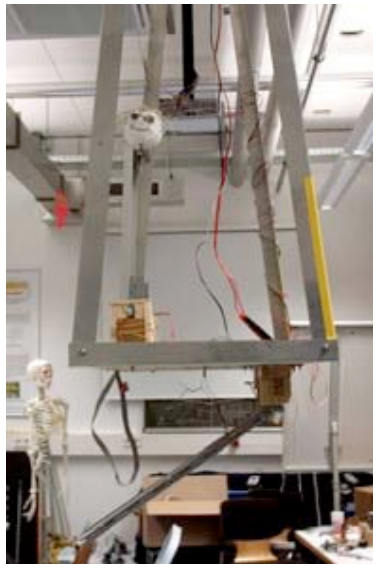
Eine Beschränkung des Suchraums führt beispielsweise nicht unmittelbar zu besseren Lösungen.

IMPLICIT SKILLS



How to teach a robot to swing?





LEARNING FROM DATA

INTELLIGENT SYSTEMS

Instead of providing a complete and consistent description of domain knowledge, it is easier to ...

- give **examples** and let the system **generalize**

→ MALE

→ *supervised learning*

- let the system **explore** and provide **feedback**

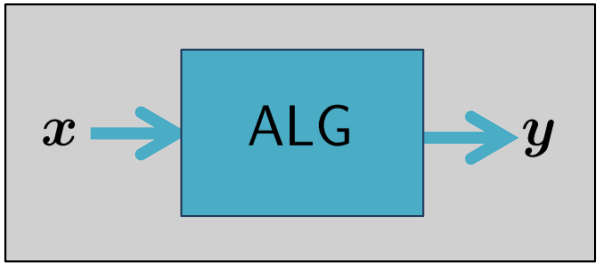
→ *reinforcement learning*

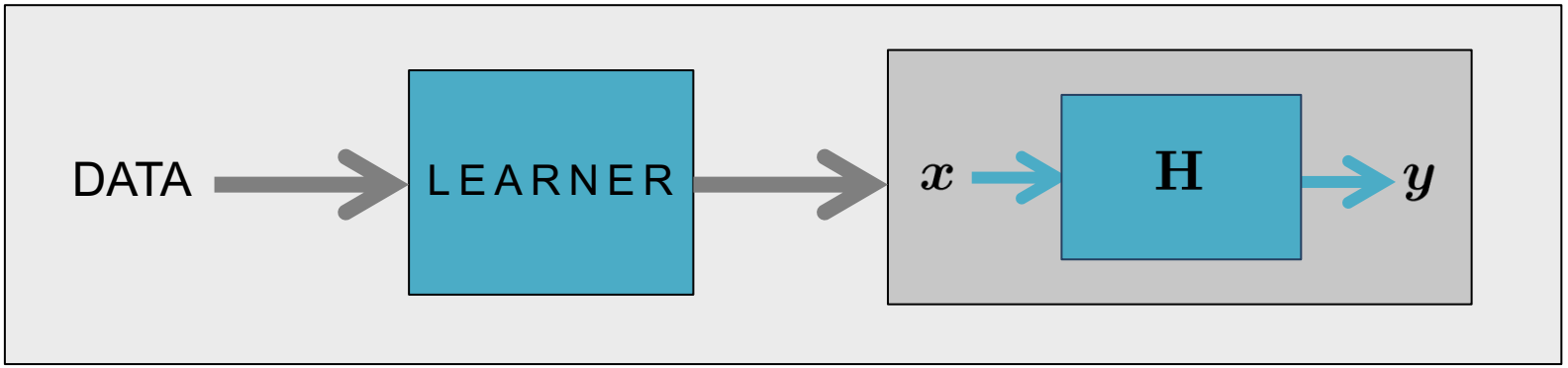
- **demonstrate** and let the system **imitate**

→ *imitation learning*

LEARNING FROM DATA

INTELLIGENT SYSTEMS



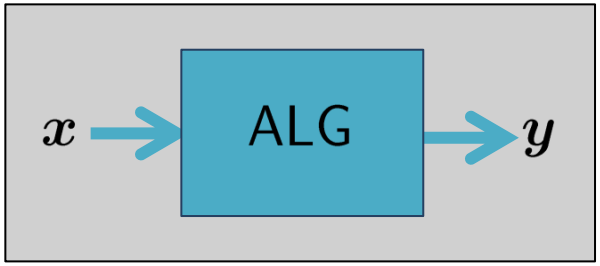


LEARNING FROM DATA

INTELLIGENT SYSTEMS

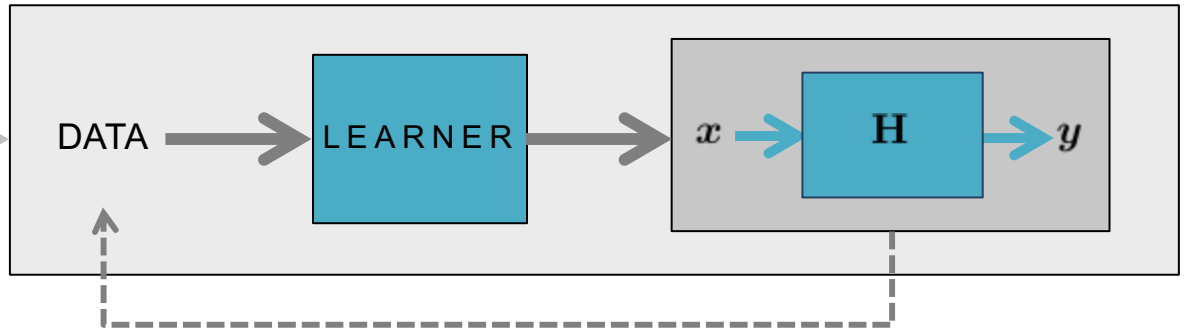
- correctness
- complexity (time, space)

computer scientist



- *correctness* (?)
- complexity (time, space)
- *sample complexity*

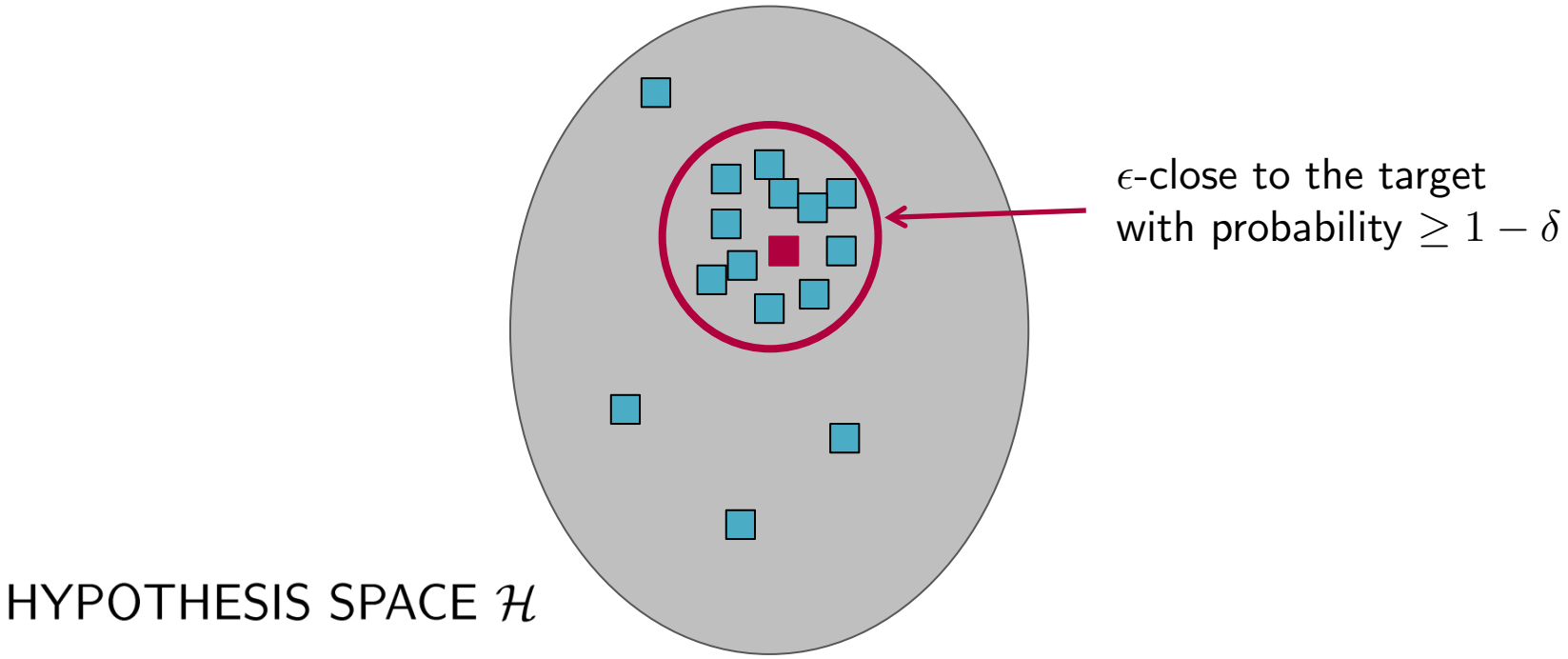
data scientist



INTELLIGENT SYSTEMS

Probably Approximately Correct (PAC) learning:

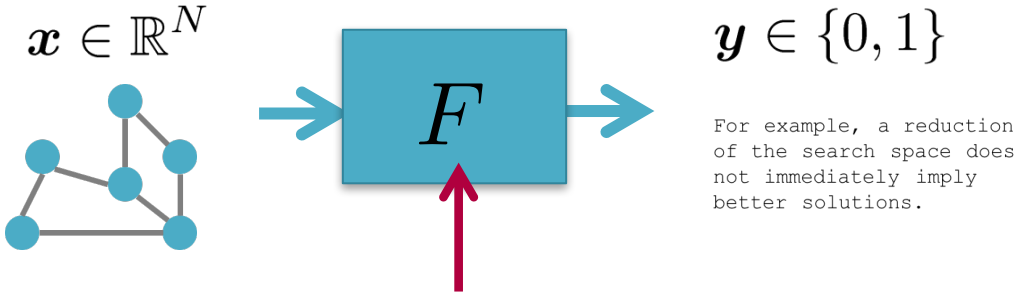
Efficiently finding a hypothesis that is „good“ with high probability!



SUPERVISED LEARNING

INTELLIGENT SYSTEMS

Machine learning is an option whenever explicitly designing an algorithm by hand appears intricate, while **data** is available that provides, in one way or the other, **useful hints** at what the sought **functionality** may look like.



*The ability to **generalize** (beyond training data) as a major theme …*

LEARNING FROM DATA

INTELLIGENT SYSTEMS

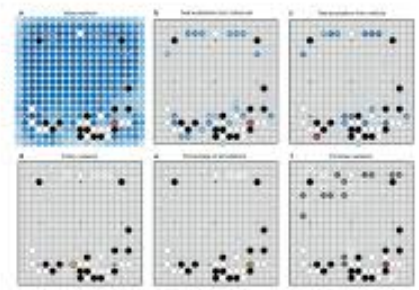
data is readily available or can easily be produced

make use of existing data

unsupervised learning

crowdsourcing

simulation, training through trial and error



reinforcement learning

amazon mechanical turk™
Artificial Artificial Intelligence

- male
- female

supervised learning

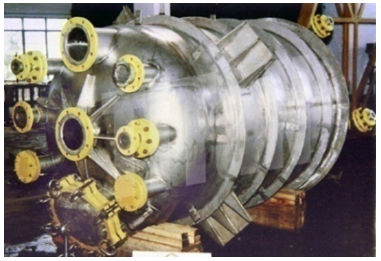
APPLICATIONS OF MACHINE LEARNING

INTELLIGENT SYSTEMS

business (CRM, response prediction, …)

smart environments

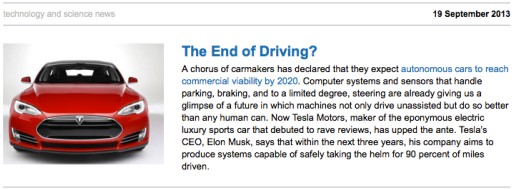
banking and finance (stock prediction, fraud detection, …)



technical systems (diagnosis, control, monitoring, …)

Internet (information retrieval, email classification, personalization, …)

biometrics (person identification, …)

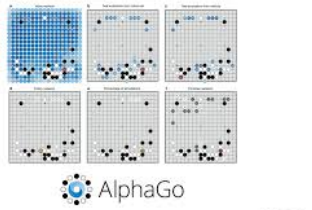


autonmous driving

medicine (diagnosis, prosthetics, ...)

media (speech/image recognition, video mining, …)

bioinformatics, genomic data analysis



games (e.g. soccer, go, …)

ANALYTIC VERSUS SYNTHETIC ML

INTELLIGENT SYSTEMS

ANALYTIC VIEW

→ *analyze and help understand a phenomenon that exists in the real world*

SYNTHETIC VIEW

→ *support the design/engineering of a system with certain desirable properties*

OUTLINE

INTELLIGENT SYSTEMS

① Prelude

② **Machine learning 101**

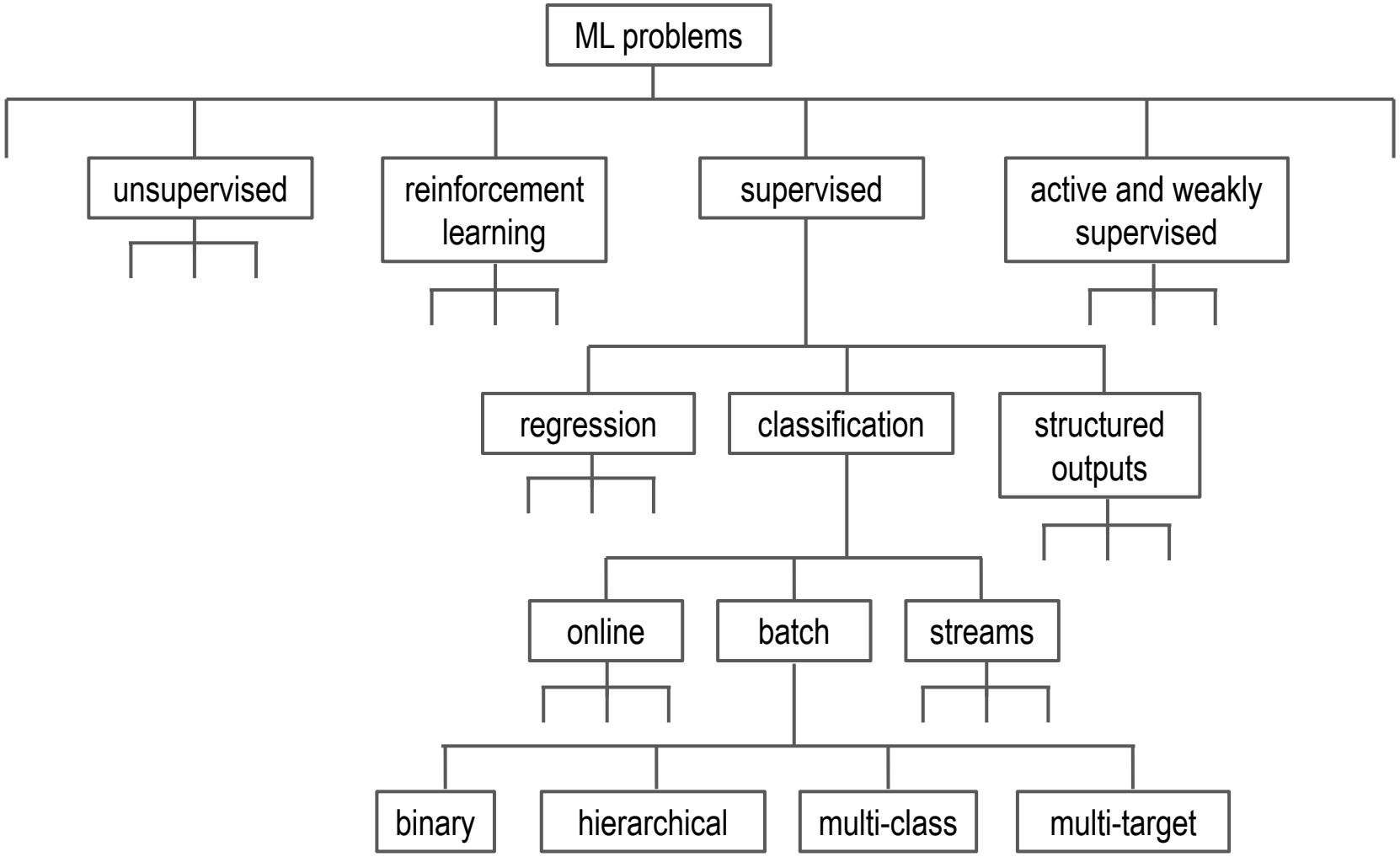
③ Potential contributions of fuzzy logic

④ Fuzzy pattern trees

⑤ Learning from fuzzy data

TAXONOMY OF ML PROBLEMS

INTELLIGENT SYSTEMS



ML PARADIGMS AND METHODOLOGIES

INTELLIGENT SYSTEMS

Learning Paradigms

- Active learning and experiment design
- Cost-sensitive learning
- Inverse reinforcement learning
- Meta learning
- Multi-task learning
- Online learning
- Reinforcement learning
- Semi-supervised learning
- Transductive learning
- Structured output prediction
- Transfer learning
- …

Machine Learning Methodologies

- Deep learning
- Gaussian processes
- Graphical models and Bayesian networks
- Inductive logic programming
- Kernel-based methods and support vector machines
- Latent variable and topic models
- Markov networks
- Preference learning and ranking
- Relational learning
- Rule and decision tree learning
- Sparsity and compressed sensing
- …

Supervised learning:

Learner is provided with explicit examples of how to act in certain situations, i.e., what outputs the target model is supposed to produce for specific inputs. Thus, the training data can be seen as examples of “correct solutions” (albeit mistakes are tolerated) that are made available by an external teacher.

Reinforcement learning:

Although feedback is provided to the learner, it is typically of an indirect nature and may come with a temporal delay. A common example is game playing, where the goal is to learn a policy that maps states (of the game) to optimal actions.

Unsupervised learning:

The learner merely observes the data (for example, handwritten digits), but without any type of supervision. The main goal in unsupervised learning is to discover structure in the data, for example represented as a grouping of data into clusters.

SUPERVISED LEARNING

INTELLIGENT SYSTEMS

EMAIL


Von Heike Wehrheim <wehrheim@upb.de>
Betreff [sfb901-tpb2] Nächstes QT-Treffen
An sfb901-tpb3@lists.upb.de, sfb901-tpb1@lists.upb.de, sfb901-tpb2@lists.upb.de, sfb901-tpc5@lists.upb.de

Antworten | Liste antworten | Weiterleiten | Archivieren | Junk | Löschen | Mehr

15/12/2016, 16:15

Liebes QT!

Im neuen Jahr sollten wir uns in unserem QT mal wieder treffen. Als Thema für das Treffen sehe ich
- Kooperationen im QT und
- Quo vadis "ML als Case Study"

Weitere Themenvorschläge nehme ich gerne entgegen.

Hier ein Doodle zur Terminfindung http://doodle.com/poll/pukqkwq8eyzma9zg

Viele Grüße und schöne Weihnachten
Heike

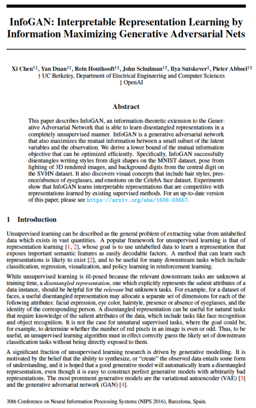
sfb901-tpb2 mailing list
sfb901-tpb2@lists.uni-paderborn.de
https://lists.uni-paderborn.de/mailman/listinfo/sfb901-tpb2

SPAM or Not SPAM

SUPERVISED LEARNING


Not Just A Black Box: Learning Important Features Through Propagating Activation Differences

accept

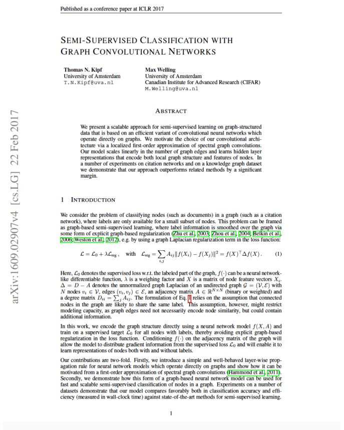

InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets

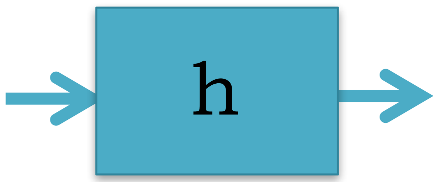
reject

....


Coupled Generative Adversarial Networks

accept

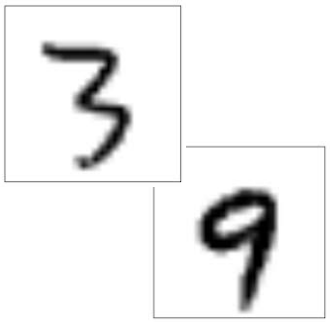

Semi-Supervised Classification with Graph Convolutional Networks



{ accept, reject }

SUPERVISED LEARNING

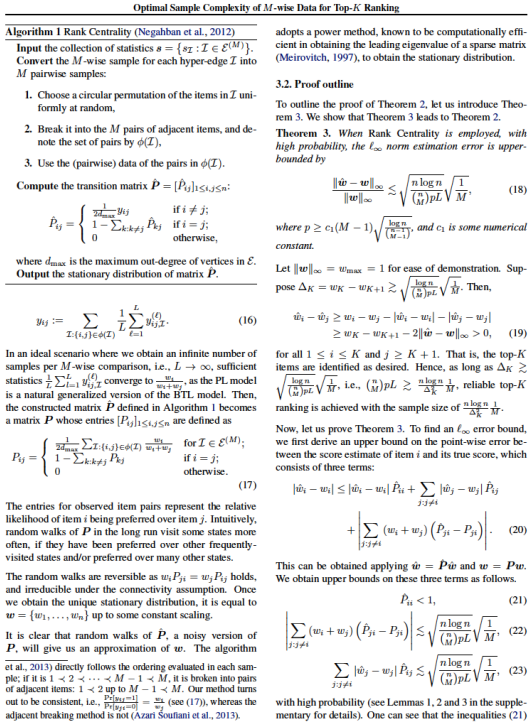
INTELLIGENT SYSTEMS



{0,1, ..., 9}

SUPERVISED LEARNING

INTELLIGENT SYSTEMS



Optimal Sample Complexity of M-wise Data for Top-K Ranking

Algorithm 1 Rank Centrality (Negahban et al., 2012)

Input the collection of statistics $\boldsymbol{s} = \{s_{\mathcal{I}} : \mathcal{I} \in \mathcal{E}^{(M)}\}$.
Convert the M-wise sample for each hyper-edge $\mathcal{I}$ into M pairwise samples:

1. Choose a circular permutation of the items in $\mathcal{I}$ uniformly at random,
2. Break it into the M pairs of adjacent items, and denote the set of pairs by $\phi(\mathcal{I})$,
3. Use the (pairwise) data of the pairs in $\phi(\mathcal{I})$.

Compute the transition matrix $\hat{\boldsymbol{P}} = [\hat{P}_{ij}]_{1\le i,j\le n}$:

$$\hat{P}_{ij} = \begin{cases} \frac{1}{2d_{\max}} y_{ij} & \text{if } i \neq j; \\ 1 - \sum_{k:k\neq j} \hat{P}_{kj} & \text{if } i = j; \\ 0 & \text{otherwise,} \end{cases}$$

where $d_{\max}$ is the maximum out-degree of vertices in $\mathcal{E}$.
Output the stationary distribution of matrix $\hat{\boldsymbol{P}}$.

$$y_{ij} := \sum_{\mathcal{I}:\{i,j\}\in\phi(\mathcal{I})} \frac{1}{L} \sum_{\ell=1}^{L} y_{ij,\mathcal{I}}^{(\ell)}. \tag{16}$$

In an ideal scenario where we obtain an infinite number of samples per M-wise comparison, i.e., $L \to \infty$, sufficient statistics $\frac{1}{L}\sum_{\ell=1}^{L} y_{ij,\mathcal{I}}^{(\ell)}$ converge to $\frac{w_i}{w_i+w_j}$, as the PL model is a natural generalized version of the BTL model. Then, the constructed matrix $\hat{\boldsymbol{P}}$ defined in Algorithm 1 becomes a matrix $\boldsymbol{P}$ whose entries $[P_{ij}]_{1\le i,j\le n}$ are defined as

$$P_{ij} = \begin{cases} \frac{1}{2d_{\max}} \sum_{\mathcal{I}:\{i,j\}\in\phi(\mathcal{I})} \frac{w_i}{w_i+w_j} & \text{for } \mathcal{I} \in \mathcal{E}^{(M)}; \\ 1 - \sum_{k:k\neq j} P_{kj} & \text{if } i = j; \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

The entries for observed item pairs represent the relative likelihood of item i being preferred over item j. Intuitively, random walks of $\boldsymbol{P}$ in the long run visit some states more often, if they have been preferred over other frequently-visited states and/or preferred over many other states.

The random walks are reversible as $w_i P_{ji} = w_j P_{ij}$ holds, and irreducible under the connectivity assumption. Once we obtain the unique stationary distribution, it is equal to $\boldsymbol{w} = \{w_1, \dots, w_n\}$ up to some constant scaling.

It is clear that random walks of $\hat{\boldsymbol{P}}$, a noisy version of $\boldsymbol{P}$, will give us an approximation of $\boldsymbol{w}$. The algorithm adopts a power method, known to be computationally efficient in obtaining the leading eigenvalue of a sparse matrix (Meirovitch, 1997), to obtain the stationary distribution.

3.2. Proof outline

To outline the proof of Theorem 2, let us introduce Theorem 3. We show that Theorem 3 leads to Theorem 2.

Theorem 3. *When* Rank Centrality *is employed, with high probability, the ℓ_∞ norm estimation error is upper-bounded by*

$$\frac{\|\hat{\boldsymbol{w}} - \boldsymbol{w}\|_\infty}{\|\boldsymbol{w}\|_\infty} \lesssim \sqrt{\frac{n\log n}{\binom{n}{M}pL}} \sqrt{\frac{1}{M}}, \tag{18}$$

where $p \ge c_1(M-1)\sqrt{\frac{\log n}{\binom{n}{M-1}}}$, and c_1 is some numerical constant.

Let $\|\boldsymbol{w}\|_\infty = w_{\max} = 1$ for ease of demonstration. Suppose $\Delta_K = w_K - w_{K+1} \gtrsim \sqrt{\frac{\log n}{\binom{n}{M}pL}} \sqrt{\frac{1}{M}}$. Then,

$$\begin{aligned} \hat{w}_i - \hat{w}_j &\ge w_i - w_j - |\hat{w}_i - w_i| - |\hat{w}_j - w_j| \\ &\ge w_K - w_{K+1} - 2\|\hat{\boldsymbol{w}} - \boldsymbol{w}\|_\infty > 0, \end{aligned} \tag{19}$$

for all $1 \le i \le K$ and $j \ge K+1$. That is, the top-K items are identified as desired. Hence, as long as $\Delta_K \gtrsim \sqrt{\frac{\log n}{\binom{n}{M}pL}} \sqrt{\frac{1}{M}}$, i.e., $\binom{n}{M}pL \gtrsim \frac{n\log n}{\Delta_K^2}\frac{1}{M}$, reliable top-$K$ ranking is achieved with the sample size of $\frac{n\log n}{\Delta_K^2}\frac{1}{M}$.

Now, let us prove Theorem 3. To find an ℓ_∞ error bound, we first derive an upper bound on the point-wise error between the score estimate of item i and its true score, which consists of three terms:

$$\begin{aligned} |\hat{w}_i - w_i| \le |\hat{w}_i - w_i|\hat{P}_{ii} + \sum_{j:j\neq i} |\hat{w}_j - w_j|\hat{P}_{ij} \\ + \left| \sum_{j:j\neq i} (w_i + w_j)\left(\hat{P}_{ji} - P_{ji}\right) \right|. \end{aligned} \tag{20}$$

This can be obtained applying $\hat{\boldsymbol{w}} = \hat{\boldsymbol{P}}\hat{\boldsymbol{w}}$ and $\boldsymbol{w} = \boldsymbol{P}\boldsymbol{w}$. We obtain upper bounds on these three terms as follows.

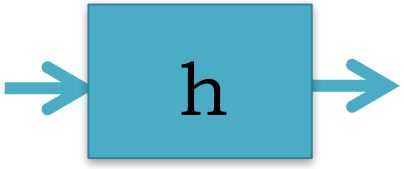
$$\hat{P}_{ii} < 1, \tag{21}$$

$$\left| \sum_{j:j\neq i} (w_i + w_j)\left(\hat{P}_{ji} - P_{ji}\right) \right| \lesssim \sqrt{\frac{n\log n}{\binom{n}{M}pL}} \sqrt{\frac{1}{M}}, \tag{22}$$

$$\sum_{j:j\neq i} |\hat{w}_j - w_j|\hat{P}_{ij} \lesssim \sqrt{\frac{n\log n}{\binom{n}{M}pL}} \sqrt{\frac{1}{M}}, \tag{23}$$

with high probability (see Lemmas 1, 2 and 3 in the supplementary for details). One can see that the inequalities (21)

et al., 2013) directly follows the ordering evaluated in each sample; if it is $1 \prec 2 \prec \cdots \prec M-1 \prec M$, it is broken into pairs of adjacent items: $1 \prec 2$ up to $M-1 \prec M$. Our method turns out to be consistent, i.e., $\frac{P(y_{ij}=1)}{P(y_{ji}=1)} = \frac{w_i}{w_j}$ (see (17)), whereas the adjacent breaking method is not (Azari Soufiani et al., 2013).



Abstract

Given a sample of instances with binary labels, the top ranking problem is to produce a ranked list of instances where the *head* of the list is dominated by positives. Popular existing approaches to this problem are based on surrogates to a performance measure known as the fraction of positives of the top (PTop). In this paper, we show that the measure and its surrogates have an undesirable property: for certain noisy distributions, it is optimal to trivially predict *the same score for all instances*. We propose a simple rectification of the measure which avoids such trivial solutions, while still focussing on the head of the ranked list and being as easy to optimise.

Given a set of (i.i.d.) **training data**

$$\mathcal{D} = \Big\{ (\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N) \Big\} \subset \mathcal{X} \times \mathcal{Y}$$

and a **hypothesis space** $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$, find a model with low **risk**

$$\mathcal{R}(h) = \int_{\mathcal{X} \times \mathcal{Y}} L\big(h(\boldsymbol{x}_i), y_i\big)\, d\,\mathbf{P}(\boldsymbol{x}, y)\,.$$

loss function

data generating process

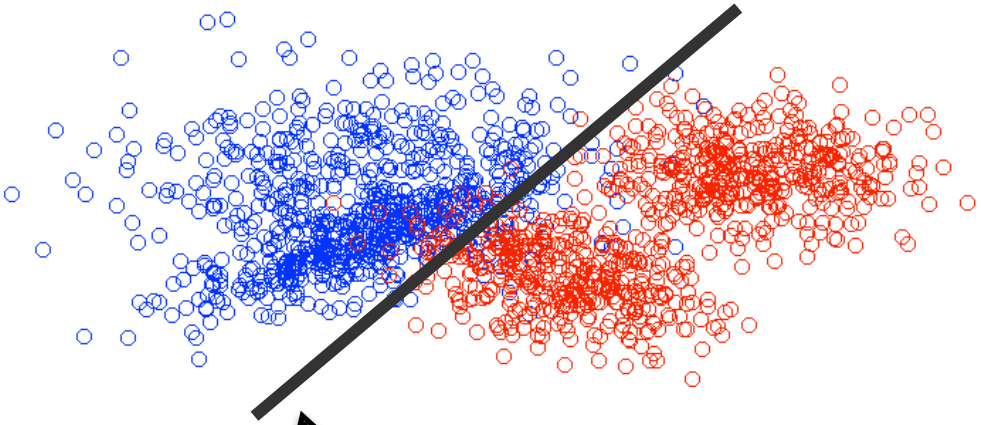
Other criteria might be important, too …

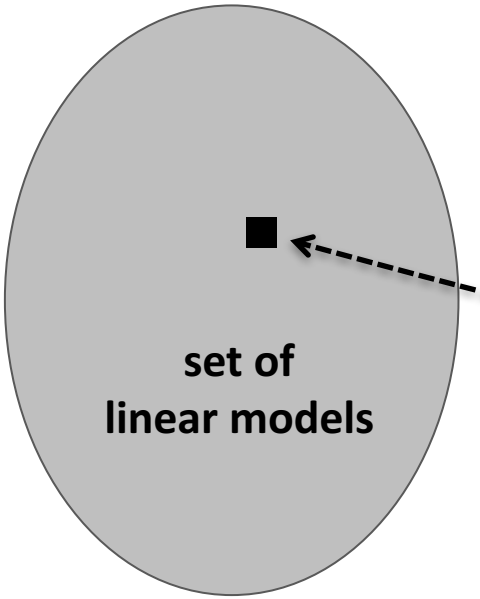
LINEAR MODELS

INTELLIGENT SYSTEMS

$$\mathcal{H} = \left\{ h : \boldsymbol{x} \mapsto [\![\boldsymbol{x}^\top \boldsymbol{w} > 0]\!] \mid \boldsymbol{w} \in \mathbb{R}^{d+1} \right\}$$

$$L(y, \hat{y}) = \begin{cases} 0 & \text{if } y = \hat{y} \\ 1 & \text{if } y \neq \hat{y} \end{cases}$$

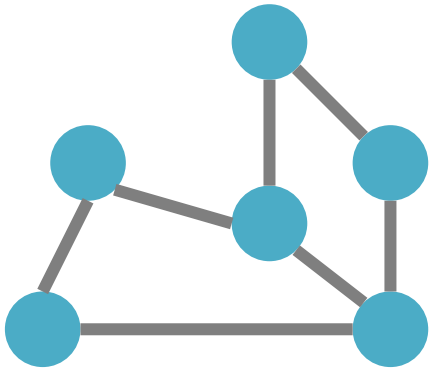




$$h^* \in \arg\min_{h \in \mathcal{H}} \int_{\mathcal{X} \times \mathcal{Y}} L\big(h(\boldsymbol{x}_i), y_i\big)\, d\,\mathbf{P}(\boldsymbol{x}, y)\,.$$

HYPOTHESIS SPACE $\mathcal{H}$

STRUCTURED DATA



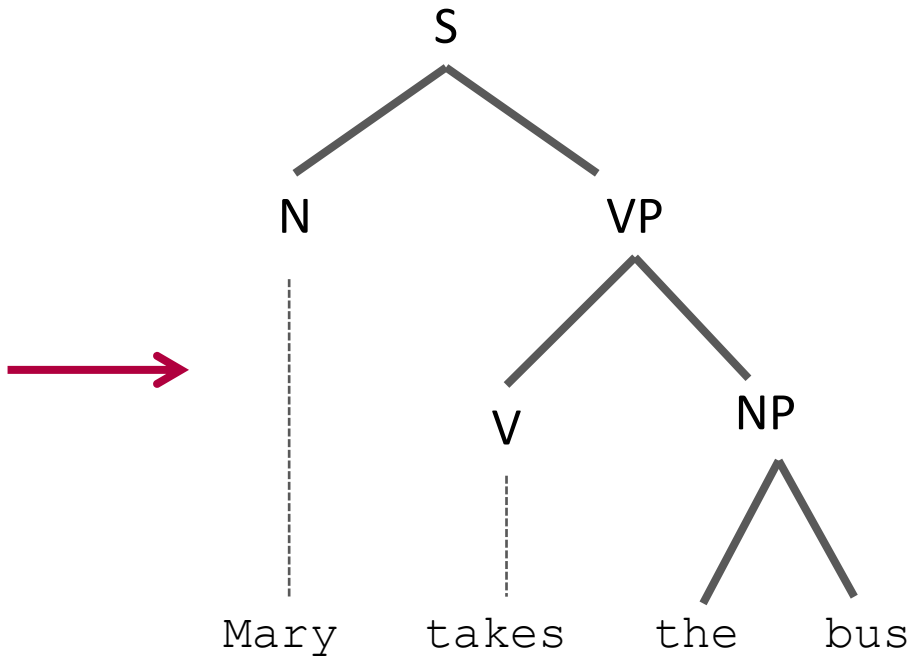
→ { active, non-active }

graph (molecule) → *class*

STRUCTURED DATA

INTELLIGENT SYSTEMS

(Mary takes the bus)



sequence → *tree*

INTELLIGENT SYSTEMS

Many ML algorithms operate in Euclidean spaces …

EMAIL


Von Heike Wehrheim <wehrheim@upb.de>
Antworten | Liste antworten | Weiterleiten | Archivieren | Junk | Löschen | Mehr
Betreff [sfb901-tpb2] Nächstes QT-Treffen
15/12/2016, 16:15
An sfb901-tpb3@lists.upb.de, sfb901-tpb1@lists.upb.de, sfb901-tpb2@lists.upb.de, sfb901-tpc5@lists.upb.de

Liebes QT!

Im neuen Jahr sollten wir uns in unserem QT mal wieder treffen. Als Thema für das Treffen sehe ich
- Kooperationen im QT und
- Quo vadis "ML als Case Study"

Weitere Themenvorschläge nehme ich gerne entgegen.

Hier ein Doodle zur Terminfindung http://doodle.com/poll/pukqkwq8eyzma9zg

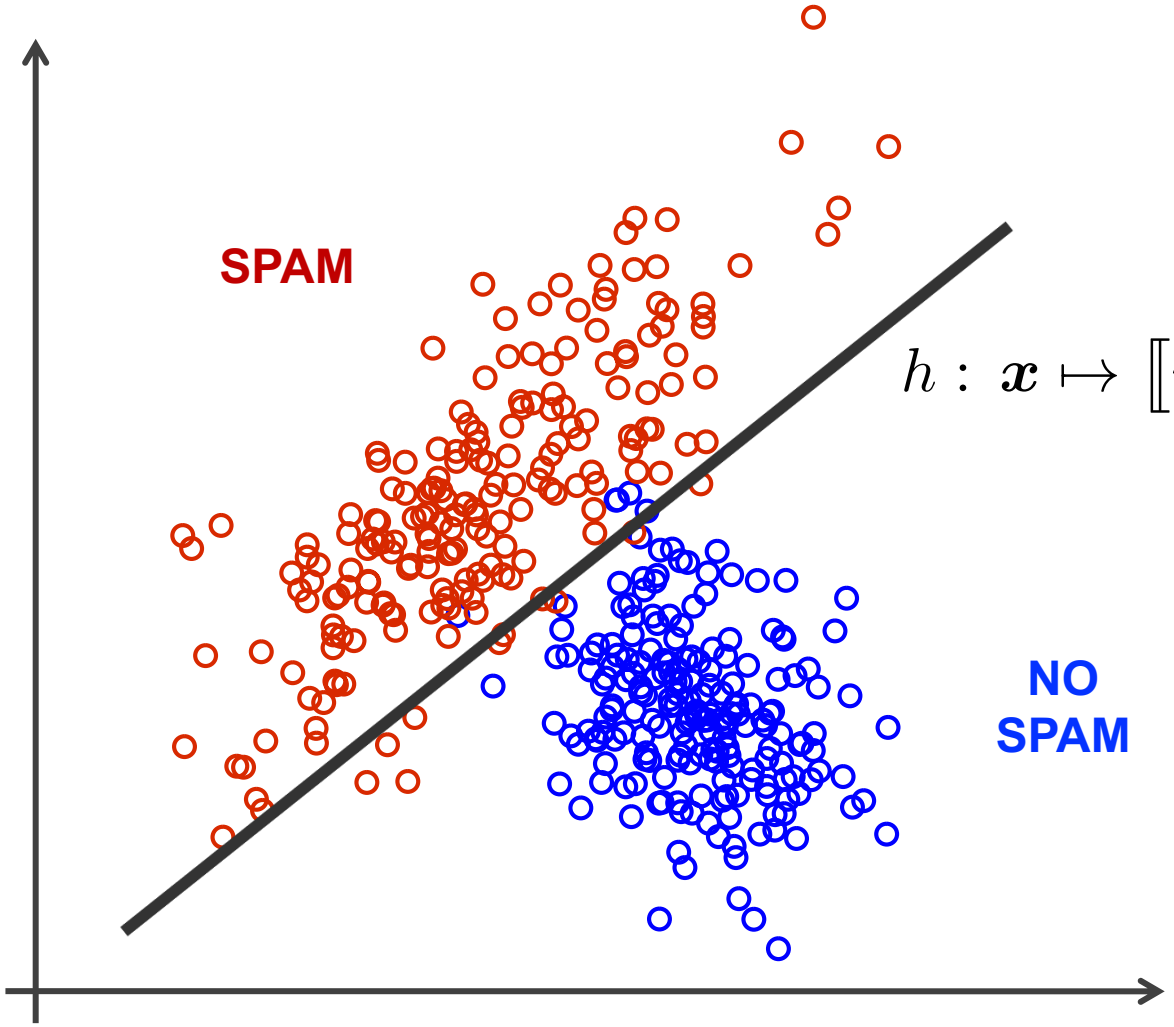
Viele Grüße und schöne Weihnachten
Heike

sfb901-tpb2 mailing list
sfb901-tpb2@lists.uni-paderborn.de
https://lists.uni-paderborn.de/mailman/listinfo/sfb901-tpb2

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$$

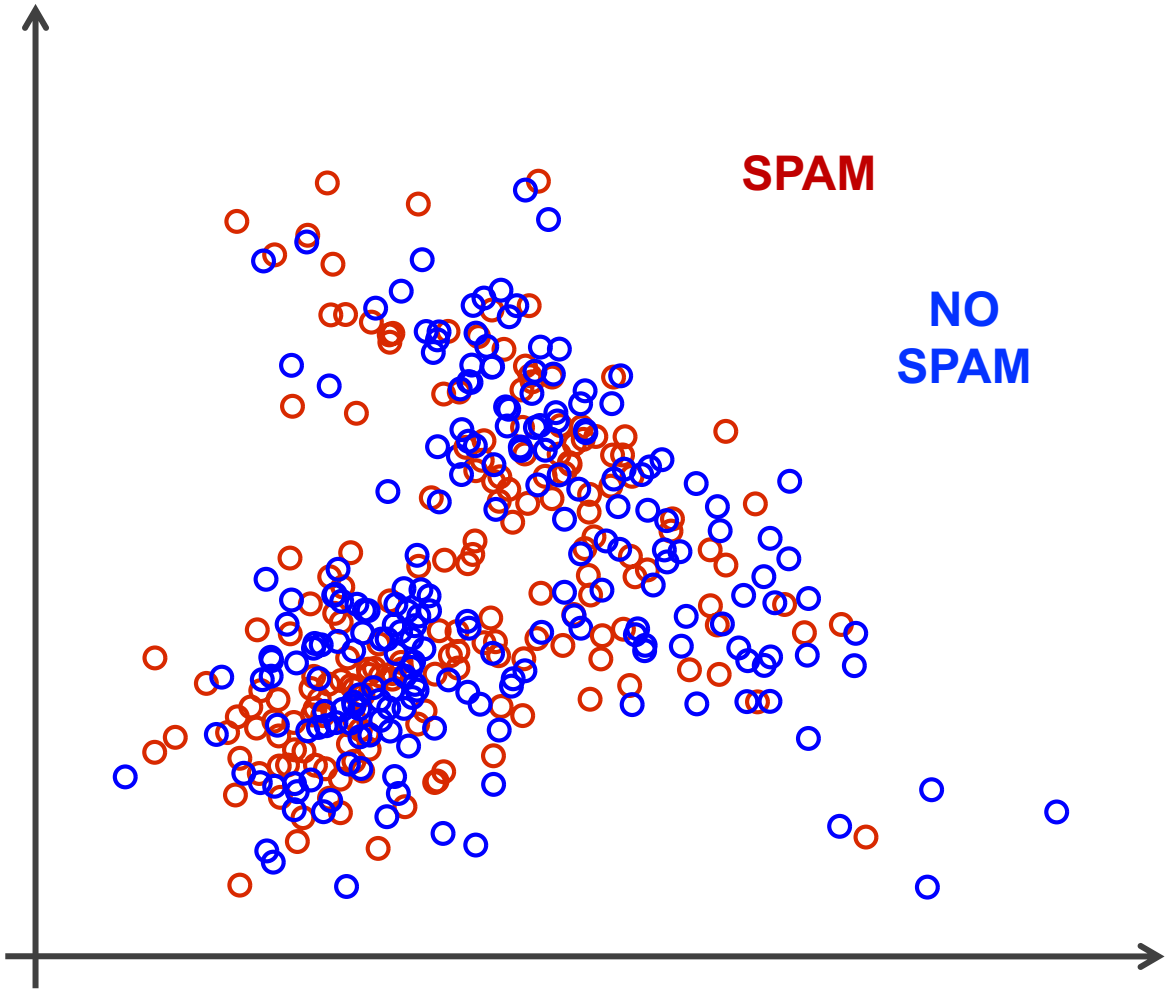
INTELLIGENT SYSTEMS

Data entities represented as **feature vectors:**



INTELLIGENT SYSTEMS

Data entities represented as **feature vectors:**



Given a set of (i.i.d.) **training data**

$$\mathcal{D} = \Big\{ (\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N) \Big\} \subset \mathcal{X} \times \mathcal{Y}$$

and a **hypothesis space** $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$, find a model with low **empirical risk**

$$\mathcal{R}_{emp}(h) = \frac{1}{N} \sum_{i=1}^{N} \ell\big(h(\boldsymbol{x}_i), y_i\big) \,,$$

i.e.,

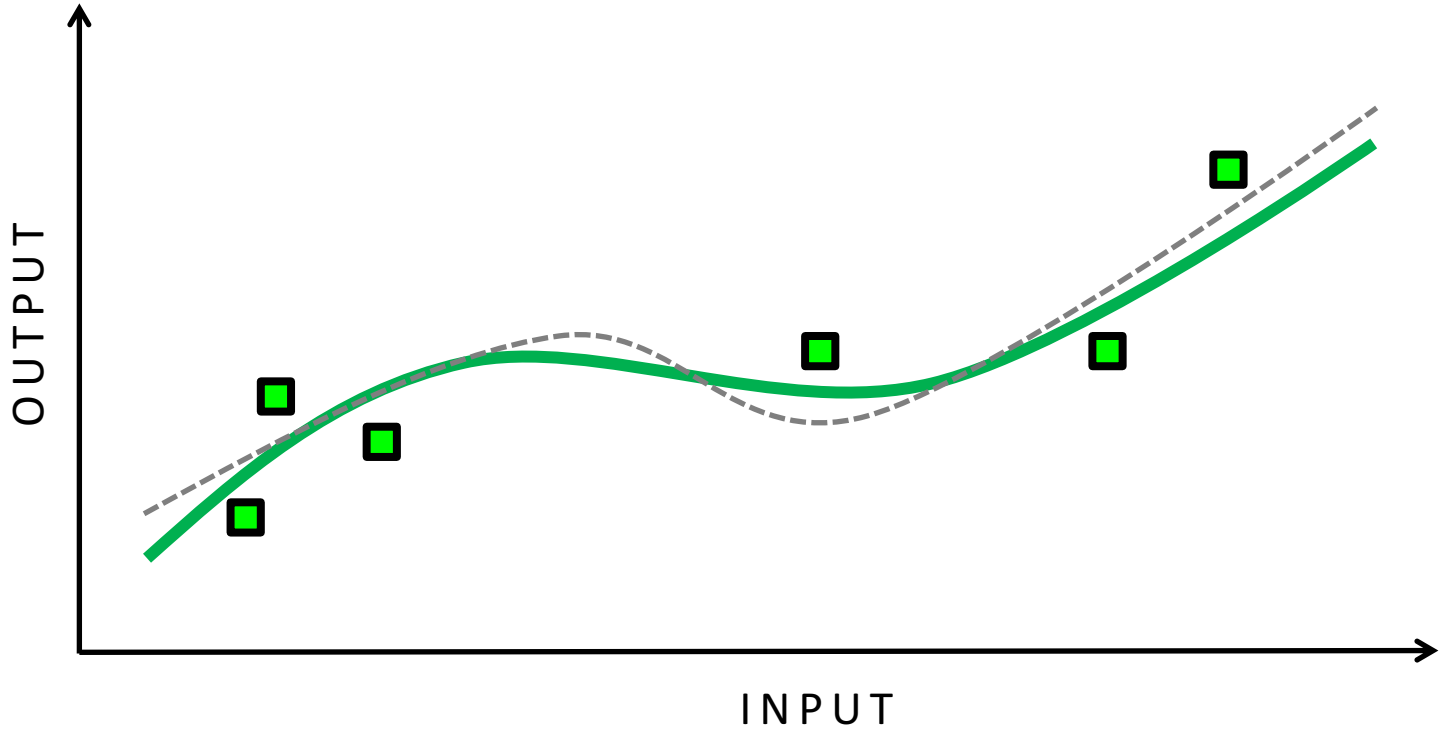
$$h^* \in \operatorname*{argmin}_{h \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \ell\big(h(\boldsymbol{x}_i), y_i\big) \,.$$

In general, ERM won't work very well …

CHOICE OF THE MODEL SPACE

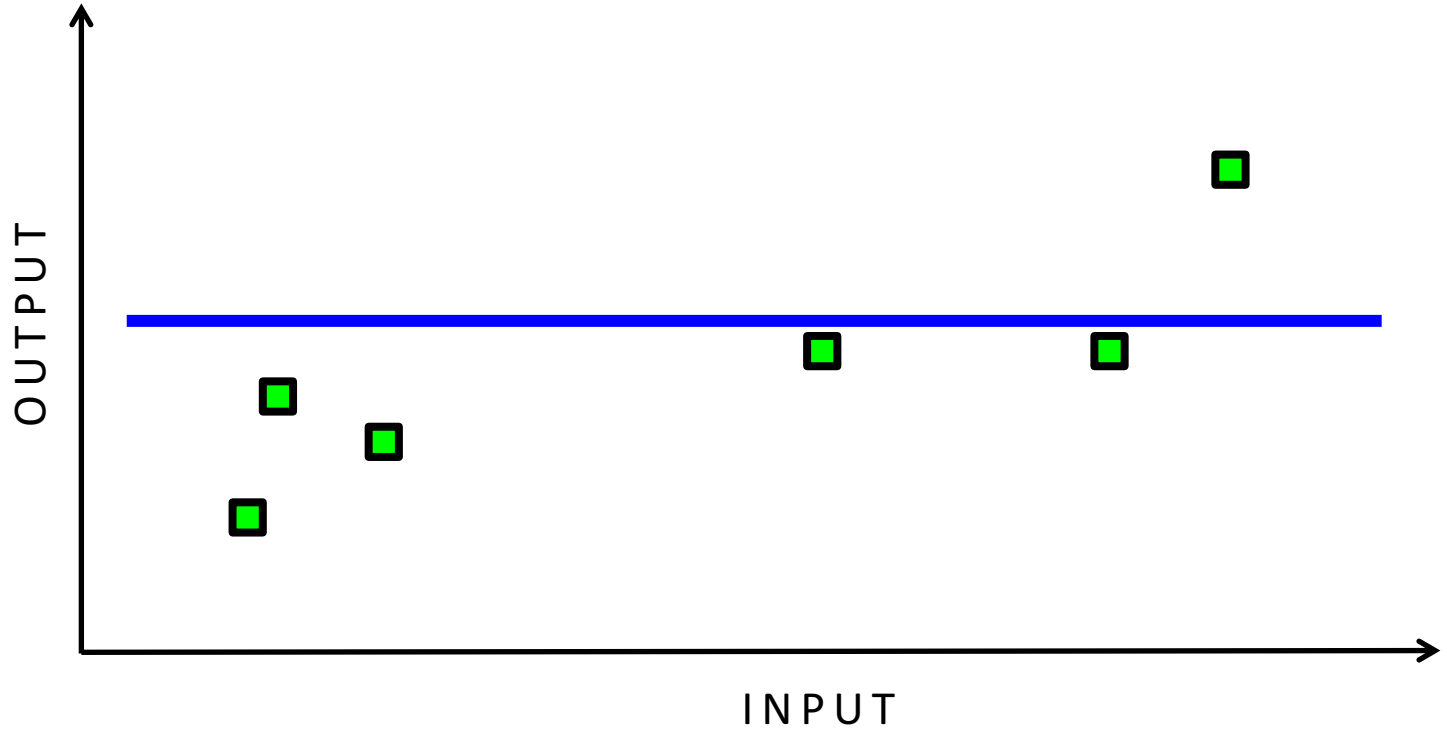
INTELLIGENT SYSTEMS

$$Y = f(X) + \epsilon$$



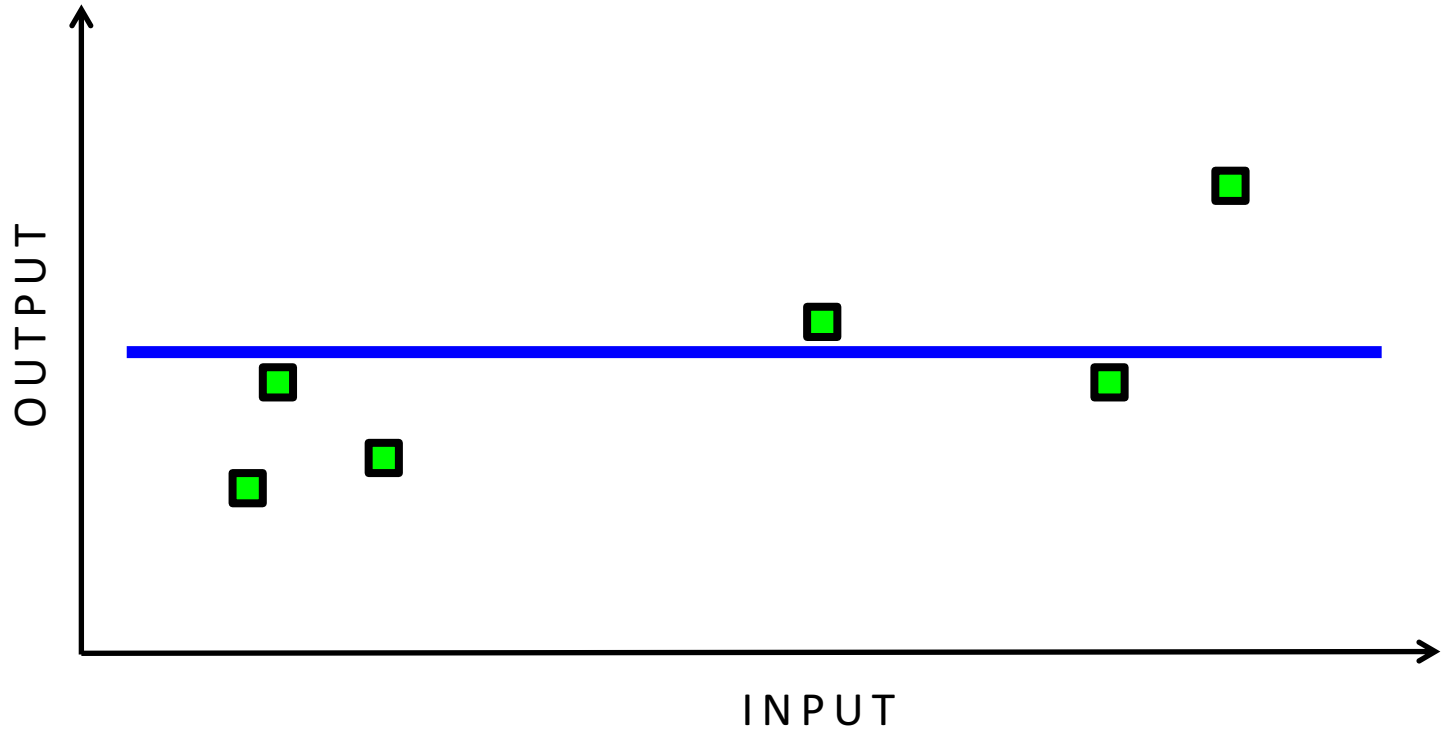
CHOICE OF THE MODEL SPACE

INTELLIGENT SYSTEMS



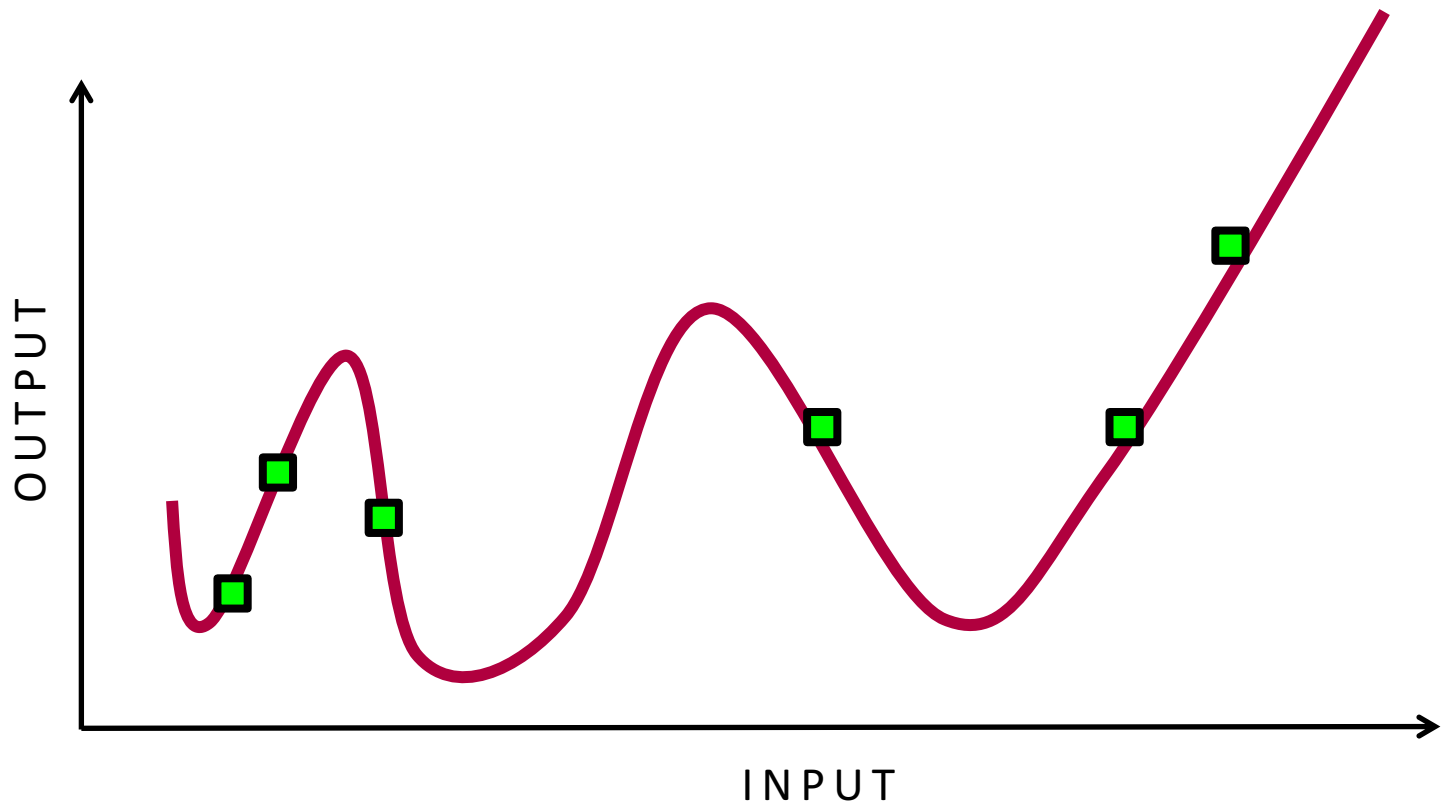
CHOICE OF THE MODEL SPACE

INTELLIGENT SYSTEMS



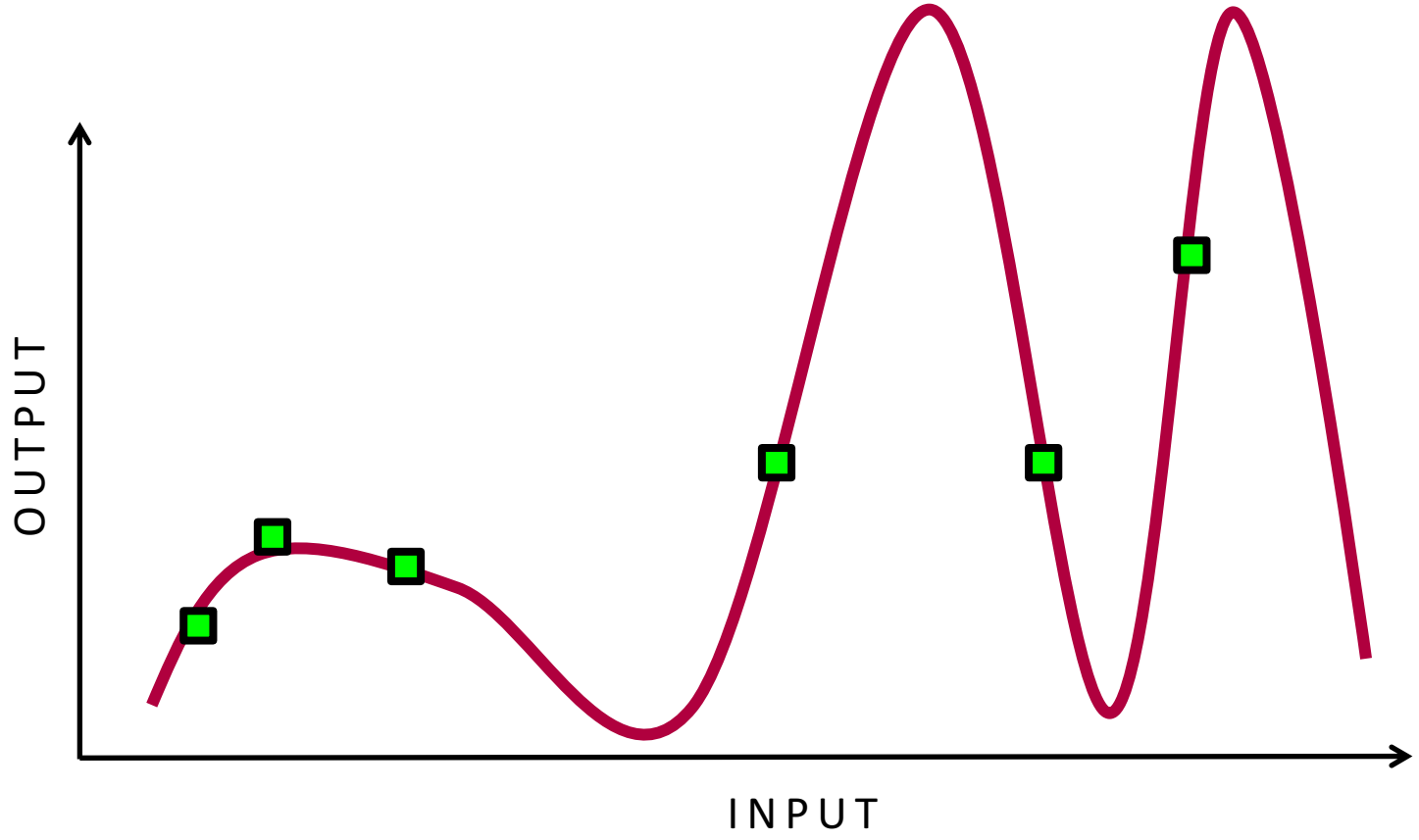
CHOICE OF THE MODEL SPACE

INTELLIGENT SYSTEMS



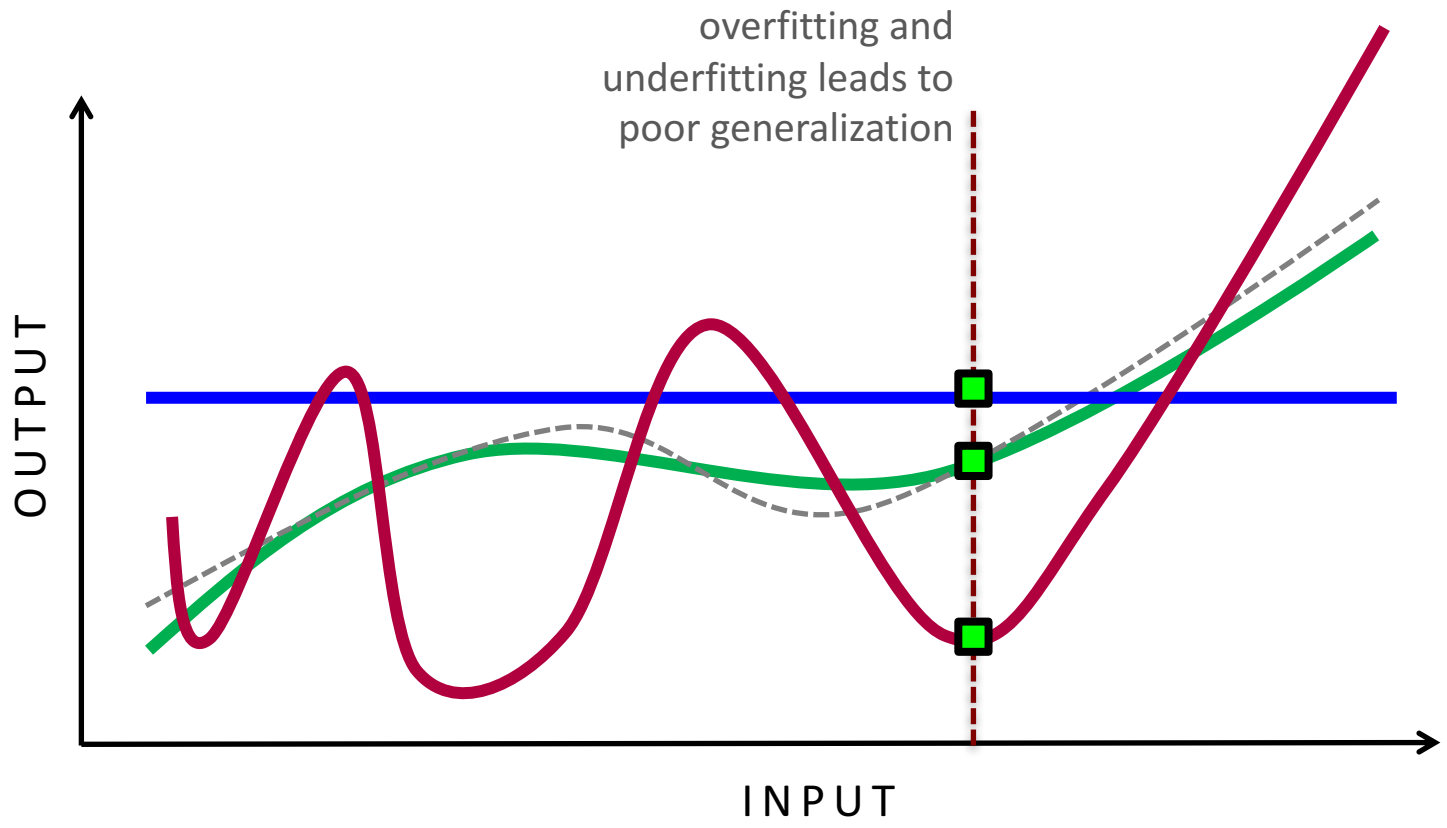
CHOICE OF THE MODEL SPACE

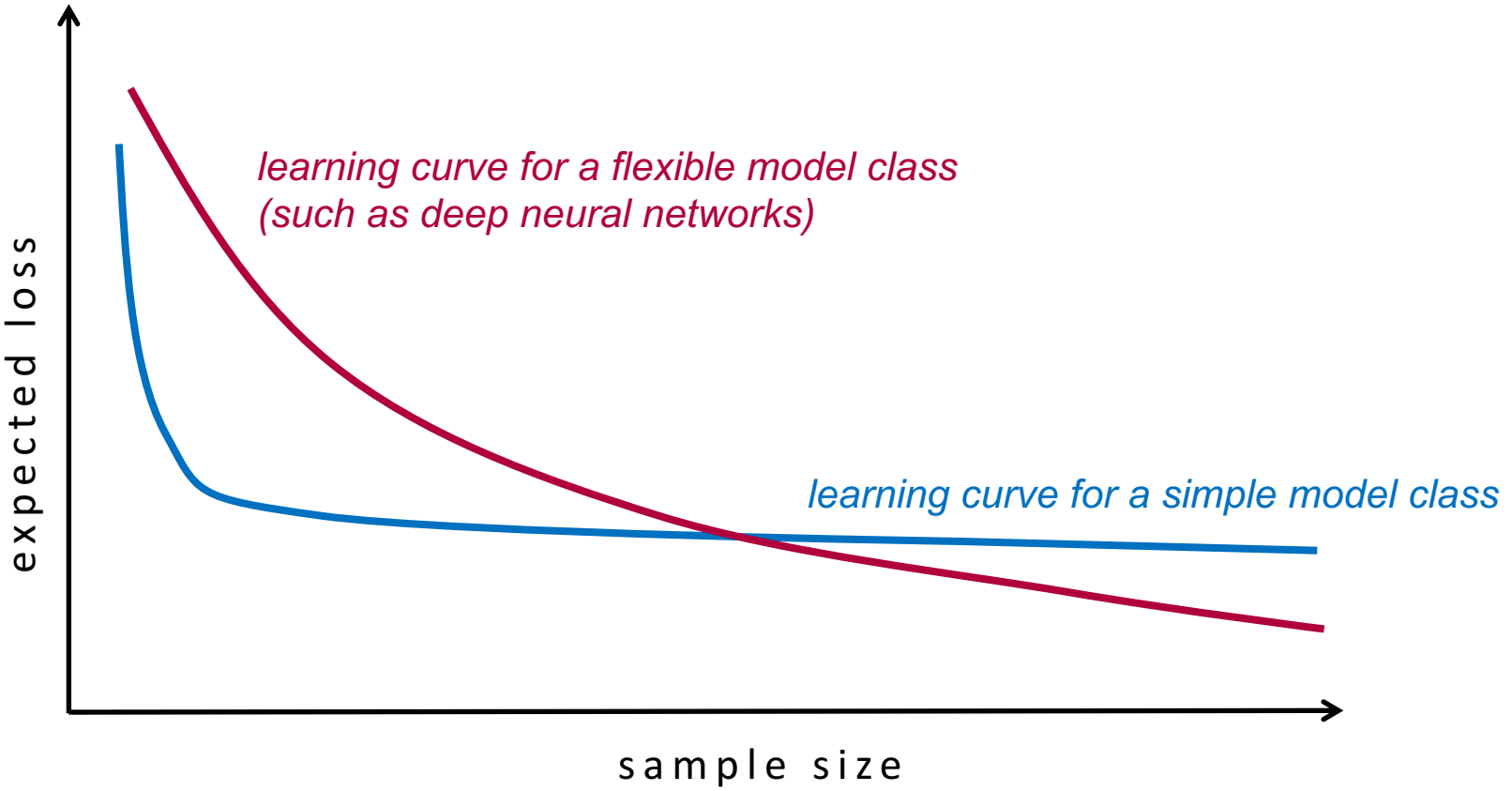
INTELLIGENT SYSTEMS



CHOICE OF THE MODEL SPACE

INTELLIGENT SYSTEMS




learning curve for a flexible model class
(such as deep neural networks)
learning curve for a simple model class
expected loss
sample size

Given a set of (i.i.d.) **training data**

$$\mathcal{D} = \Big\{ (\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N) \Big\} \subset \mathcal{X} \times \mathcal{Y}$$

and a **hypothesis space** $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$, find a model minimizing

$$\mathcal{R}_{reg}(h) = \frac{1}{N} \sum_{i=1}^{N} L\big(h(\boldsymbol{x}_i), y_i\big) + \lambda\, \Phi(h)\,,$$

regularizer

for example

$$\mathcal{R}_{reg}(h) = \frac{1}{N} \sum_{i=1}^{N} \big(\boldsymbol{w}^\top \boldsymbol{x}_i - y_i\big)^2 + \lambda\, \|\boldsymbol{w}\|_2\,.$$

Consider hypotheses $h_{\boldsymbol{w}}(\cdot)$ parameterized by $\boldsymbol{w} \in \mathbb{R}^d$, such as

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} \ .$$

Gradient descent:

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \gamma \, \frac{1}{N} \sum_{i=1}^{N} \nabla_{\boldsymbol{w}} \, \ell\big(h_{\boldsymbol{w}_t}(\boldsymbol{x}_i), y_i\big)$$

Stochastic gradient descent:

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \gamma \, \frac{1}{N} \nabla_{\boldsymbol{w}} \, \ell\big(h_{\boldsymbol{w}_t}(\boldsymbol{x}_t), y_t\big)$$

for randomly chosen $(\boldsymbol{x}_t, y_t)$.

SURROGATE AND MARGIN LOSSES

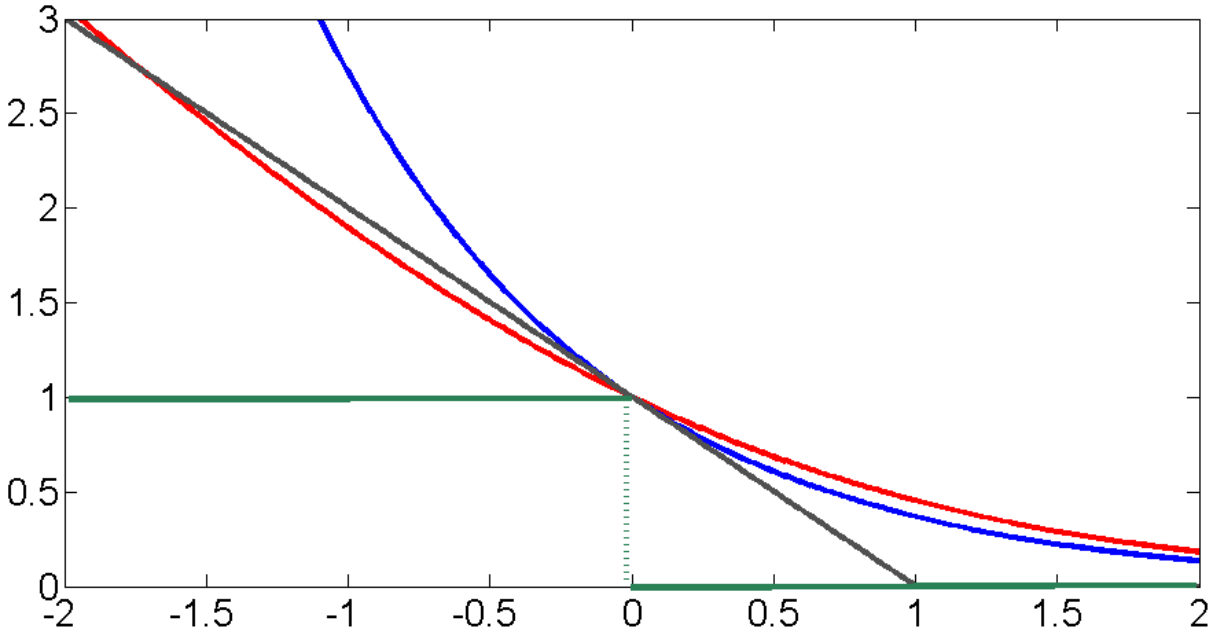
INTELLIGENT SYSTEMS

Let $\mathcal{Y} = \{-1, +1\}$ and consider a class of scoring classifiers $h : \mathcal{X} \longrightarrow \mathbb{R}$.

A **margin loss** is a function of the form

$$\ell(y, s) = f(ys) \ ,$$

where $f : \mathbb{R} \longrightarrow \mathbb{R}$ is a non-increasing function.



0/1 loss

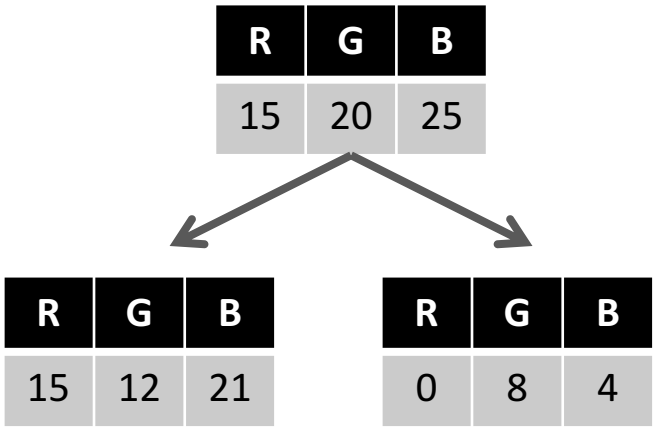
logistic

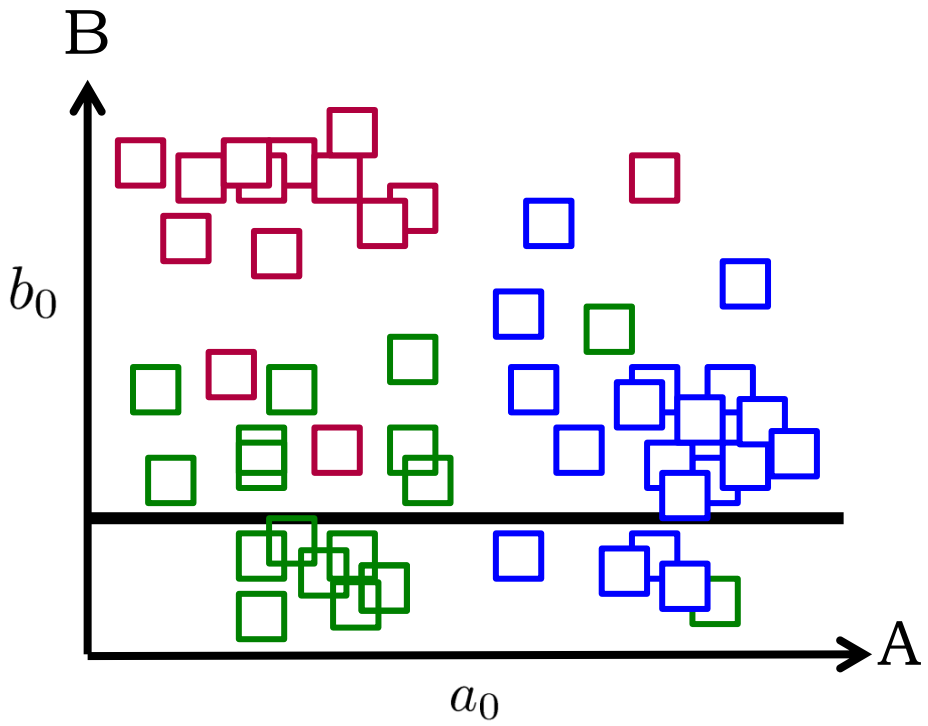
hinge

exponential

INTELLIGENT SYSTEMS

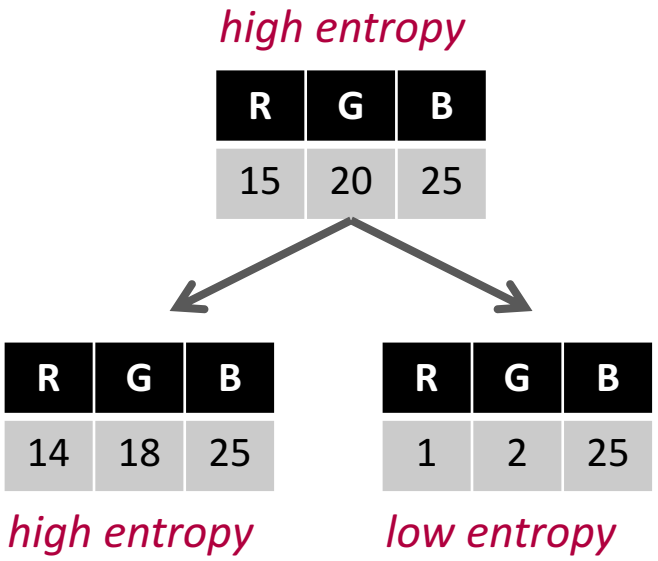
Decision trees define a classification function through **recursive partitioning** of the instance space.

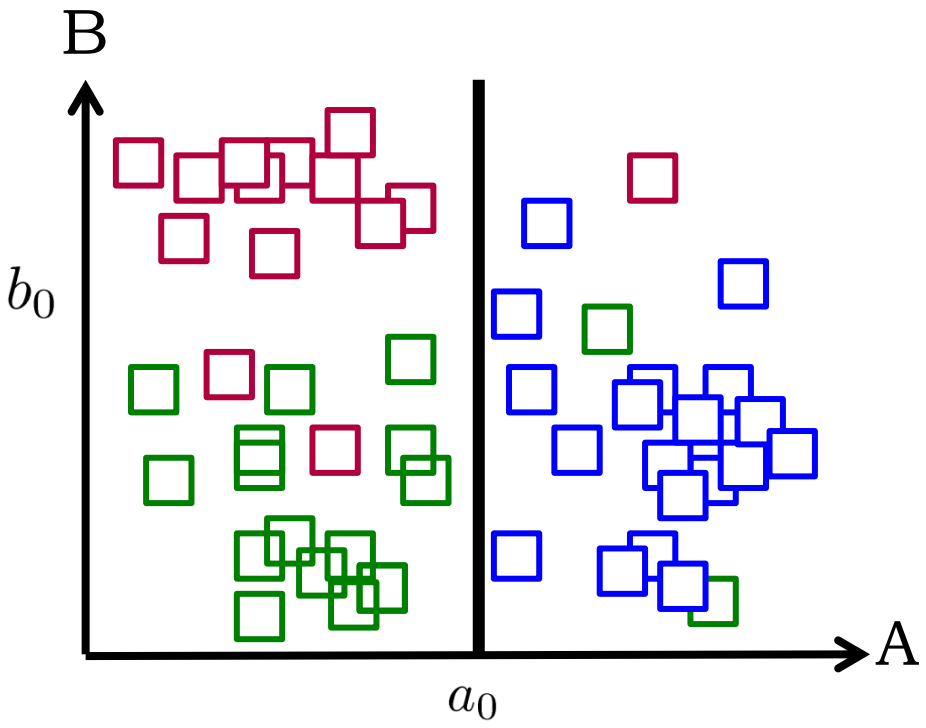




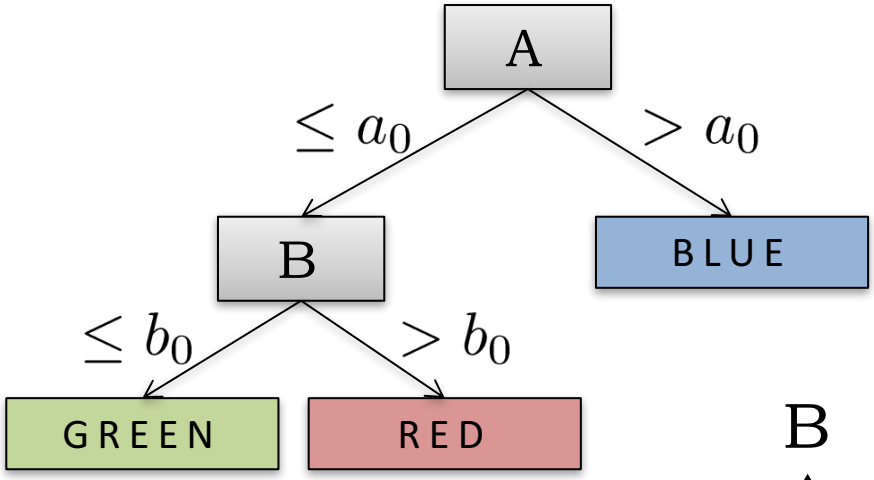
INTELLIGENT SYSTEMS

Decision trees define a classification function through **recursive partitioning** of the instance space.

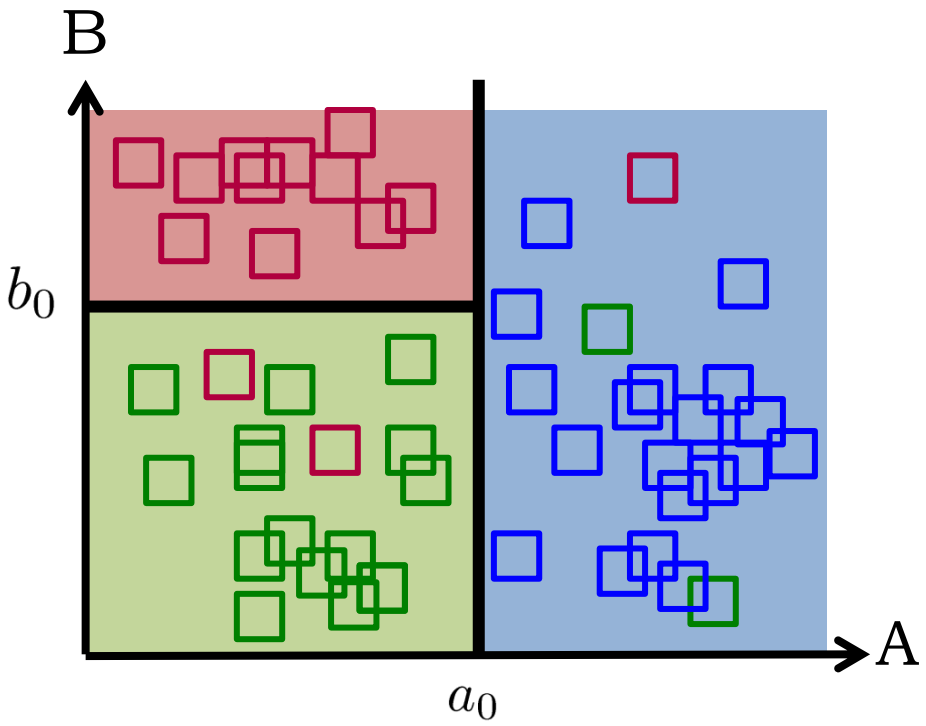




DECISION TREES



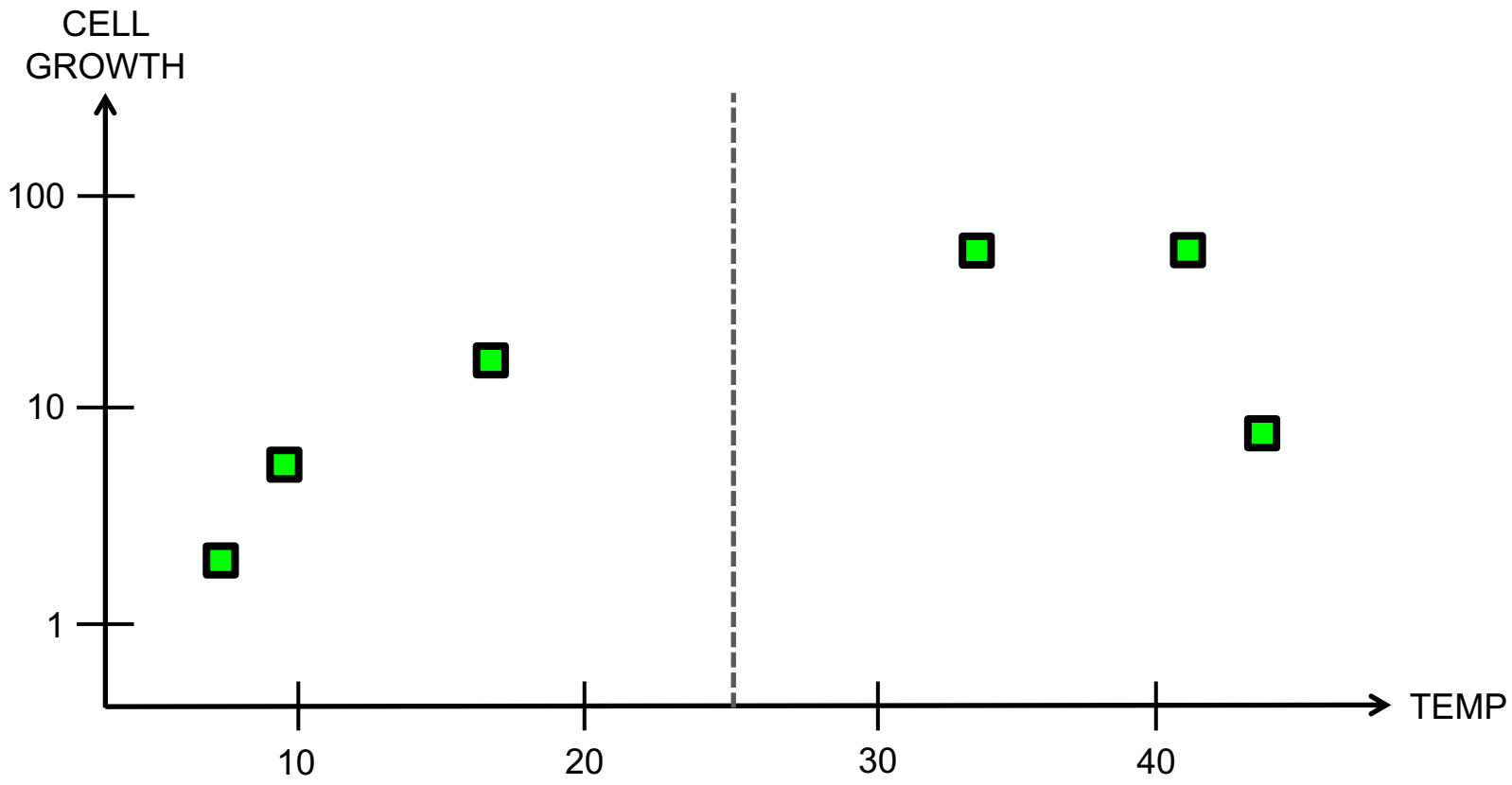
Decision trees define a classification function through **recursive partitioning** of the instance space.



- Performing well on the **training data** does not mean you will **generalize** well to new data!
- In many cases, **model induction** eventually comes down to solving an **optimization problem**.
- While **solving** this problem is an important (algorithmic) part of ML, the true challenge is to **specify the right problem**, i.e., the objective function to the optimized!
- Besides, incorporating the available **domain knowledge** is often key to success!

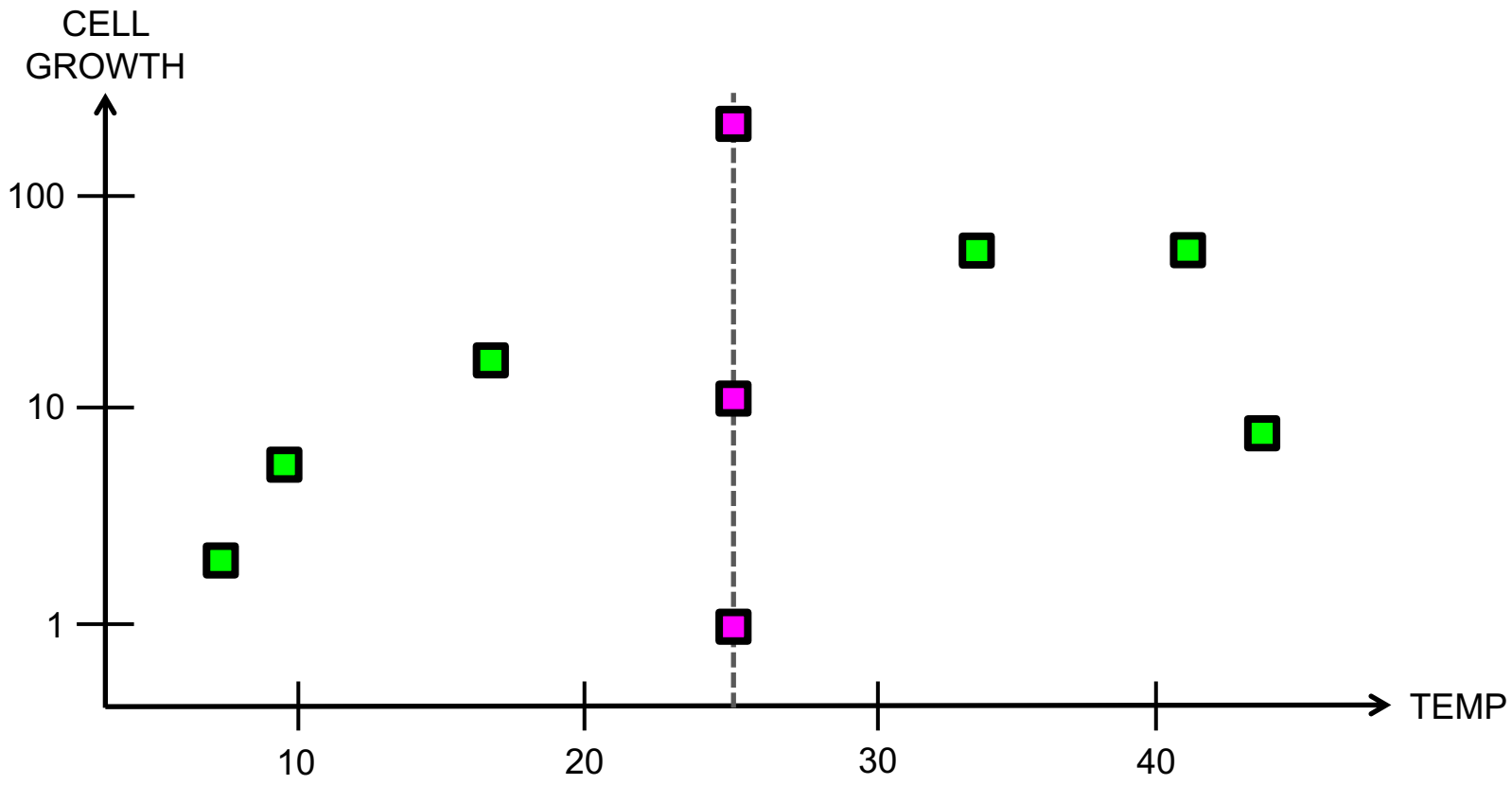
KNOWLEDGE AND DATA

INTELLIGENT SYSTEMS



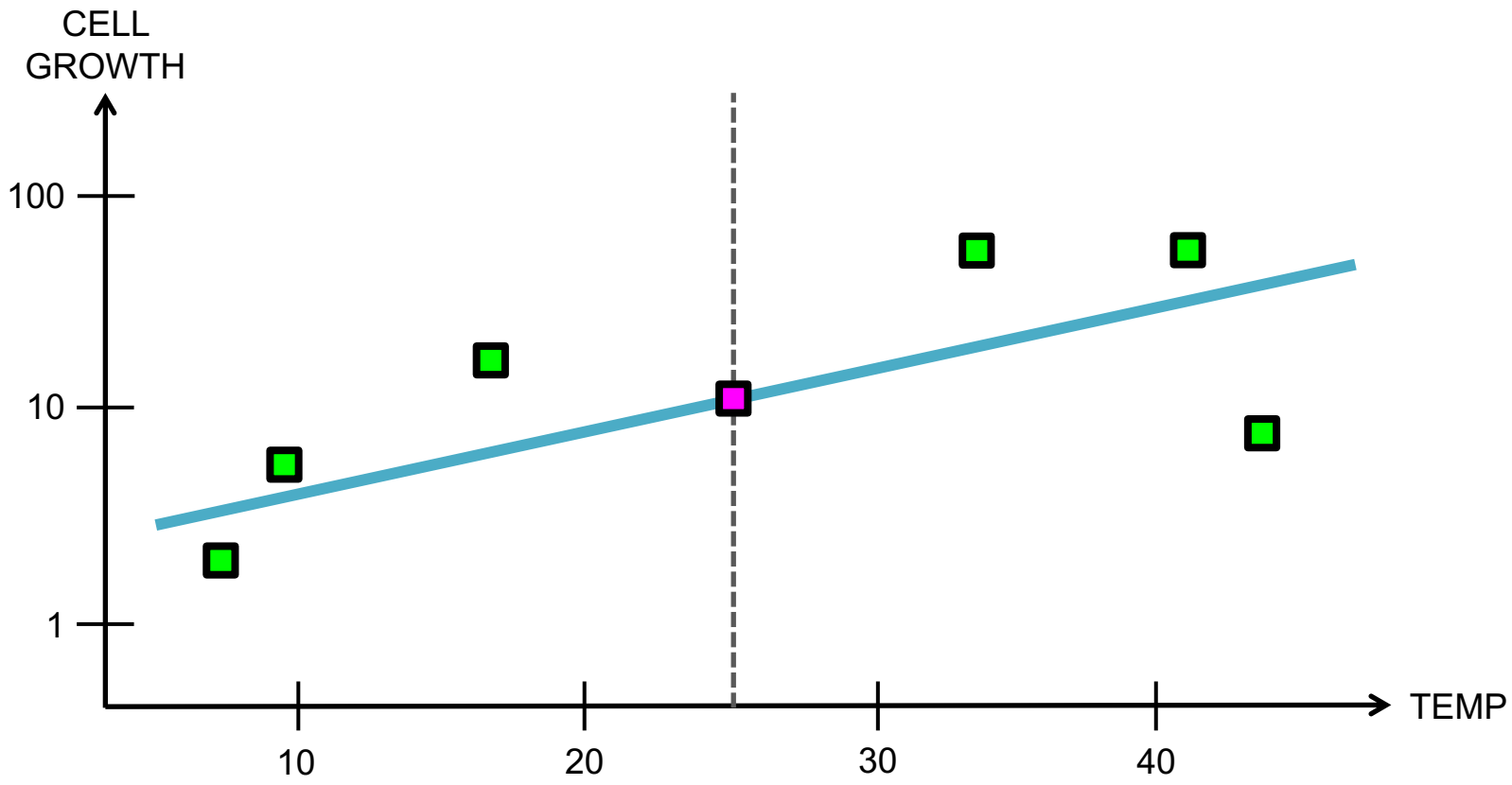
KNOWLEDGE AND DATA

INTELLIGENT SYSTEMS



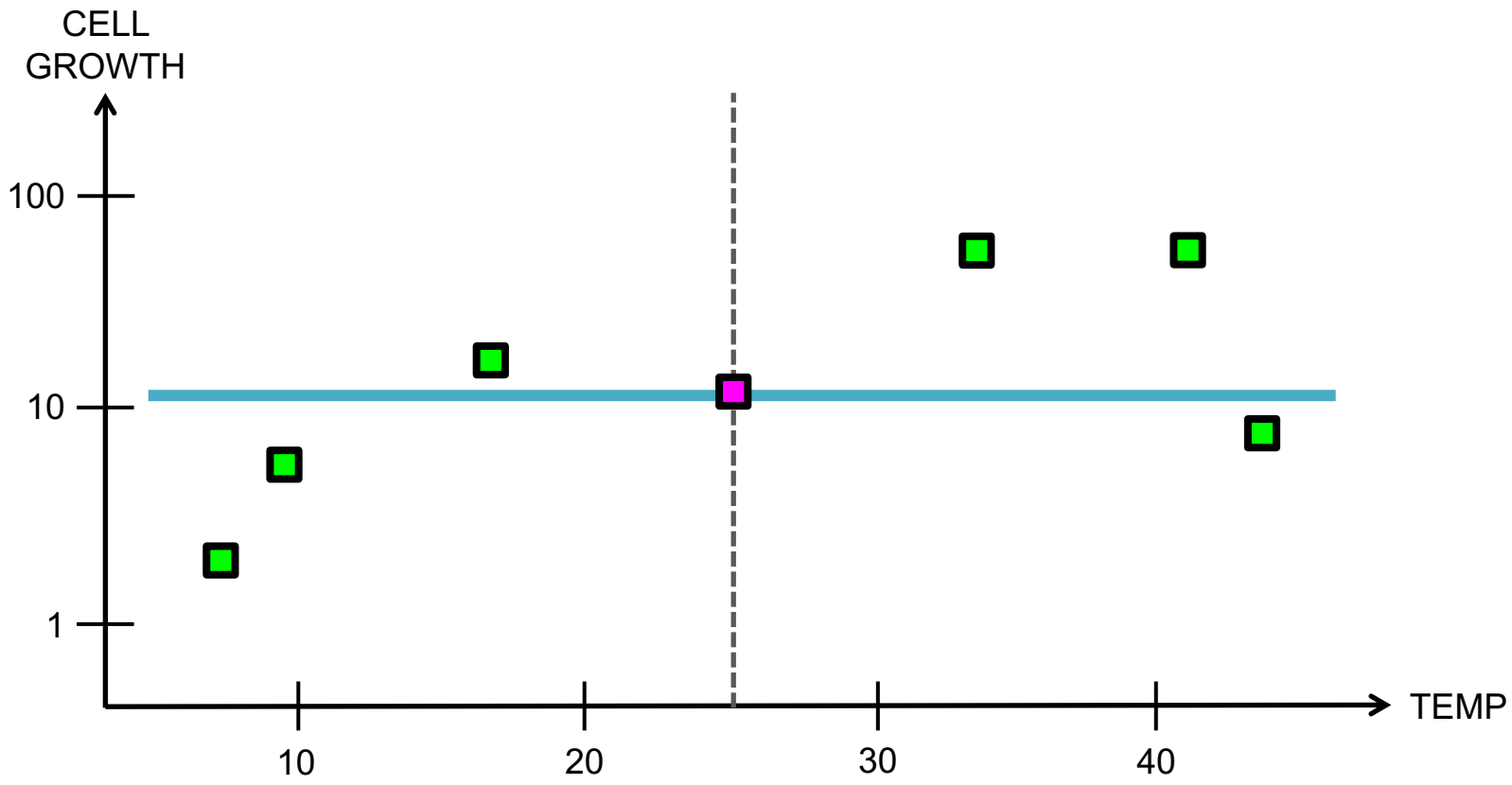
KNOWLEDGE AND DATA

INTELLIGENT SYSTEMS



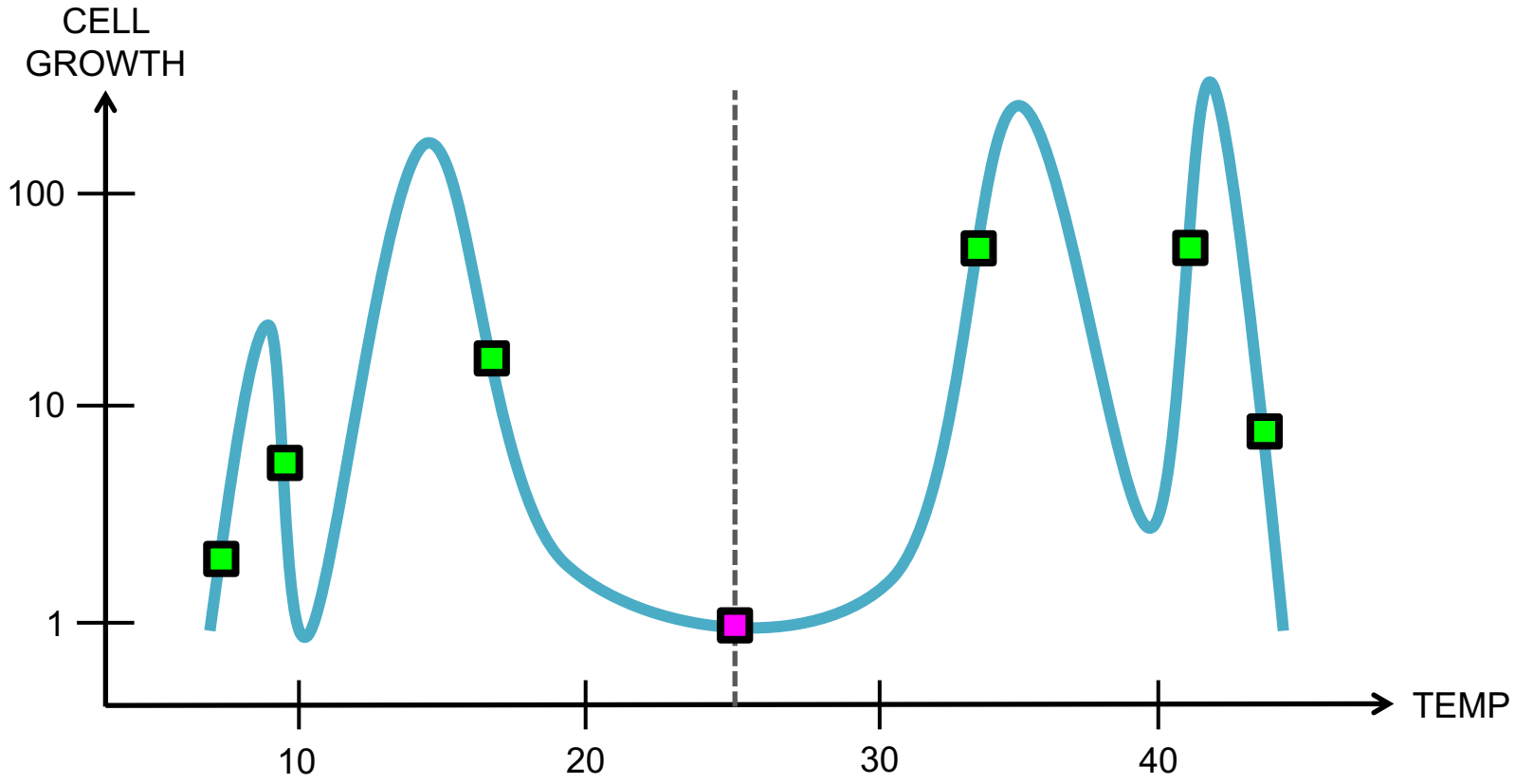
KNOWLEDGE AND DATA

INTELLIGENT SYSTEMS



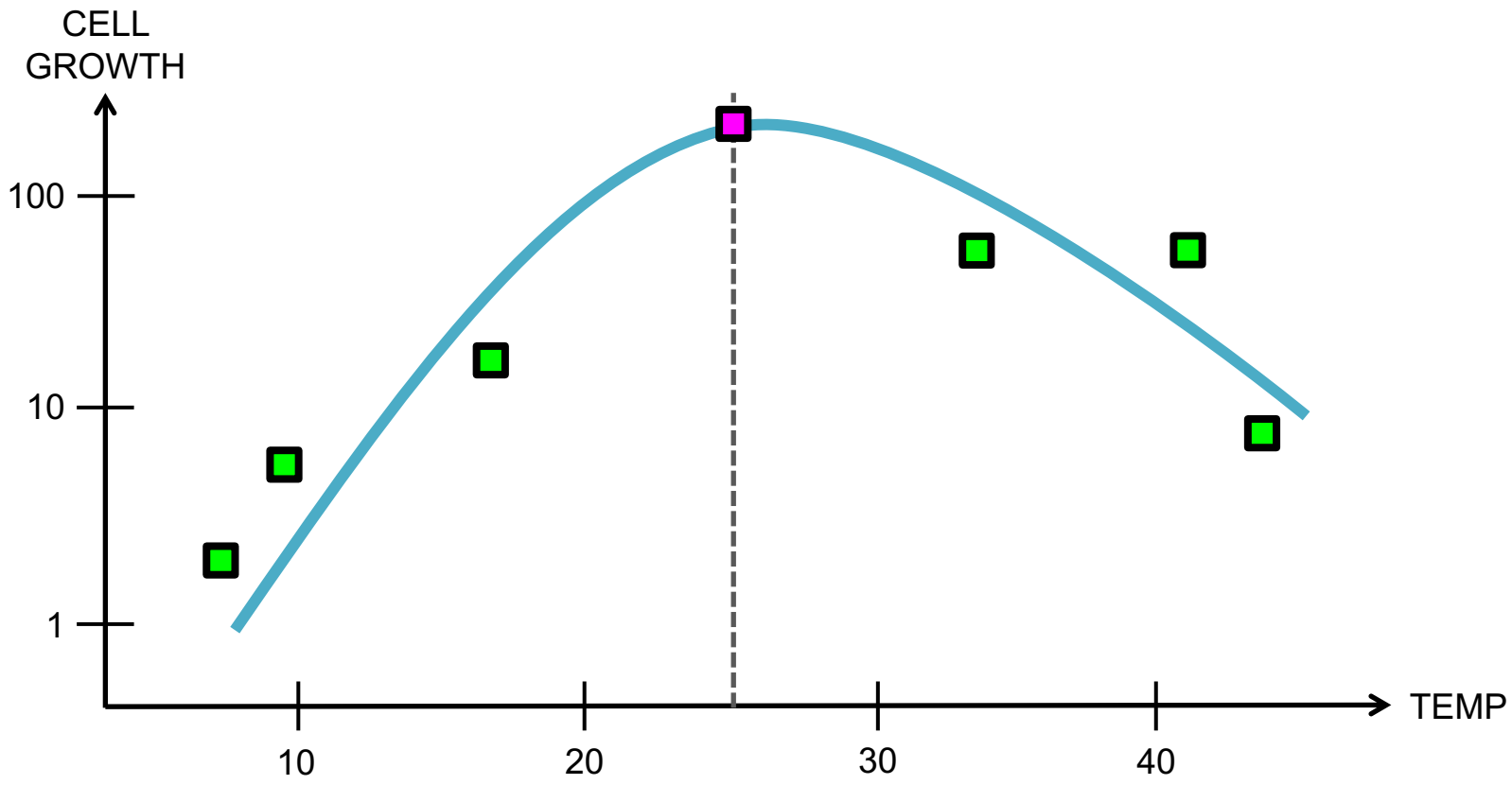
KNOWLEDGE AND DATA

INTELLIGENT SYSTEMS



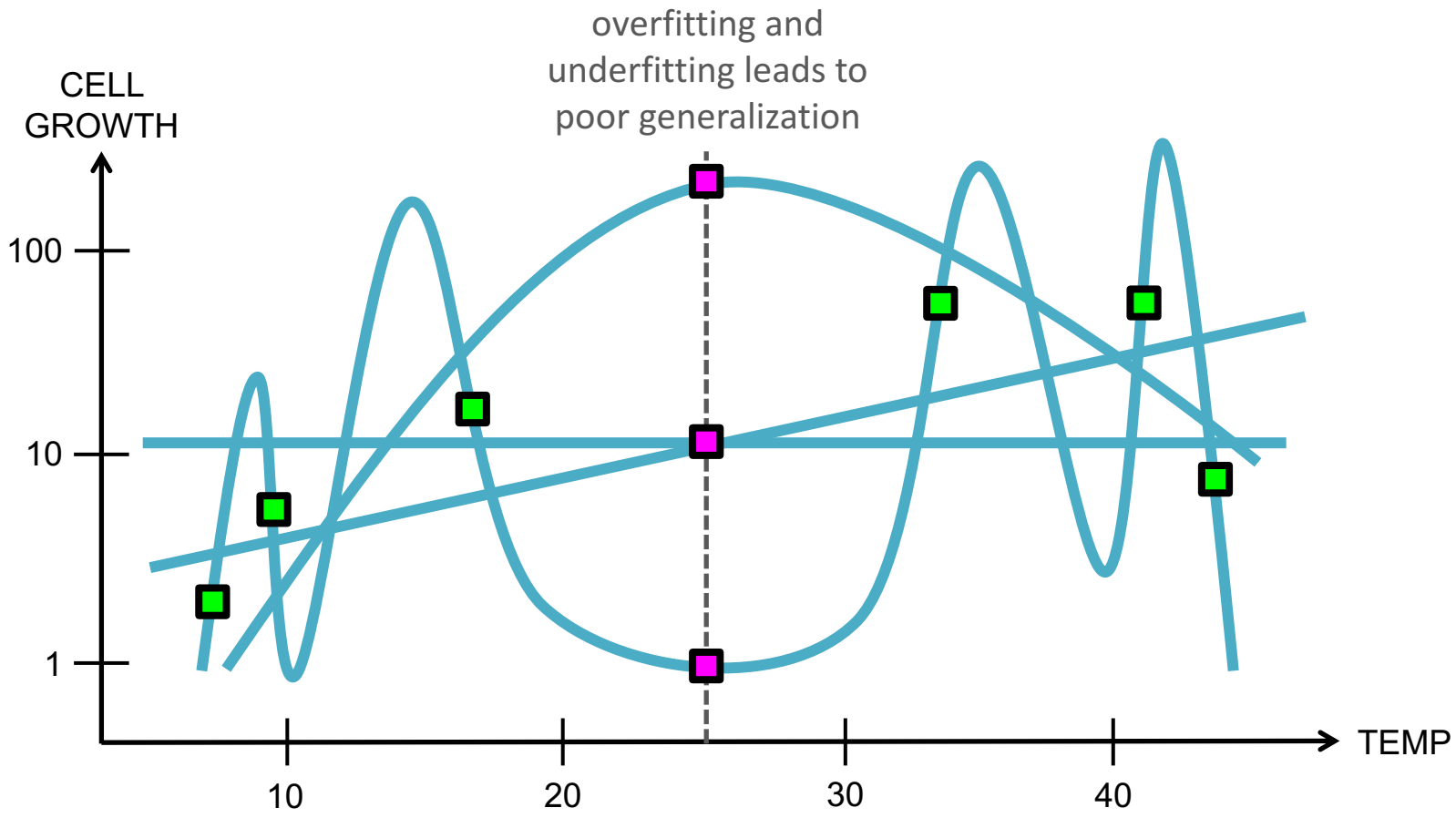
KNOWLEDGE AND DATA

INTELLIGENT SYSTEMS



KNOWLEDGE AND DATA

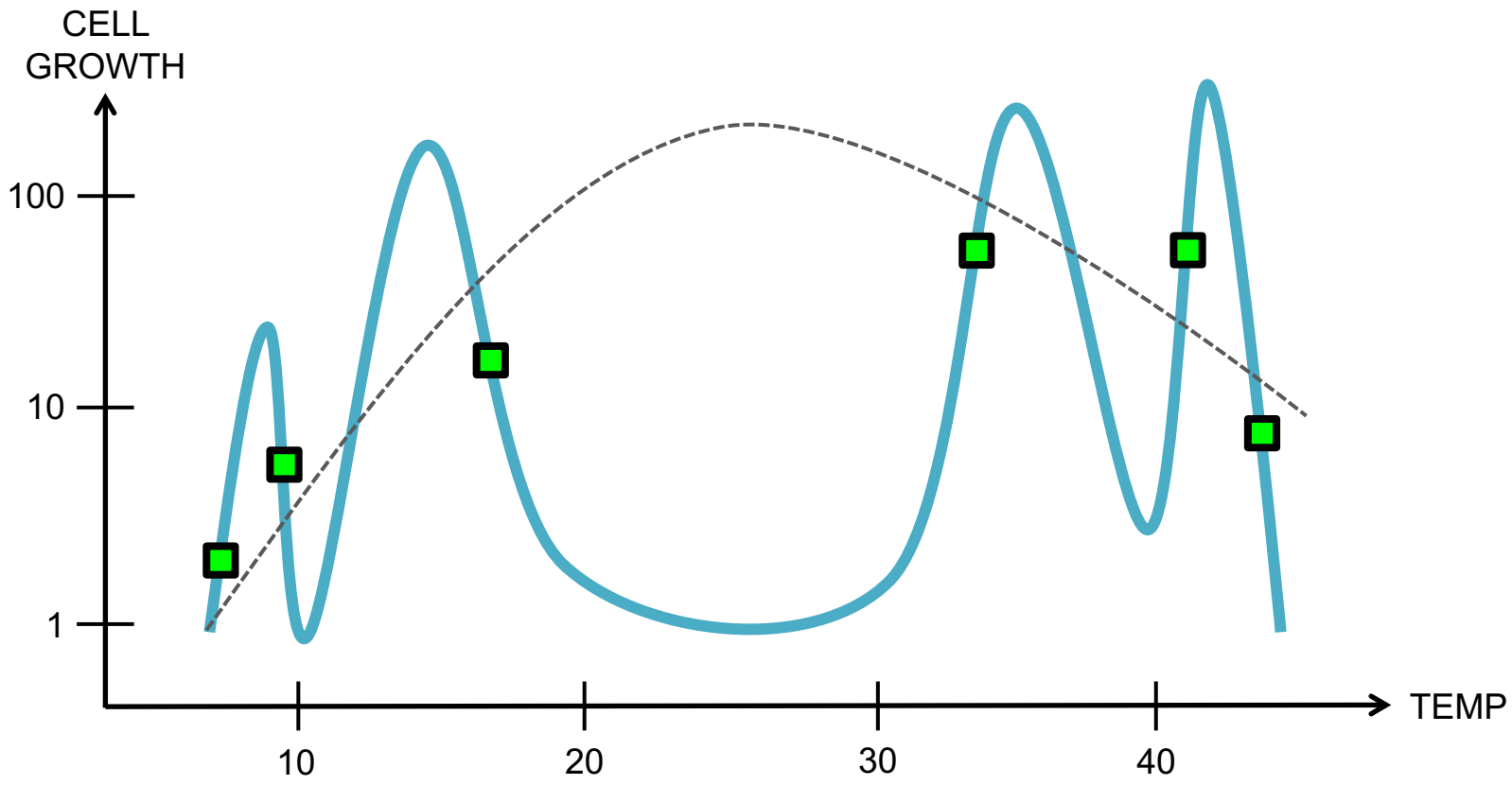
INTELLIGENT SYSTEMS



*Learning = updating **prior knowledge** against the background of observed data*

KNOWLEDGE AND DATA

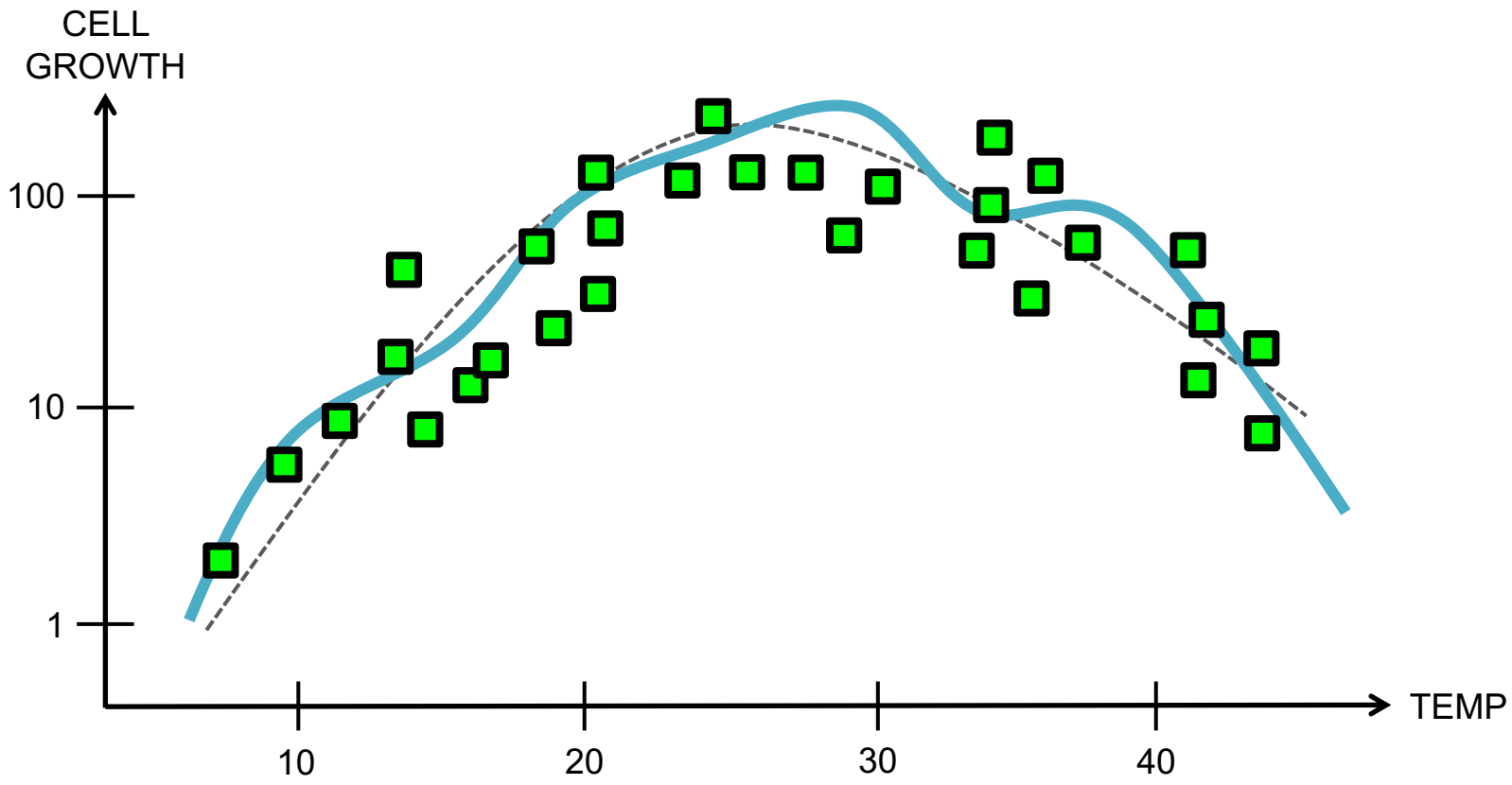
INTELLIGENT SYSTEMS



Data compensates for lack of knowledge …

KNOWLEDGE AND DATA

INTELLIGENT SYSTEMS



Data compensates for lack of knowledge …

- What **hypothesis space** $\mathcal{H}$ to work with? What **suggorate loss** should be minimized, given a specific target loss?
- How to bound **generalization performance** $\mathcal{R}(h)$ based on empirical performance $\mathcal{R}_{emp}(h)$ and sample size N?

$$\mathbf{P}\big(|\mathcal{R}(h) - \mathcal{R}_{emp}(h)| > \epsilon\big) \leq \Omega\big(\mathsf{VC}(\mathcal{H}), N, \epsilon\big)$$

- How does the induced model h compare to the **Bayes-optimal** predictor

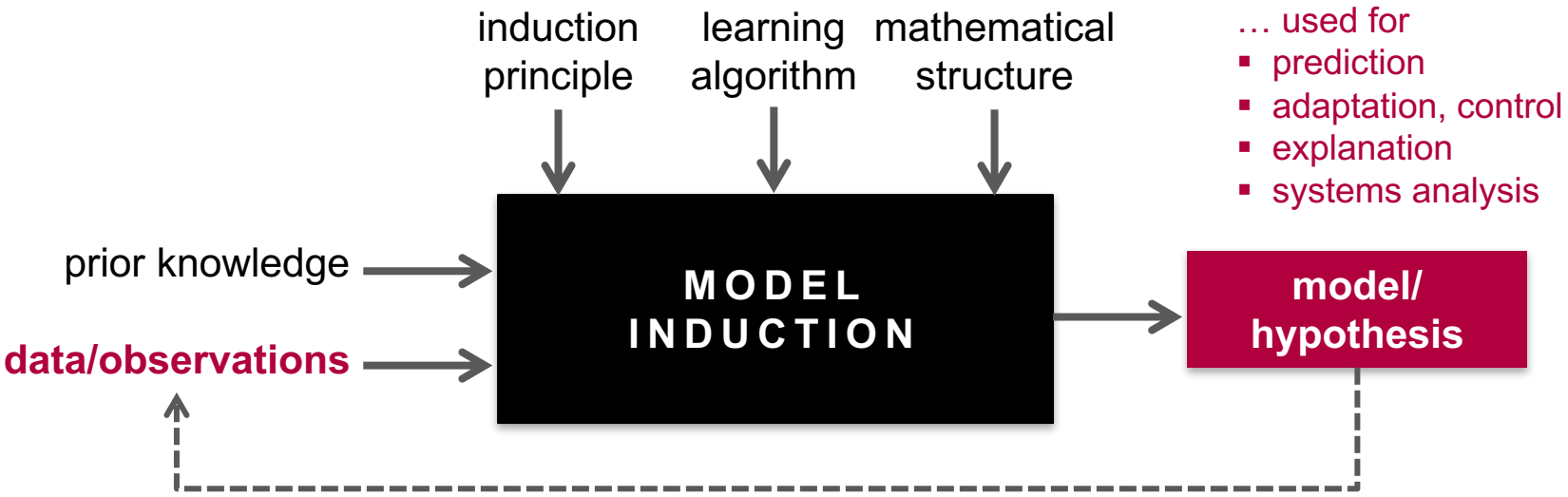
$$h^* : \boldsymbol{x} \mapsto \arg\min_{\hat{y} \in \mathcal{Y}} \int \ell(y, \hat{y})\, \mathbf{P}(y \,|\, \boldsymbol{x})\, d\, y \ ,$$

  which, for each $\boldsymbol{x} \in \mathcal{X}$, predicts the expected loss minimizer?

- ...

LEARNING FROM DATA

INTELLIGENT SYSTEMS



***Problem of induction**: adapting a suitable representation so as to fit observed data and generalize well to new data.*

OUTLINE

INTELLIGENT SYSTEMS

① Prelude

② Machine learning 101

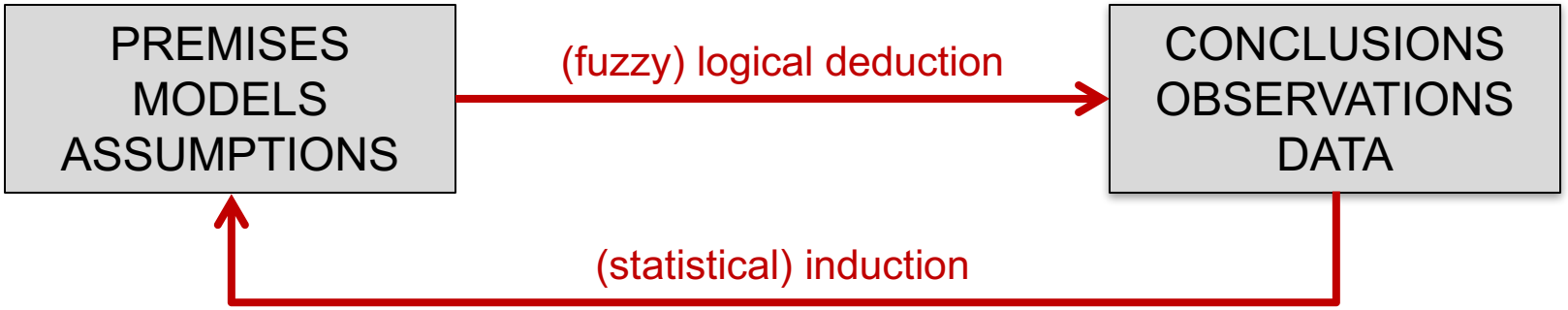
③ Potential contributions of fuzzy logic

④ Fuzzy pattern trees

⑤ Learning from fuzzy data

DEDUCTIVE VS INDUCTIVE REASONING

In its major application fields (approximate reasoning, control, decision making, querying, retrieval and information systems, …), fuzzy logic has primarily been used as a tool for **knowledge representation and information processing.**



KNOWLEDGE-BASED MODELING

INTELLIGENT SYSTEMS

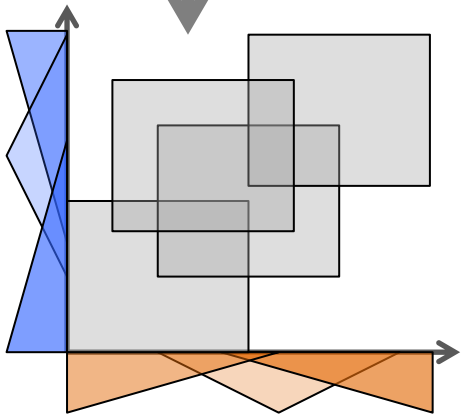
expert

formalization

symbolic representation

```
IF ... THEN ...
IF ... THEN ...
IF ... THEN ...
IF ... THEN ...
IF ... THEN ...
IF ... THEN ...
...
```

precisiation



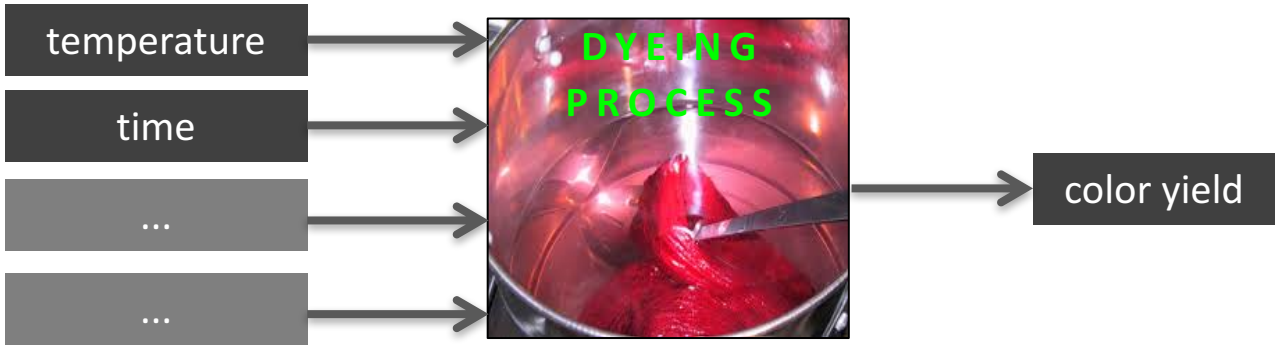
inference

model

$$y = F(\boldsymbol{x})$$

fuzzy logic as a human-machine interface

FUZZY RULES




```
IF temp is low  AND time is low  THEN yield is low
IF temp is low  AND time is med  THEN yield is low
IF temp is low  AND time is high THEN yield is med
IF temp is high AND time is low  THEN yield is med
IF temp is high AND time is med  THEN yield is med
IF temp is high AND time is high THEN yield is high
```

temp = 80, time = 120, yield = ??

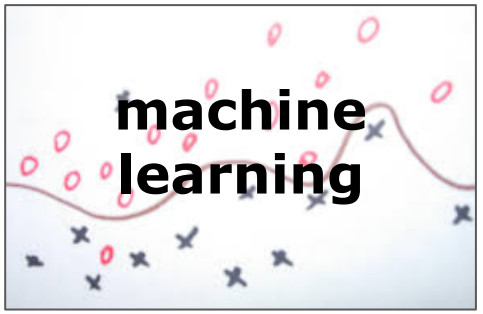
DATA-DRIVEN MODELING

INTELLIGENT SYSTEMS

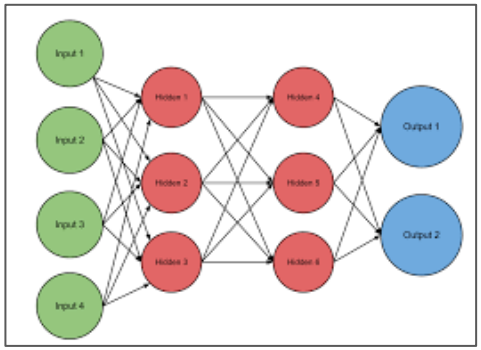
data

preprocessing

numerical representation



induction



inference

model

$$y = F(\boldsymbol{x})$$

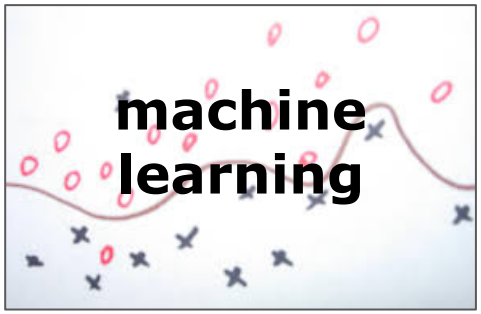
DATA-DRIVEN FUZZY MODELING

INTELLIGENT SYSTEMS

data

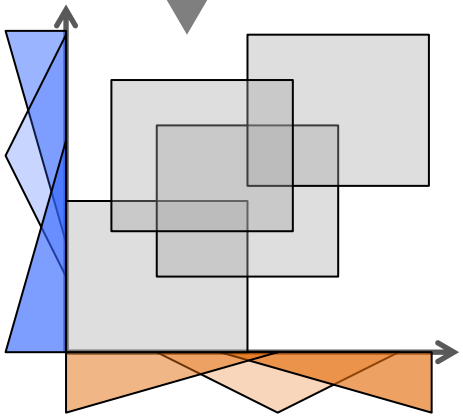
preprocessing

numeric representation



induction

Any benefits of learning a fuzzy model?



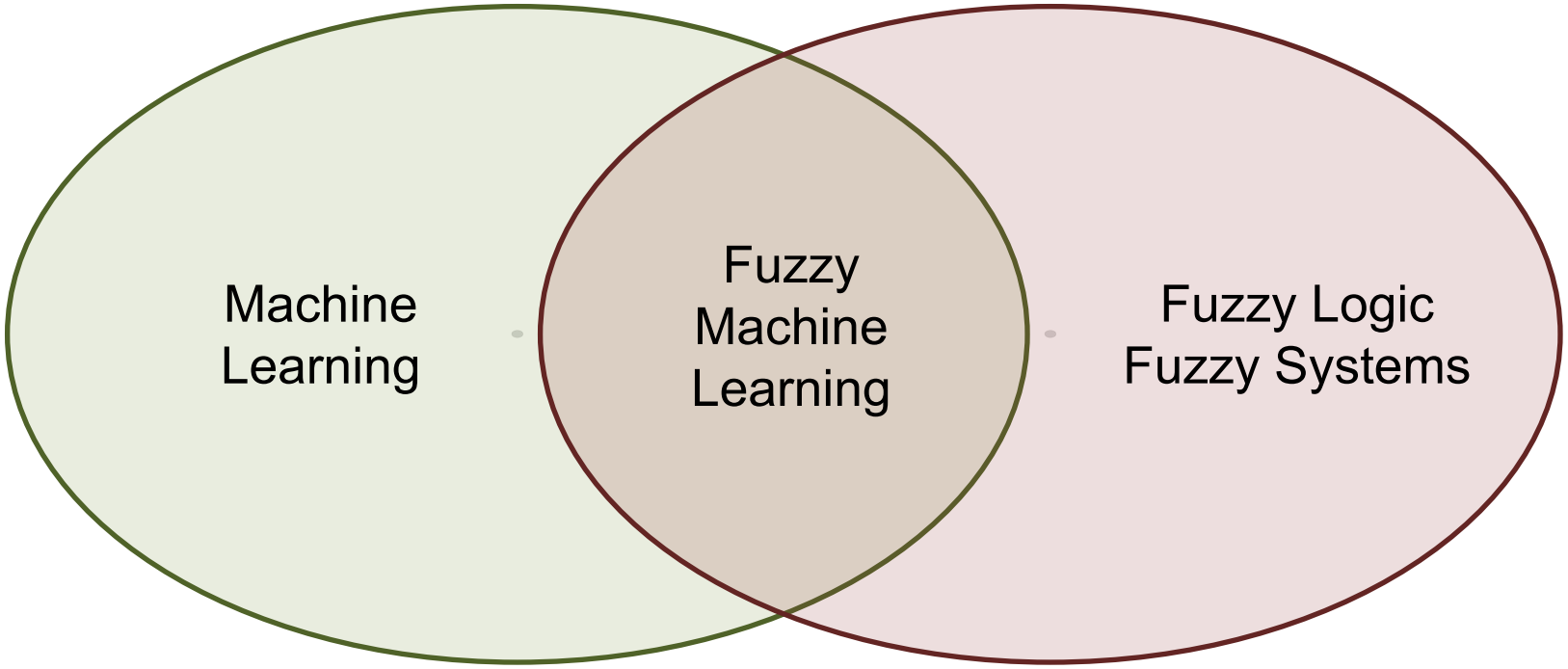
inference

model

$$y = F(\boldsymbol{x})$$

FUZZY MACHINE LEARNING

INTELLIGENT SYSTEMS



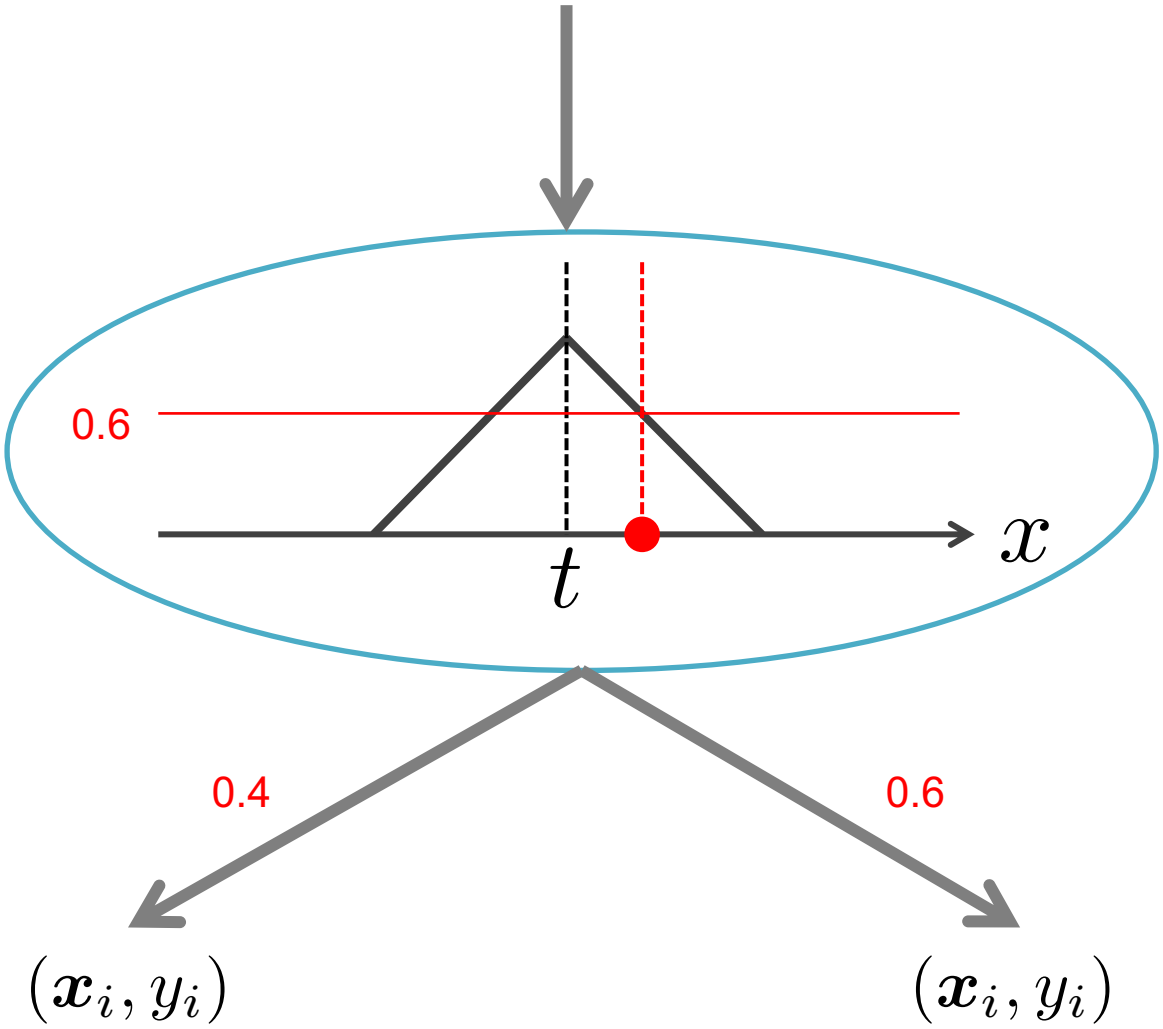
- Extending machine learning through concepts, tools, and techniques from fuzzy logic and fuzzy systems modeling.
- Enhancing fuzzy systems modeling through data-driven approaches.

Fuzzification of models

- Fuzzy extension of standard, non-fuzzy models and methods (blurring the distinction between symbolic and numeric methods, logic-based and arithmetic expressions, discrete and continuous models).
- For example, fuzzy rule induction, fuzzy decisions trees, fuzzy nearest neighbor estimation, fuzzy support vector machines, etc.
- Extension of the representation of corresponding models by means of fuzzy concepts, e.g., fuzzy instead of crisp partitions in decision tree learning.

FUZZY SPLITS IN DECISION TREES

INTELLIGENT SYSTEMS



Fuzzification of models

- Fuzzy extension of standard, non-fuzzy models and methods (blurring the distinction between symbolic and numeric methods, logic-based and arithmetic expressions, discrete and continuous models).
- For example, fuzzy rule induction, fuzzy decisions trees, fuzzy nearest neighbor estimation, fuzzy support vector machines, etc.
- Extension of the representation of corresponding models by means of fuzzy concepts, e.g., fuzzy instead of crisp partitions in decision tree learning.
- Often leads to **increased flexibility** of the model class (e.g., non-axis-parallel boundaries in rule models), which may or may not be an advantage with regard to **generalization performance**.
- Typically accompanied by an increased **computational complexity**.
- Link to fuzzy sets and fuzzy logic often not obvious.

INTELLIGENT SYSTEMS

Interpretability

- Exploiting the usefulness of fuzzy logic in constructing interpretable models (→ linguistic modeling).
- Often taken for granted, without providing a real “proof” of interpretability.
- Are logical structures (necessarily) less interpretable than analytical expressions and ”formulas”?
- Transparency is compromised by the size of (accurate) models (e.g., number and length of rules), complex interaction and inference.
- Transparency/accuracy compromise.
- Are fuzzy sets constructed in a data-driven way semantically meaningful?
- Knowledge-based versus data-driven construction of fuzzy models (human changes role from “producer” to “consumer”).

Uncertainty

- Learning from data is inseparably connected with uncertainty.
- One may argue that probability alone is not enough to capture all relevant sorts of uncertainty.
- Generalized uncertainty formalisms (typically based on non-additive measures), such as possibility theory, may therefore be useful.
- Distinction between aleatoric and epistemic uncertainty.
- Representing uncertain data (ontic or epistemic).

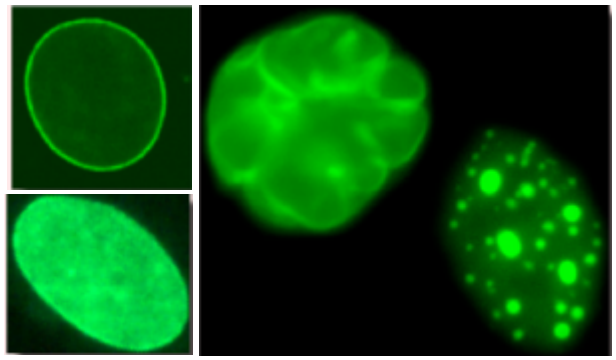
FURTHER POTENTIAL

INTELLIGENT SYSTEMS

- Fuzzy modeling beyond expressing functional dependencies.
- For example, formalization of the learning problem (e.g., loss functions), mathematical structure of data spaces (e.g., state space abstraction in reinforcement learning), feature modeling, etc.
- Well-studied (fuzzy) concepts such as aggregation functions and fuzzy relations allow for specifying key notions, such as fuzzy order relations and generalized transitivity.
- Many ways for incorporating prior knowledge in the learning process (e.g., specifying structure of a rule-based system, learning parameters).
- Non-inductive inference, such as (similarity-based or analogical) knowledge transfer in transfer and multi-task learning.

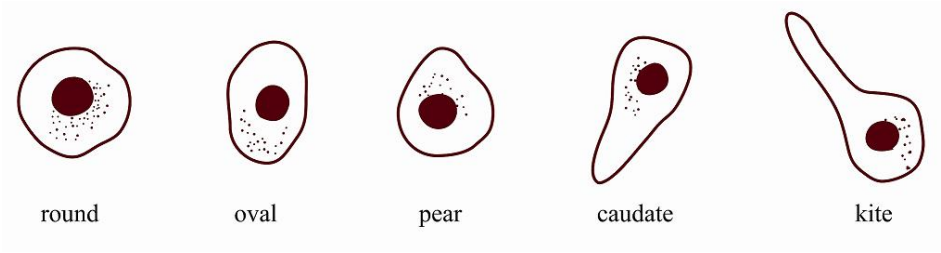
FUZZY FEATURE MODELING

INTELLIGENT SYSTEMS



healthy pathological

shapes of cells



Hudchinson-Gilford syndrome

[Thibault et al. *Cells nuclei classification using **shape** and **texture** indexes*. WSCG 2008.]

FUZZY FEATURE MODELING

INTELLIGENT SYSTEMS

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_m x_m$$

The expert only needs to know THAT but not HOW the features influence the output!

meaningful high-level feature (e.g., roundness)

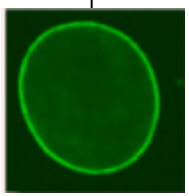
linguistic modeling fuzzy inference

FUZZY FEATURE ENGINEERING

Reduces dimensionality and increases interpretability!

low-level feature (e.g., pixel)

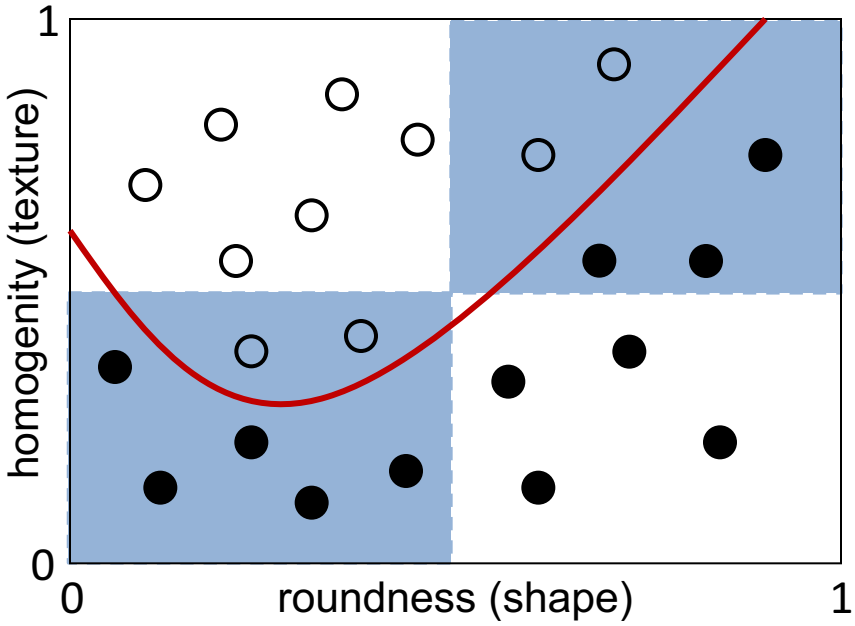
z_1 z_2 z_3 z_4 ... z_d



FUZZY FEATURE MODELING

INTELLIGENT SYSTEMS

Use of gradual (instead of binary) features can increase **discriminative power**!



	homo-geneous	not homo-geneous
round	● ○	●
not round	○	● ○

separable

non-separable

THERE IS A NEED FOR MODELING !

$$y = h(x_1, x_2, \ldots, x_m)$$

data and output modeling

specification of the model space (with the right capacity)

feature modeling and selection

OUTLINE

INTELLIGENT SYSTEMS

① Prelude

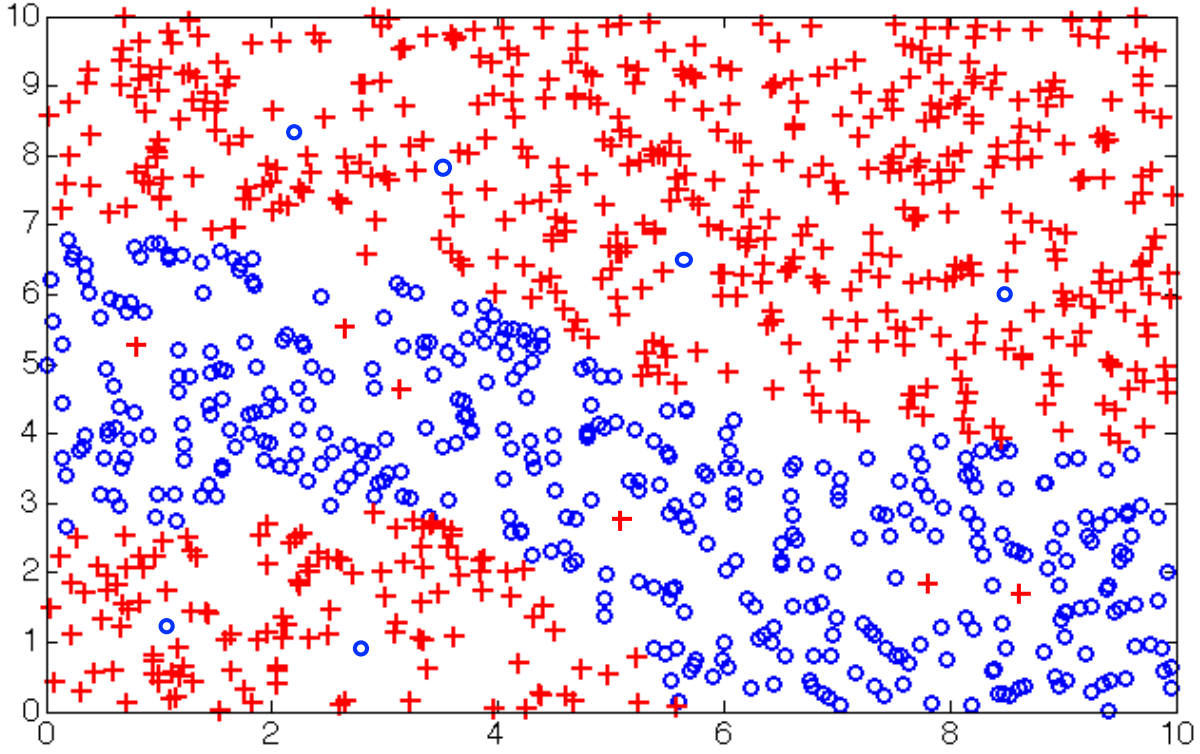
② Machine learning 101

③ Potential contributions of fuzzy logic

④ **Fuzzy pattern trees**

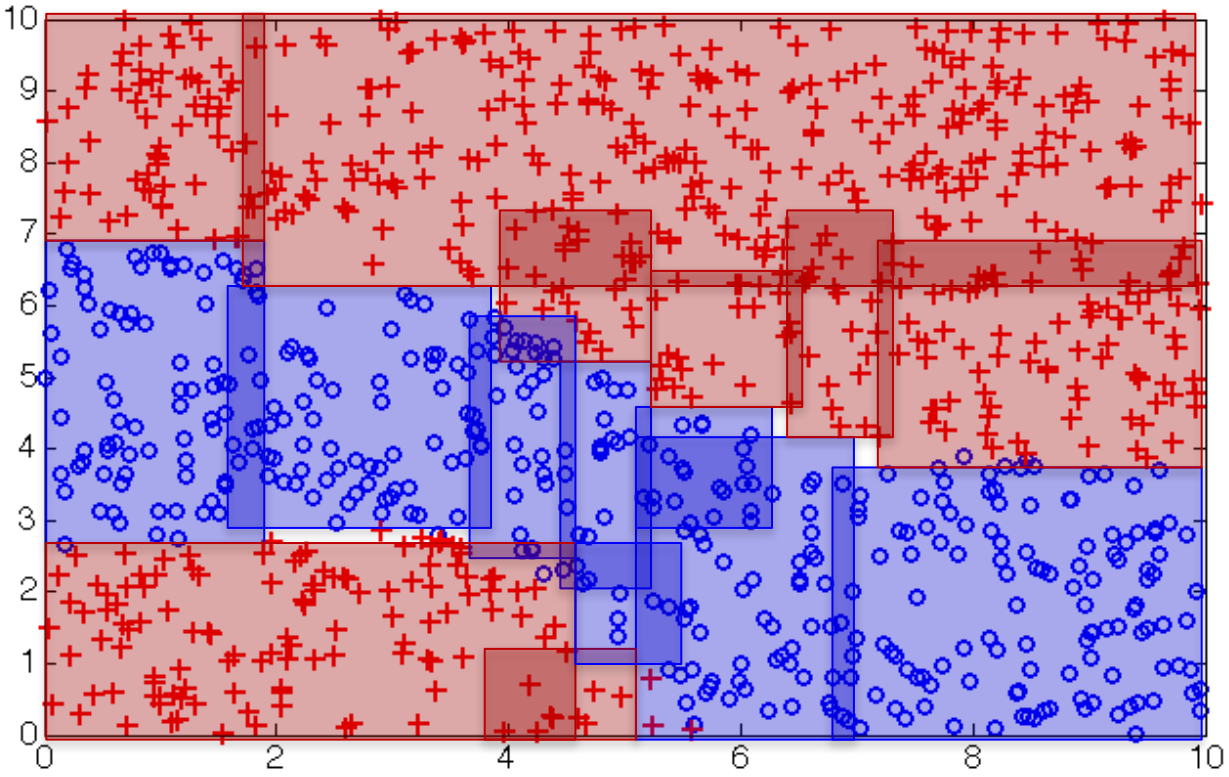
⑤ Learning from fuzzy data

Fuzzy rule models are universal approximators, but ...



INTELLIGENT SYSTEMS

Fuzzy rule models are universal approximators, but ...

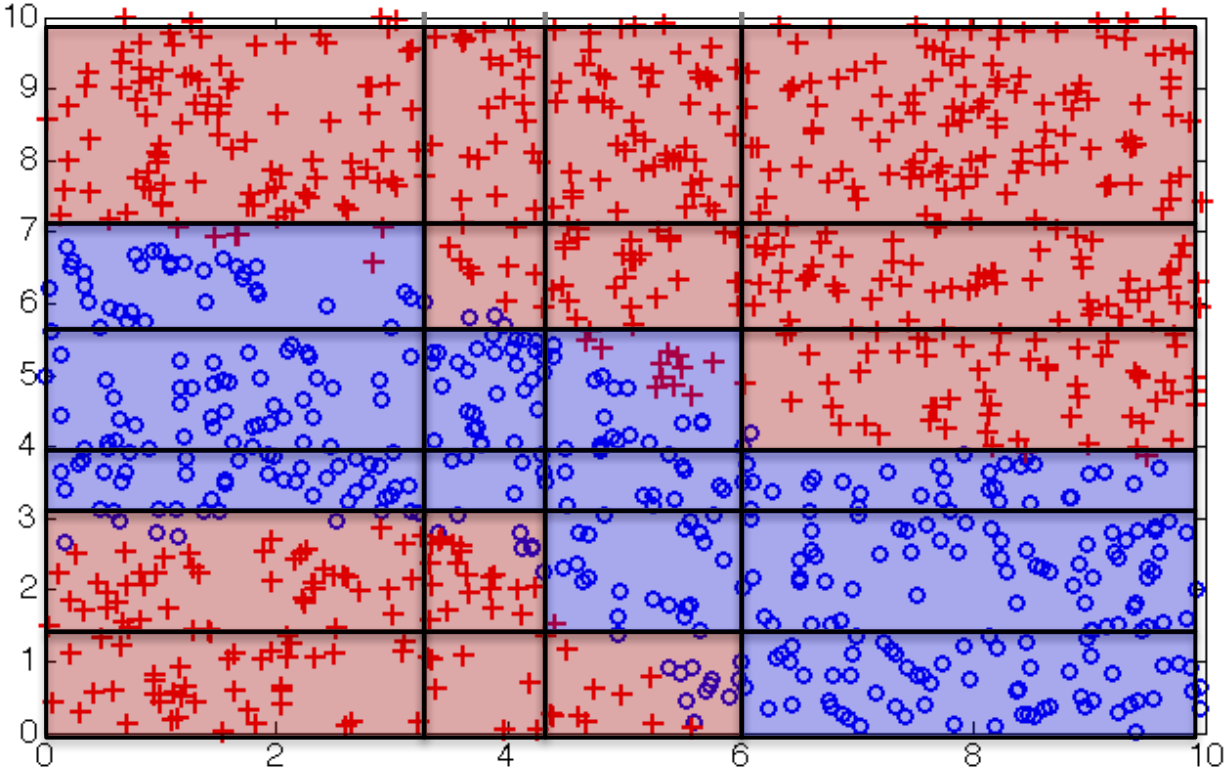


Flexibility of fuzzy models requires many rules!

PROBLEMS OF RULE-BASED METHODS

INTELLIGENT SYSTEMS

Fuzzy rule models are universal approximators, but ...



Flexibility of fuzzy models requires many rules!

INTELLIGENT SYSTEMS

FUZZY RULE SYSTEMS

- flat structure
- local model components
- restricted (logical) combination

FUZZY PATTERN TREES

- hierarchical structure
- global model components
- flexible aggregation functions

FUZZY PATTERN TREES

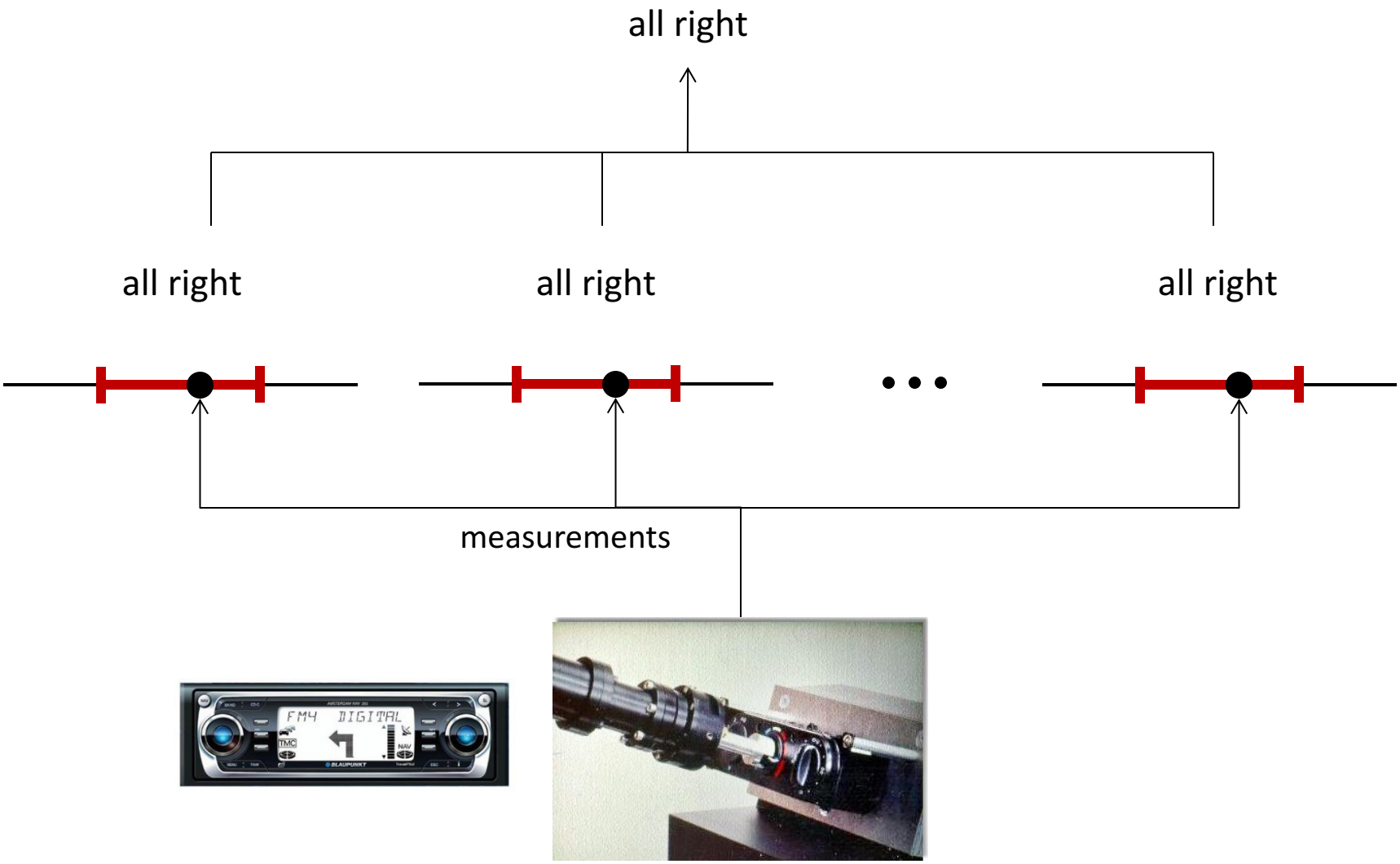
INTELLIGENT SYSTEMS

- FPT is a type of fuzzy model that was independently introduced in
  - Z. Huang, TD. Gedeon, and M. Nikravesh. Pattern trees induction: A new machine learning approach. IEEE TFS 16(4), 2008.
  - Y. Yi, T. Fober and E.H. Fuzzy Operator Trees for Modeling Rating Functions. Int. J. Comp. Intell. and Appl. 8(1), 2009.

- It has recently been further developed in
  - R. Senge and E.H. Pattern Trees for Regression and Fuzzy Systems Modeling. Proc. WCCI-2010, Barcelona, Spain, 2010.
  - R. Senge and E.H. Top-Down Induction of Fuzzy Pattern Trees. IEEE TFS, 19(2), 2011.
  - R. Senge and E.H. Fast Fuzzy Pattern Tree Learning for Classification. IEEE TFS, 23(6), 2015.

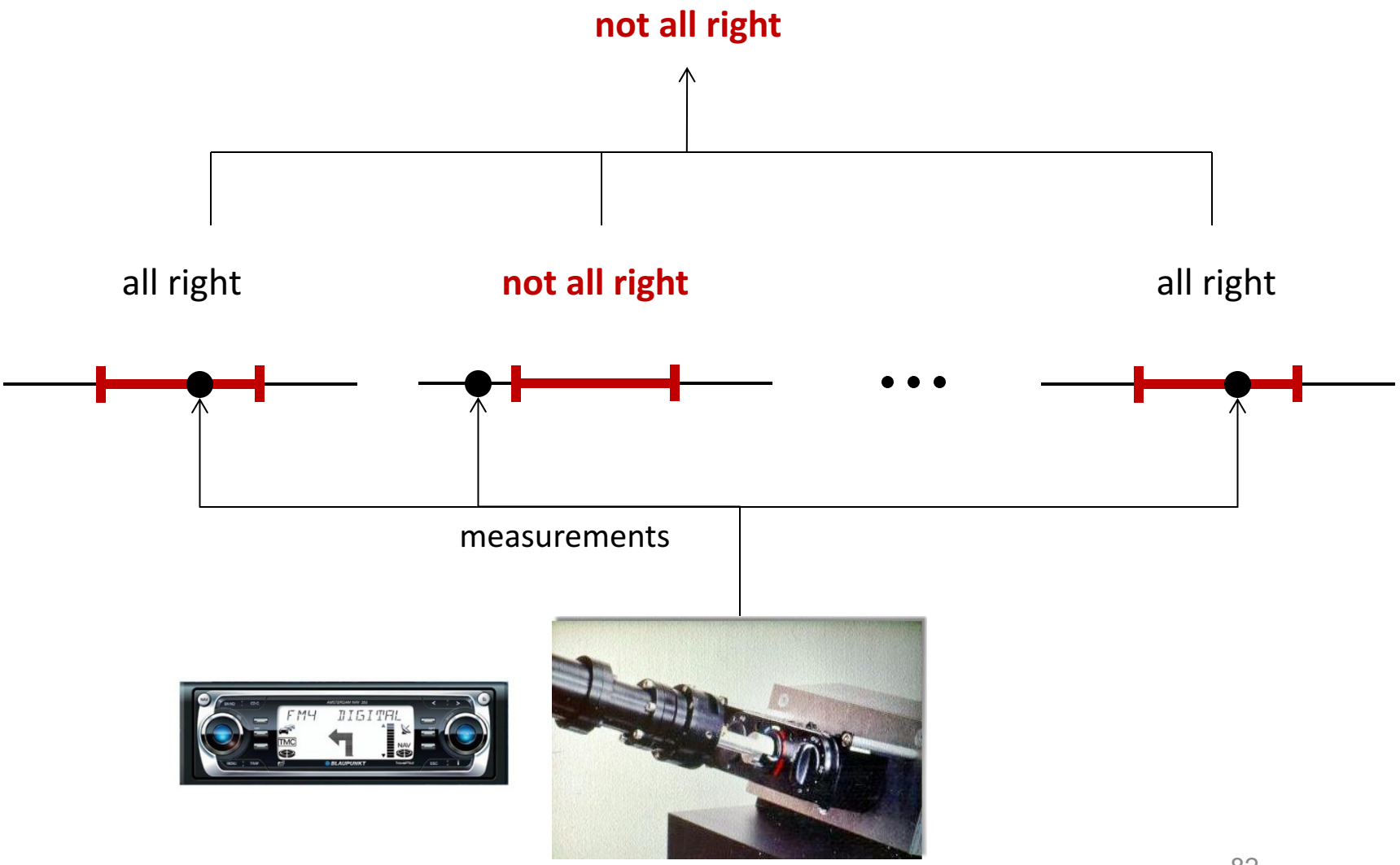
SIMPLE QUALITY CONTROL

INTELLIGENT SYSTEMS

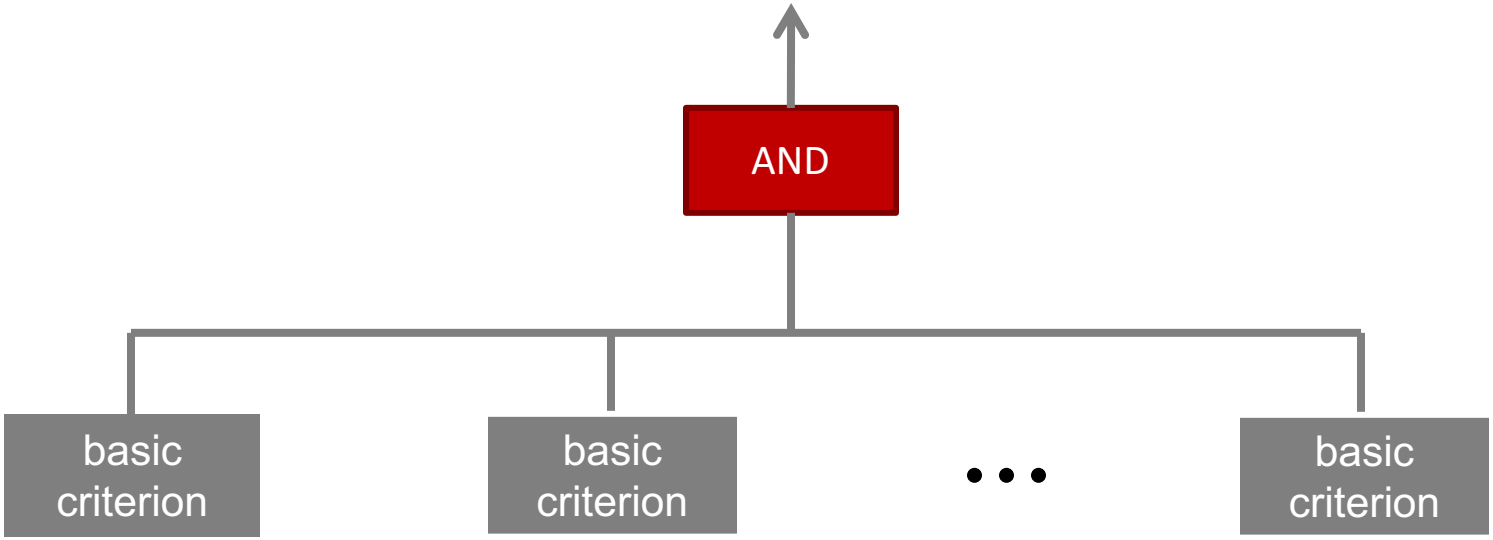


SIMPLE QUALITY CONTROL

INTELLIGENT SYSTEMS



SIMPLE QUALITY CONTROL



$$Q(x_1, x_2, \ldots, x_n) = \bigwedge_{i=1}^{n} P_i(x_i)$$

DRAWBACKS OF THIS APPROACH

INTELLIGENT SYSTEMS

- Bivalent, **non-gradual evaluation** is not natural and does not support a proper ranking of products.
- **No compensation**: Several good properties cannot compensate for a single bad one.
- Extremely **sensitive toward noise**.
- „Flat“ structure of the evaluation scheme is **not scalable**.

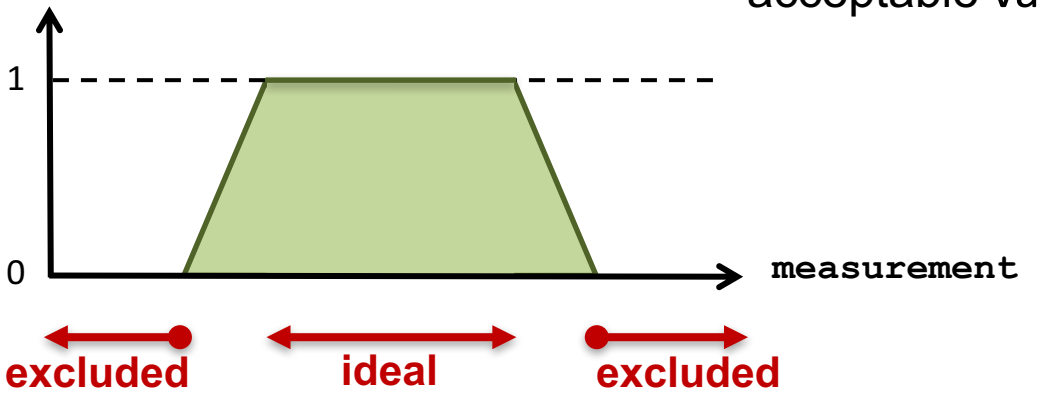
single device

interior of a car

EVALUATION OF INDIVIDUAL CRITERIA

INTELLIGENT SYSTEMS

„fuzzy range“ of acceptable values



GENERALIZED AGGREGATION

Fuzzy pattern trees support three different **modes of aggregation** for criteria:

- **CONJUNCTIVE** („and“): both criteria must be fulfilled
- **DISJUNCTIVE** („or“): either of the criteria must be fulfilled
- **AVERAGING**

AGGREGATION: CONJUNCTION

INTELLIGENT SYSTEMS

T-norms $\top : [0,1]^2 \to [0,1]$ as generalized conjunctions:

- $\top(x,0) = 0,\ \top(1,x) = x$
- $\top(x,y) = \top(y,x)$
- $\top(x,y) \geq \top(x,z)$ for $y > z$
- $\top(x,\top(y,z)) = \top(\top(x,y),z)$

Examples:

- $\top_M(x,y) = \min(x,y)$
- $\top_P(x,y) = x \times y$
- $\top_L(x,y) = \max(x+y-1,0)$
- $\top_\alpha(x,y) = \frac{x \cdot y}{\max\{x,y,\alpha\}}$

Order relation: $\top_L \leq \top_P \leq \top_M$

AGGREGATION: DISJUNCTION

INTELLIGENT SYSTEMS

T-conorms $\top : [0,1]^2 \to [0,1]$ as generalized conjunctions:

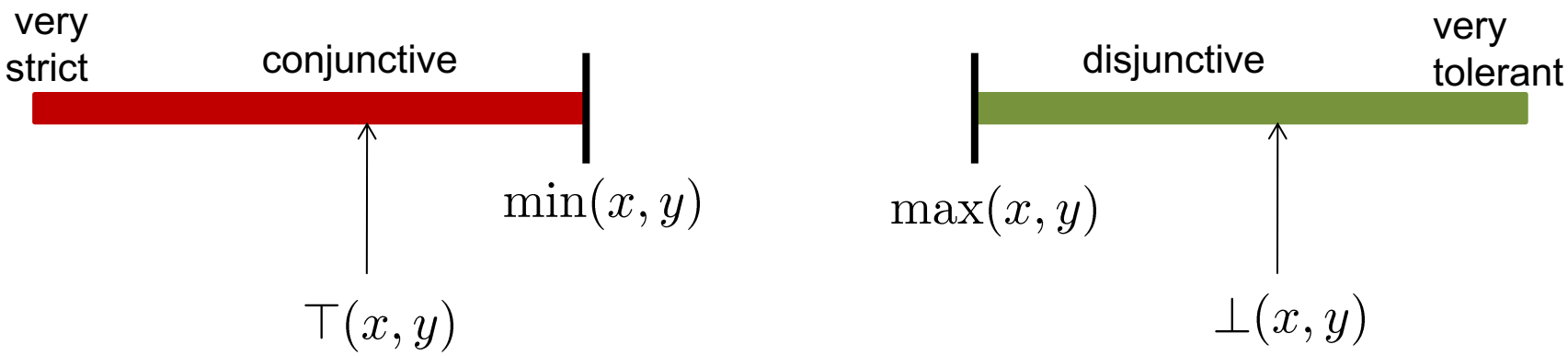
- $\perp(x,0) = \alpha$, $\perp(1,x) = 1$
- $\perp(x,y) = \perp(y,x)$
- $\perp(x,y) \geq \perp(x,z)$ for $y > z$
- $\perp(x,\top(y,z)) = \perp(\perp(x,y),z)$

Examples:

- $\perp_M(x,y) = \max(x,y)$
- $\perp_P(x,y) = x + y - x \times y$
- $\perp_L(x,y) = \min(x+y,1)$
- $\perp_\alpha(x,y) = \frac{x+y-x\cdot y-\min\{x,y,1-\alpha\}}{\max\{1-x,1-y,\alpha\}}$

Order relation: $\perp_M \leq \perp_P \leq \perp_L$

GENERALIZED AGGREGATION



The **discrete Choquet integral** of $f : C \to \mathbb{R}_+$ with respect to μ is defined as follows:

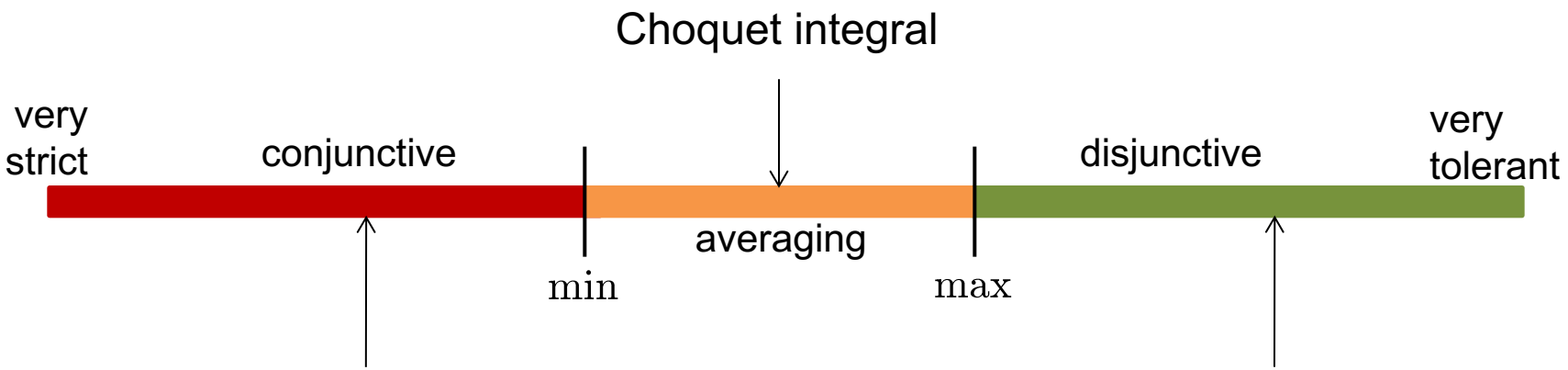
$$\mathcal{C}_\mu(f) = \sum_{i=1}^{m} \left(f(c_{(i)}) - f(c_{(i-1)})\right) \cdot \mu\left(A_{(i)}\right) \ ,$$

where $(\cdot)$ is a permutation of $\{1, \ldots, m\}$ such that $0 \leq f(c_{(1)}) \leq f(c_{(2)}) \leq \ldots \leq f(c_{(m)})$, and $A_{(i)} = \{c_{(i)}, \ldots, c_{(m)}\}$.

The Choquet integral expressed in terms of its Möbius transform:

$$\mathcal{C}_\mu(f) = \sum_{T \subseteq C} \boldsymbol{m}_\mu(T) \times \min_{c_i \in T} f(c_i)$$

GENERALIZED AGGREGATION



$$\top_\alpha(x,y) = \frac{x \cdot y}{\max\{x, y, \alpha\}}$$

$$\perp_\alpha(x,y) = \frac{x + y - x \cdot y - \min\{x, y, 1-\alpha\}}{\max\{1-x, 1-y, \alpha\}}$$

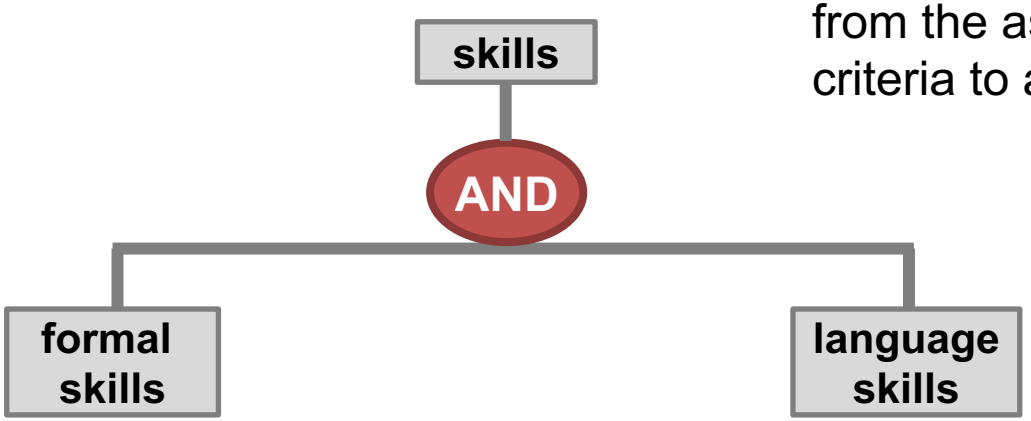
INTELLIGENT SYSTEMS

skills

from the assessment of basic criteria to an overall evaluation

HIERARCHICAL MODELING

INTELLIGENT SYSTEMS

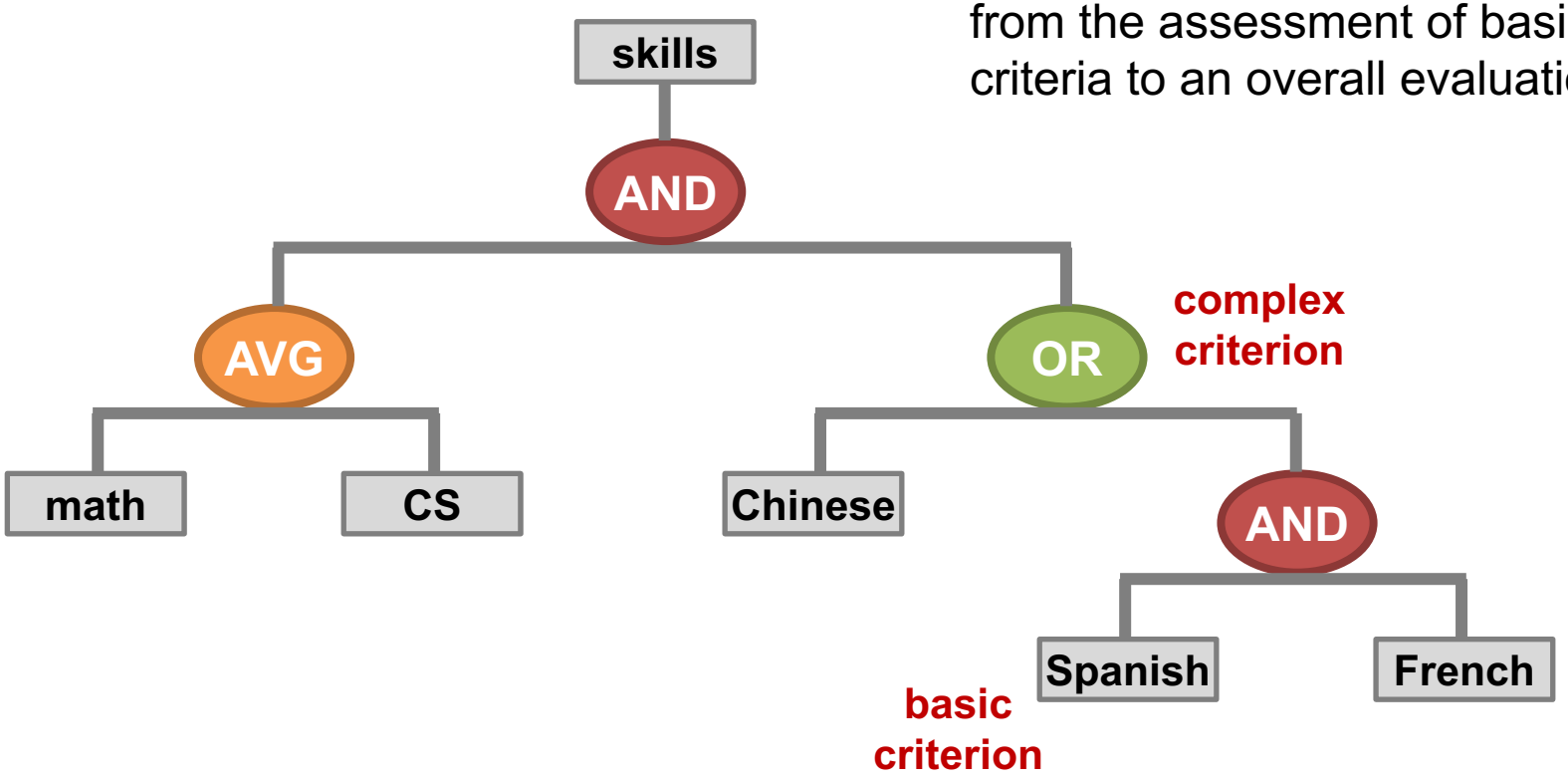


from the assessment of basic criteria to an overall evaluation

HIERARCHICAL MODELING

INTELLIGENT SYSTEMS

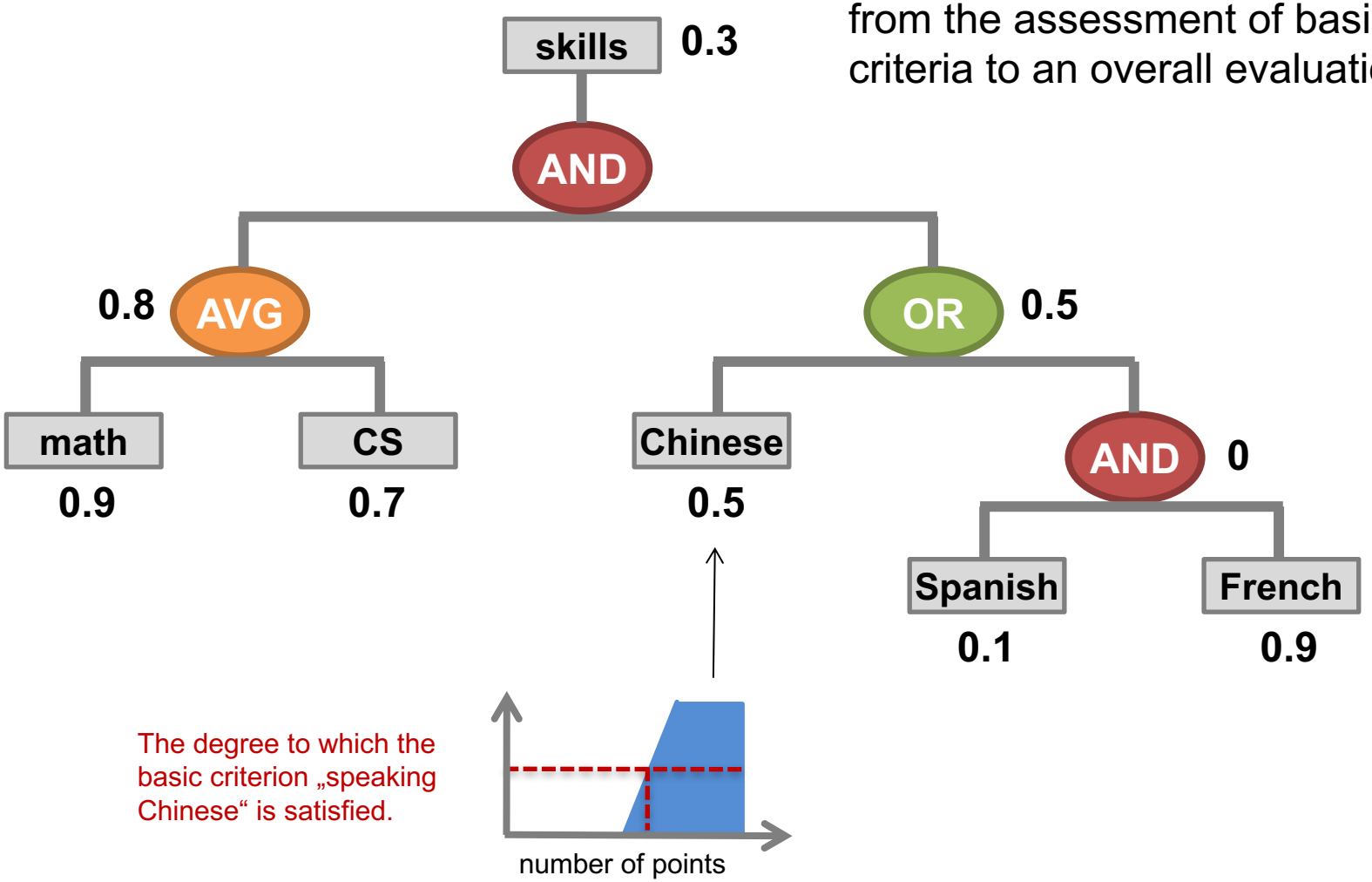
from the assessment of basic criteria to an overall evaluation



HIERARCHICAL MODELING

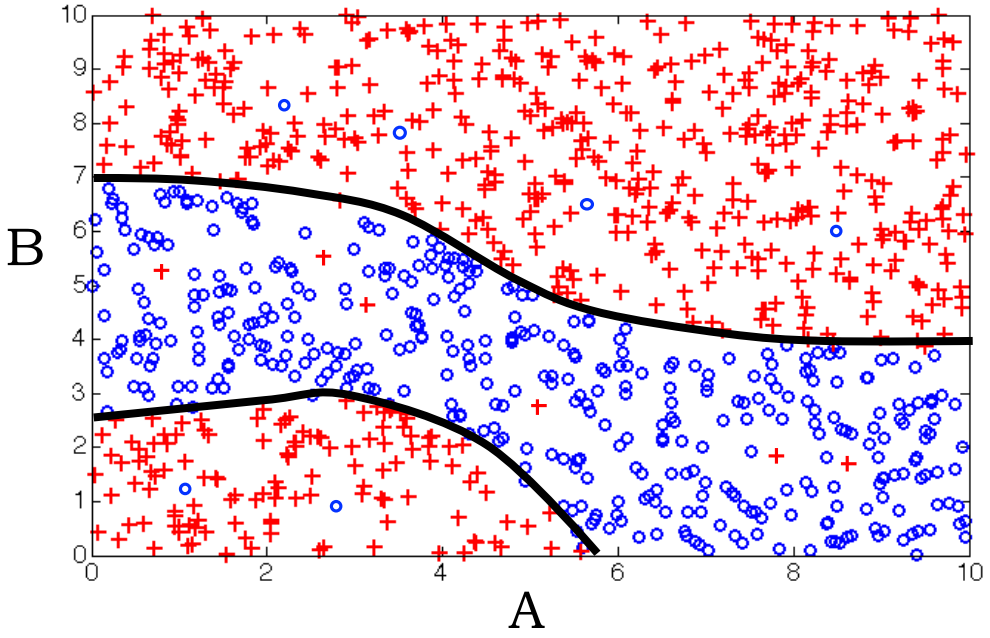
INTELLIGENT SYSTEMS

from the assessment of basic criteria to an overall evaluation

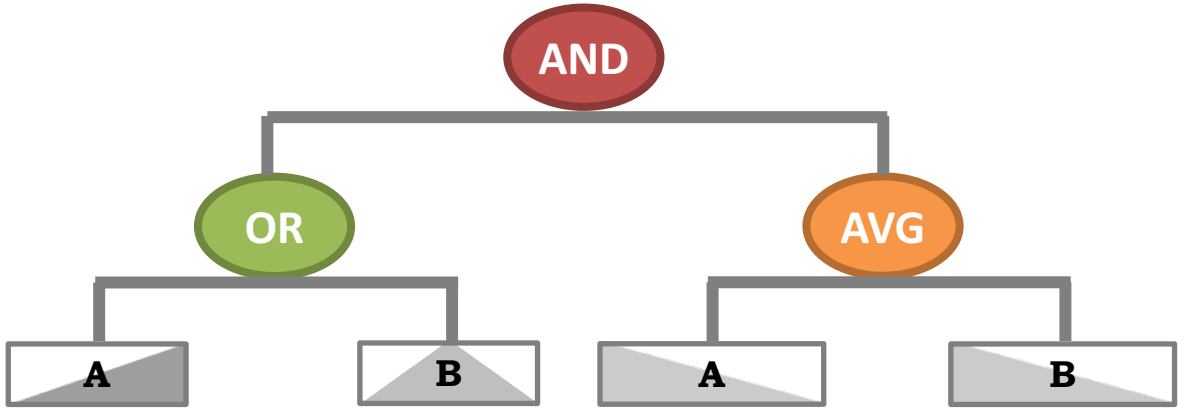


FUZZY PATTERN TREES

INTELLIGENT SYSTEMS

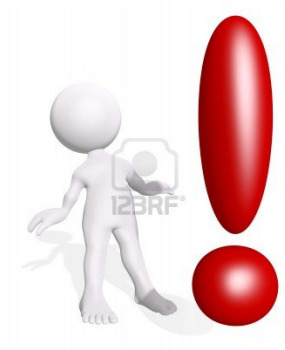


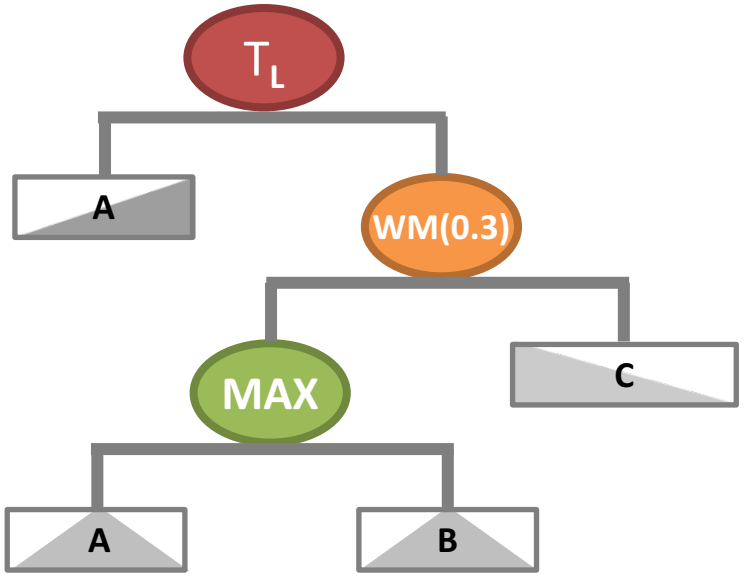
Modeling non-linear dependencies and decision boundaries in a compact, interpretable way!



KNOWLEDGE VS DATA-DRIVEN MODELING

INTELLIGENT SYSTEMS

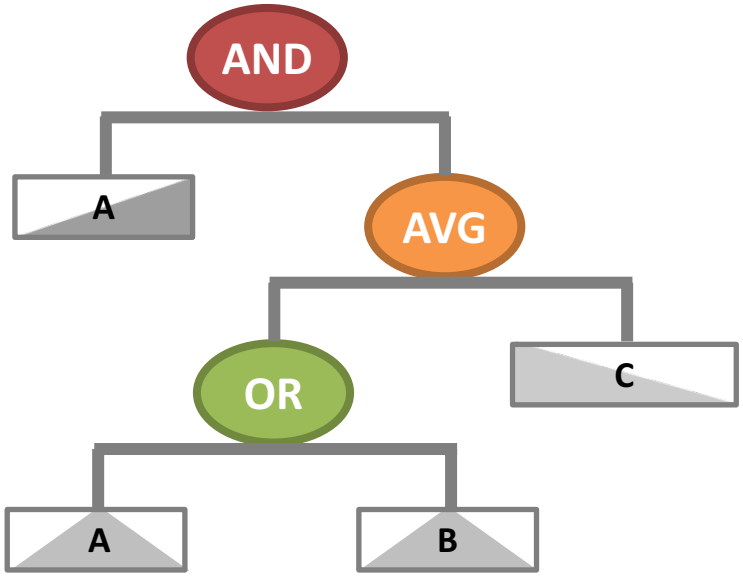




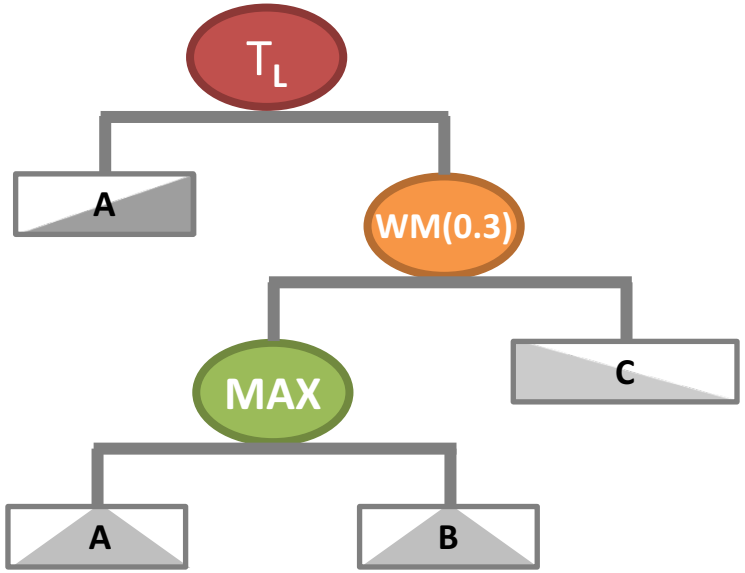
purely knowledge-driven model construction, no learning

KNOWLEDGE VS DATA-DRIVEN MODELING

INTELLIGENT SYSTEMS



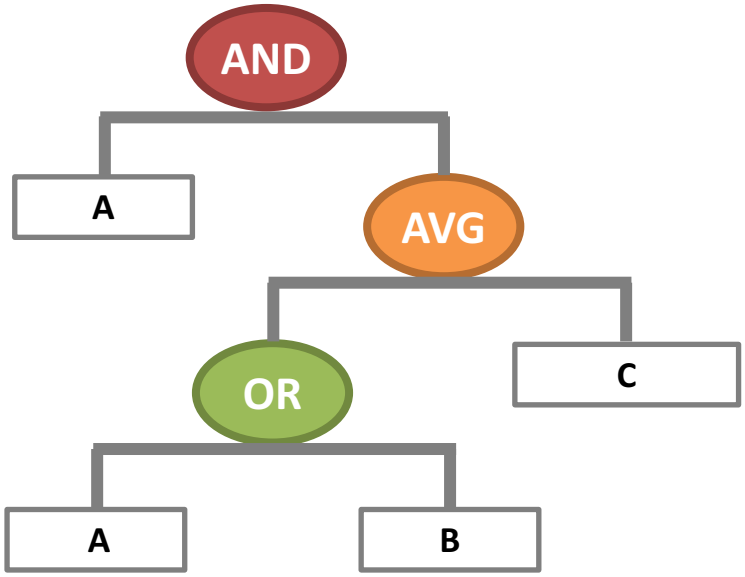
0.6	3	10	1
0.7	2	15	0
0.9	2	16	0
0.3	2	16	1
0.5	6	17	1
...	.	..	.



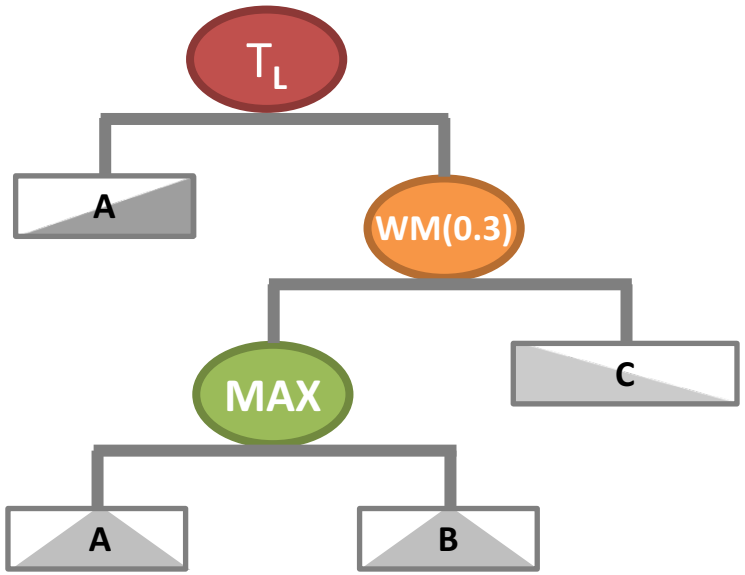
model calibration (parameter estimation)

KNOWLEDGE VS DATA-DRIVEN MODELING

INTELLIGENT SYSTEMS



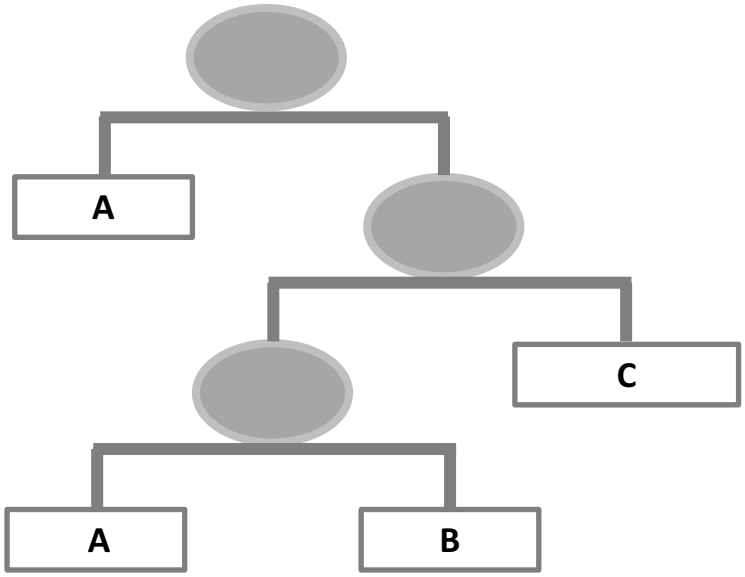
0.6	3	10	1
0.7	2	15	0
0.9	2	16	0
0.3	2	16	1
0.5	6	17	1
...	.	..	.



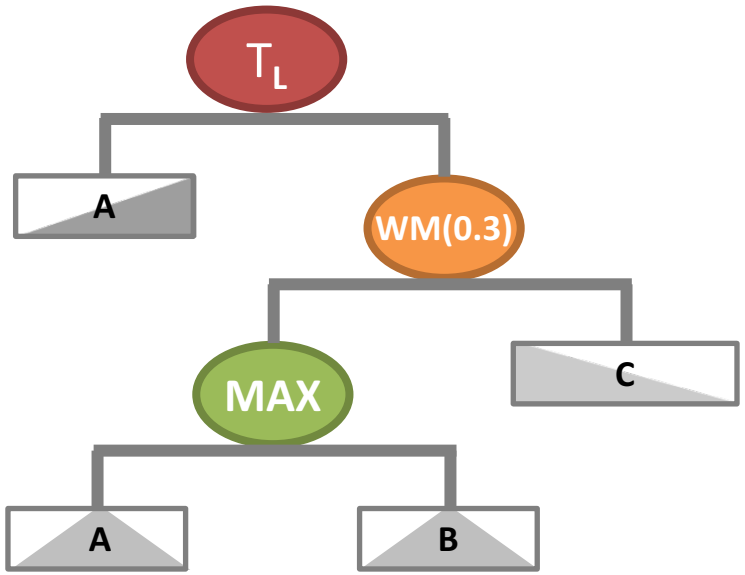
model calibration (parameter estimation)

KNOWLEDGE VS DATA-DRIVEN MODELING

INTELLIGENT SYSTEMS



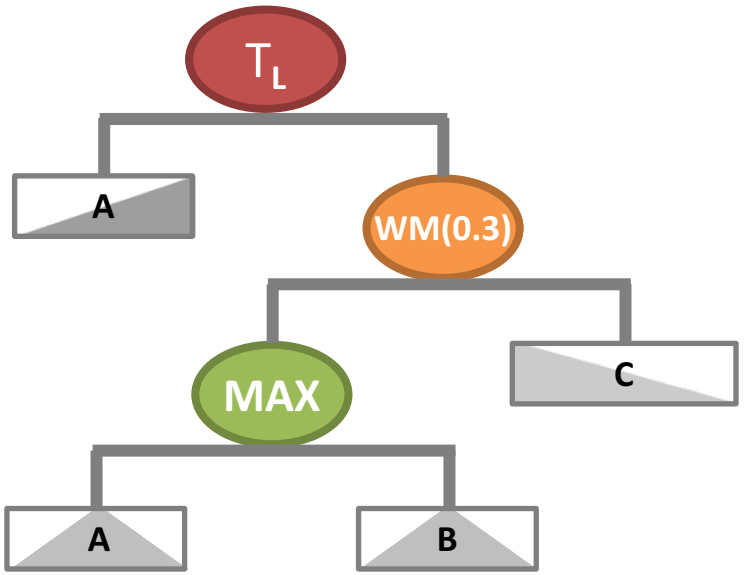
0.6	3	10	1
0.7	2	15	0
0.9	2	16	0
0.3	2	16	1
0.5	6	17	1
...	.	..	.



KNOWLEDGE VS DATA-DRIVEN MODELING

INTELLIGENT SYSTEMS

0.6	3	10	1
0.7	2	15	0
0.9	2	16	0
0.3	2	16	1
0.5	6	17	1
...	.	..	.



learning structure + parameters, purely data-driven model construction

DATA PREPROCESSING

INTELLIGENT SYSTEMS

acidity	alcohol	sulfates	sulfur	quality
7.4	9.4	0.56	11	5
7.8	10	0.46	13	3
7.8	10.5	0.80	25	6
11.2	9.3	0.91	17	3
7.4	9.8	0.55	12	5
7.3	10.6	0.53	21	4
8.9	9.4	0.66	17	8

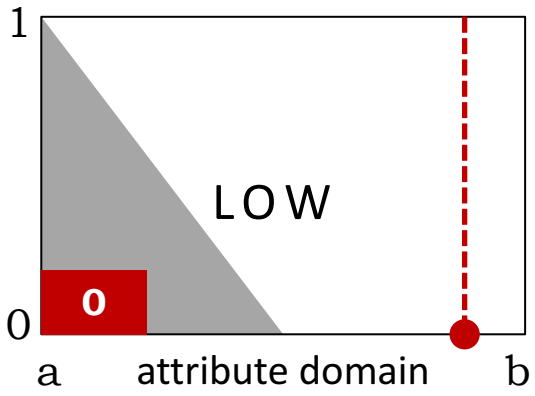
WINE QUALITY DATA: The task is to predict the quality of wine (on a scale from 0 to 10) based on properties like acidity, level of alcohol, etc.

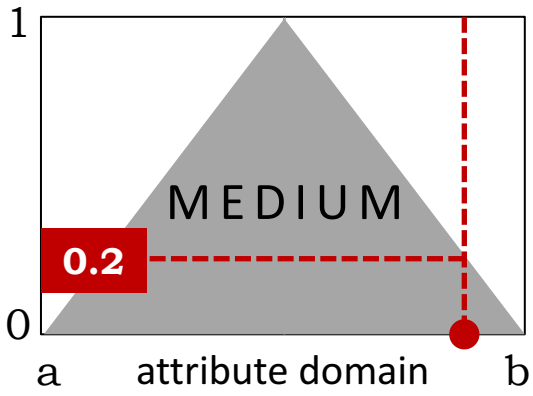
DATA PREPROCESSING

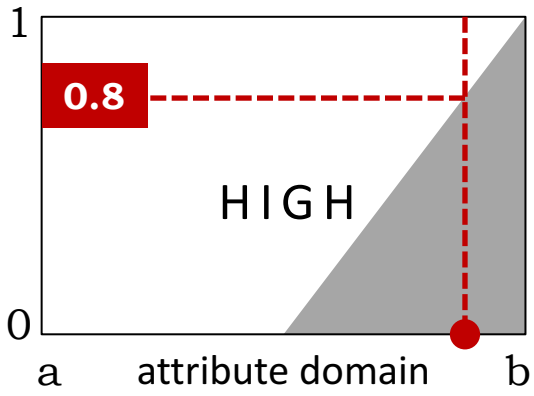
INTELLIGENT SYSTEMS

acidity	alcohol			sulfates	sulfur	quality
	low	*med*	*high*			*G(y)*
7.4	0.89	0.11	0.00	0.56	11	0.50
7.8	0.03	0.97	0.00	0.46	13	0.30
7.8	0.22	0.78	0.00	0.8	25	0.60
11.2	1.00	0.00	0.00	0.91	17	0.30
7.4	0.00	0.00	1.00	0.55	12	0.50
7.3	0.00	0.81	0.19	0.53	21	0.40
8.9	0.84	0.16	0.00	0.66	17	0.80

WINE QUALITY DATA: The task is to predict the quality of wine (on a scale from 0 to 10) based on properties like acidity, level of alcohol, etc.





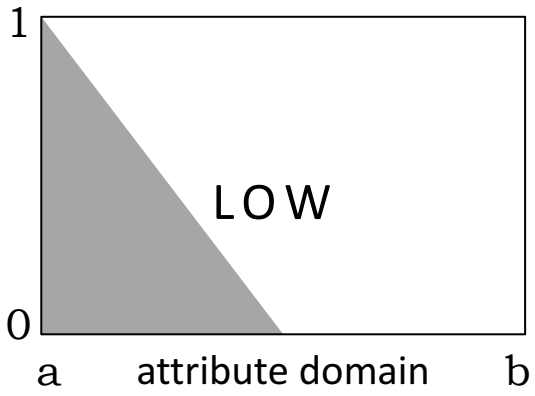


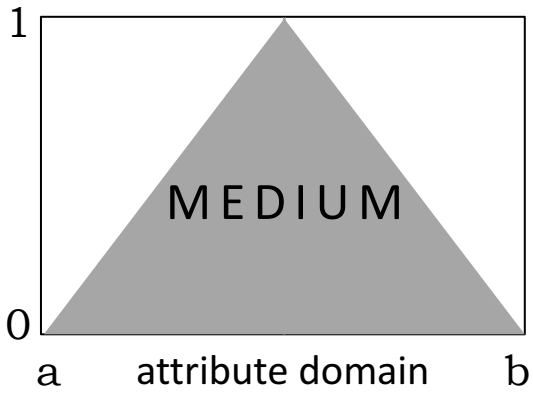
DATA PREPROCESSING

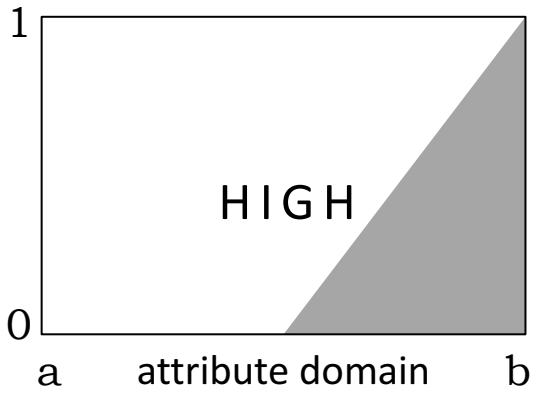
INTELLIGENT SYSTEMS

acidity			alcohol			sulfates			sulfur			quality
			low	*med*	*high*							*G(y)*
			0.89	0.11	0.00							0.50
			0.03	0.97	0.00							0.30
			0.22	0.78	0.00							0.60
			1.00	0.00	0.00							0.30
			0.00	0.00	1.00							0.50
			0.00	0.81	0.19							0.40
			0.84	0.16	0.00							0.80

WINE QUALITY DATA: The task is to predict the quality of wine (on a scale from 0 to 10) based on properties like acidity, level of alcohol, etc.





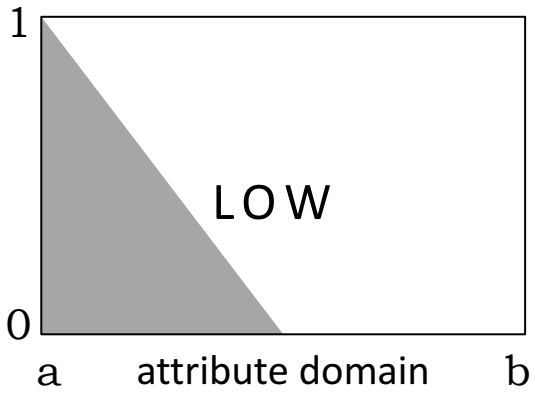


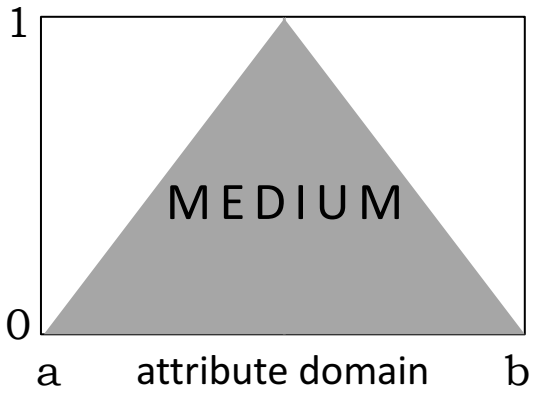
DATA PREPROCESSING

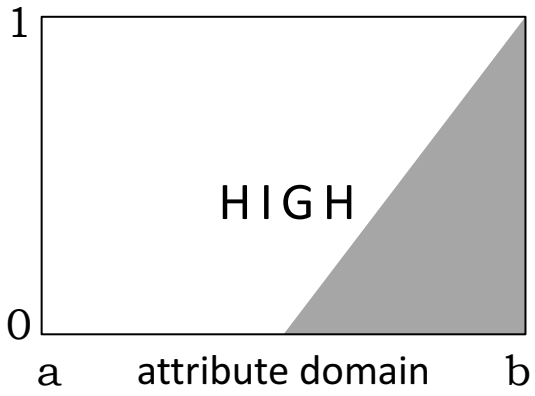
INTELLIGENT SYSTEMS

acidity			alcohol			sulfates			sulfur			quality
			low	*med*	*high*							*G(y)*
			0.89	0.11	0.00							0.50
			0.03	0.97	0.00							0.30
			0.22	0.78	0.00							0.60
			1.00	0.00	0.00							0.30
			0.00	0.00	1.00							0.50
			0.00	0.81	0.19							0.40
			0.84	0.16	0.00							0.80

WINE QUALITY DATA: The task is to predict the quality of wine (on a scale from 0 to 10) based on properties like acidity, level of alcohol, etc.







FUZZY PATTERN TREE INDUCTION

INTELLIGENT SYSTEMS

- Starting **with primitive pattern trees** (fuzzy subset of an attribute's domain),
- **candidate trees** are iteratively expanded and parameterized …
- … and selected based on a tree **performance measure** (MSE on training data),
- until a **stopping condition** is met.

Search for a good model in the space of all pattern trees!

FUZZY PATTERN TREE INDUCTION

INTELLIGENT SYSTEMS

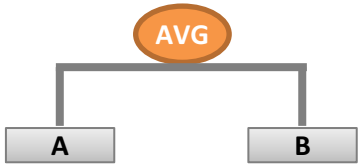
- Starting **with primitive pattern trees** (fuzzy subset of an attribute's domain),

- **candidate trees** are iteratively expanded and parameterized …
- … and selected based on a tree **performance measure** (MSE on training data),
- until a **stopping condition** is met.

FUZZY PATTERN TREE INDUCTION

INTELLIGENT SYSTEMS

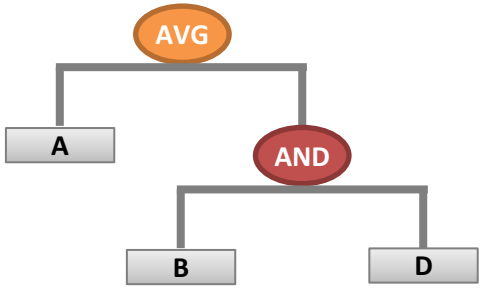
- Starting **with primitive pattern trees** (fuzzy subset of an attribute's domain),
- **candidate trees** are iteratively expanded and parameterized …
- … and selected based on a tree **performance measure** (MSE on training data),
- until a **stopping condition** is met.



FUZZY PATTERN TREE INDUCTION

INTELLIGENT SYSTEMS

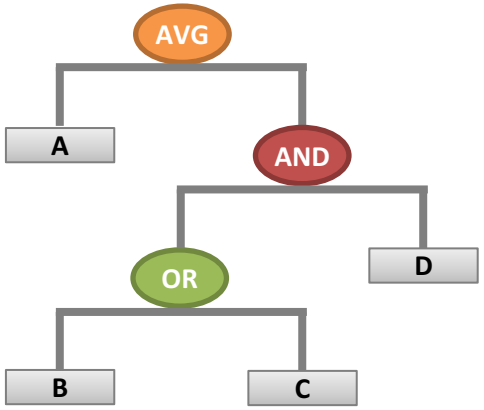
- Starting **with primitive pattern trees** (fuzzy subset of an attribute's domain),
- **candidate trees** are iteratively expanded and parameterized …
- … and selected based on a tree **performance measure** (MSE on training data),
- until a **stopping condition** is met.



FUZZY PATTERN TREE INDUCTION

INTELLIGENT SYSTEMS

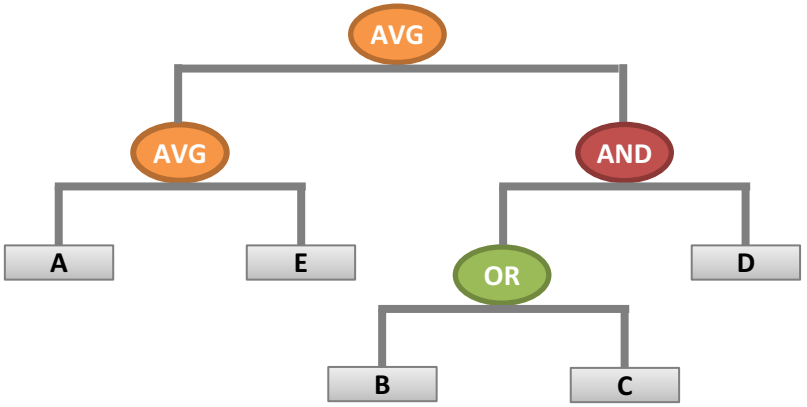
- Starting **with primitive pattern trees** (fuzzy subset of an attribute's domain),
- **candidate trees** are iteratively expanded and parameterized …
- … and selected based on a tree **performance measure** (MSE on training data),
- until a **stopping condition** is met.



FUZZY PATTERN TREE INDUCTION

INTELLIGENT SYSTEMS

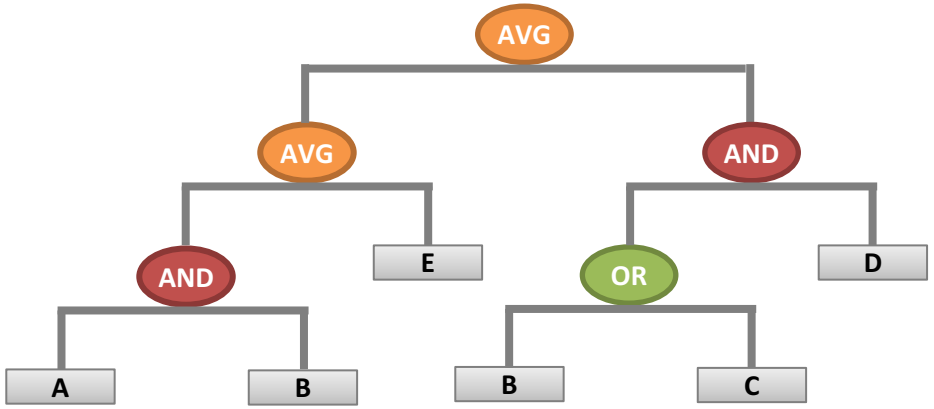
- Starting **with primitive pattern trees** (fuzzy subset of an attribute's domain),
- **candidate trees** are iteratively expanded and parameterized …
- … and selected based on a tree **performance measure** (MSE on training data),
- until a **stopping condition** is met.



FUZZY PATTERN TREE INDUCTION

INTELLIGENT SYSTEMS

- Starting **with primitive pattern trees** (fuzzy subset of an attribute's domain),
- **candidate trees** are iteratively expanded and parameterized …
- … and selected based on a tree **performance measure** (MSE on training data),
- until a **stopping condition** is met.



greedy beam search

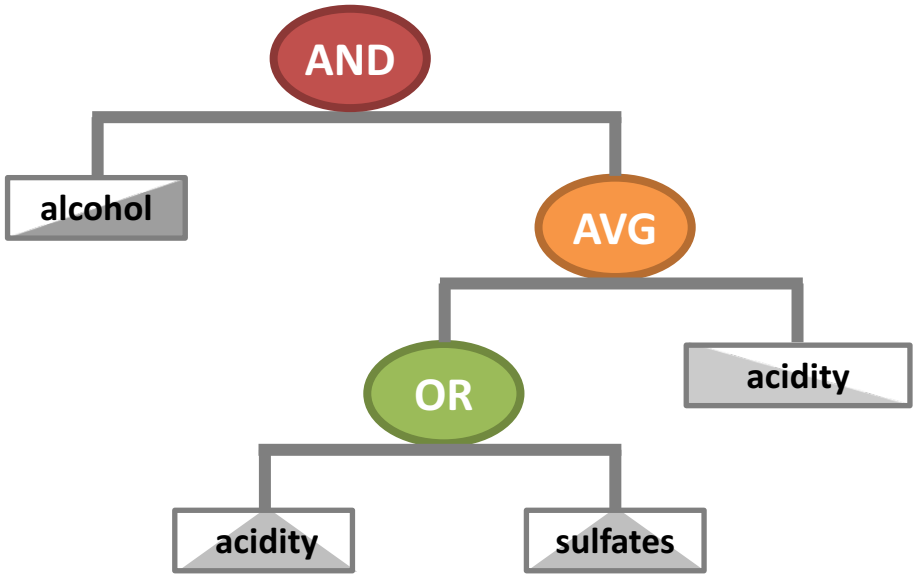
*Iterative construction of **high-level features** from **low-level features**.*

*Instead of **top-down**, pattern trees can also be induced **bottom-up**!*

EXAMPLE: THE WINE QUALITY DATA

INTELLIGENT SYSTEMS

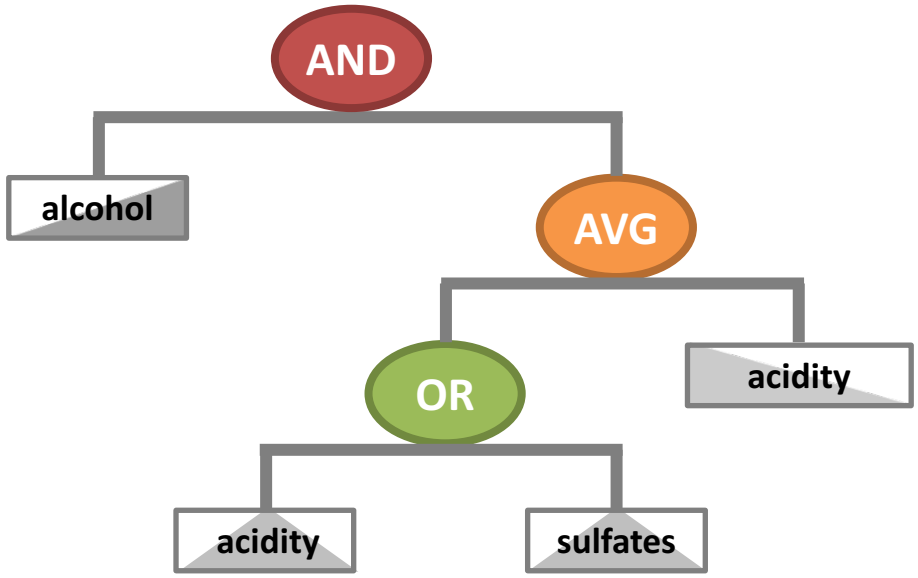
Pattern tree induced from a given set of data (wine properties + rating):



FEATURES OF FUZZY PATTERN TREES

INTELLIGENT SYSTEMS

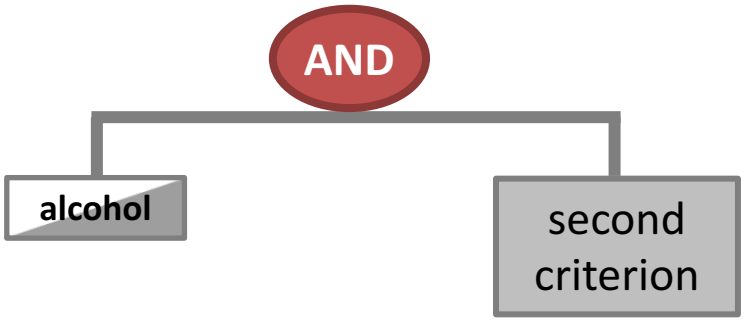
- **interpretability** of the model class
- **modularity:** recursive partitioning of critria into sub-criteria



FEATURES OF FUZZY PATTERN TREES

INTELLIGENT SYSTEMS

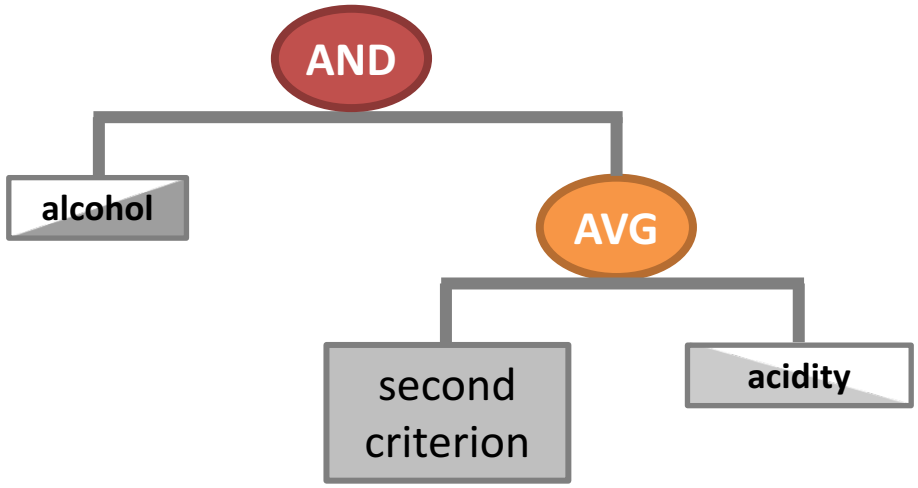
- **interpretability** of the model class
- **modularity:** recursive partitioning of critria into sub-criteria



FEATURES OF FUZZY PATTERN TREES

INTELLIGENT SYSTEMS

- **interpretability** of the model class
- **modularity:** recursive partitioning of critria into sub-criteria

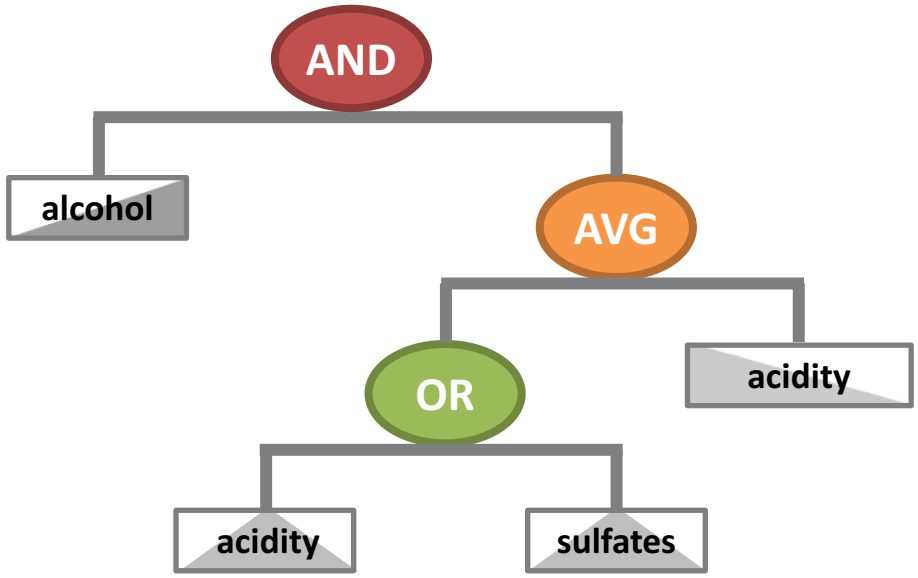


Likewise: type of aggregation vs. concrete parametrization

FEATURES OF FUZZY PATTERN TREES

INTELLIGENT SYSTEMS

- **interpretability** of the model class
- **modularity:** recursive partitioning of critria into sub-criteria
- **flexibility** without the tendency to overfit the data
- **monotonicity** in single attributes
- built-in **feature selection**
- **competitive performance** for **classification** and **regression**



EXPERIMENTS WITH BENCHMARK DATA

INTELLIGENT SYSTEMS

CLASSIFICATION

Dataset	PT-class	C4.5	SVM	RIPPER	NN	SLAVE	
…	…	…	…	…	…	…	
…	…	…	…	…	…	…	
…	…	…	…	…	…	…	
average rank	**2.49**	**2.69**	**3.08**	**3.39**	**3.65**	**3.68**	

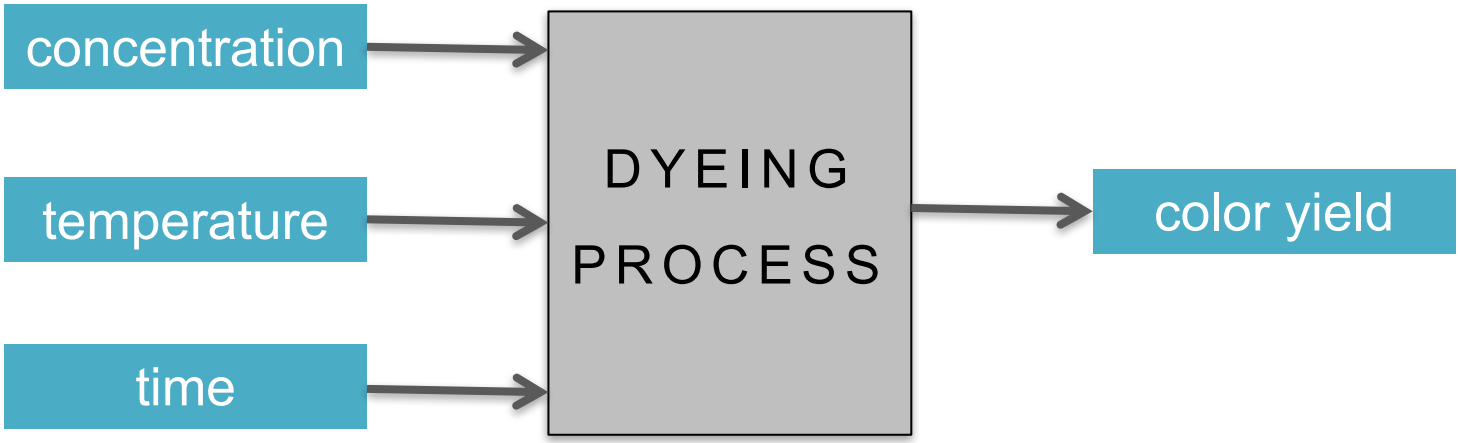
REGRESSION

Dataset	PT-reg	LR	REPtree	SMO-lin	MLP	SMO-rbf	FR
auto-mpg	1	5	4	6	3	7	8
concrete	2	5	1	7	3	6	8
flare1M	6	1	2	5	7	3	8
flare2C	4	1	2	5	7	6	8
forestfires	6	4	3	2	8	1	7
housing	2	5	3	6	1	7	8
imports-85	5	3	7	1	2	6	8
machine	2	6	7	1	8	5	4
servo	2	5	3	7	1	8	6
slump	3	2	7	4	1	6	8
winequality-red	1	2	6	3	7	4	8
winequality-white	4	2	1	3	6	5	8
average rank	**3.17**	**3.42**	**3.83**	**4.17**	**4.5**	**5.33**	**7.42**

CASE STUDY: POLYESTER DYEING

INTELLIGENT SYSTEMS

Modeling of color yield in polyester high temperature dyeing as a function of disperse dyes concentration, temperature and time.

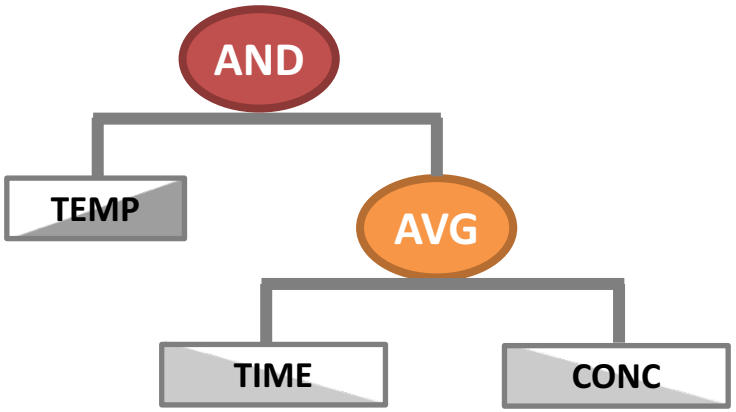


Data: 120 input/output examples for 7 different colors.

CASE STUDY: POLYESTER DYEING

INTELLIGENT SYSTEMS

Example of a fuzzy pattern tree:

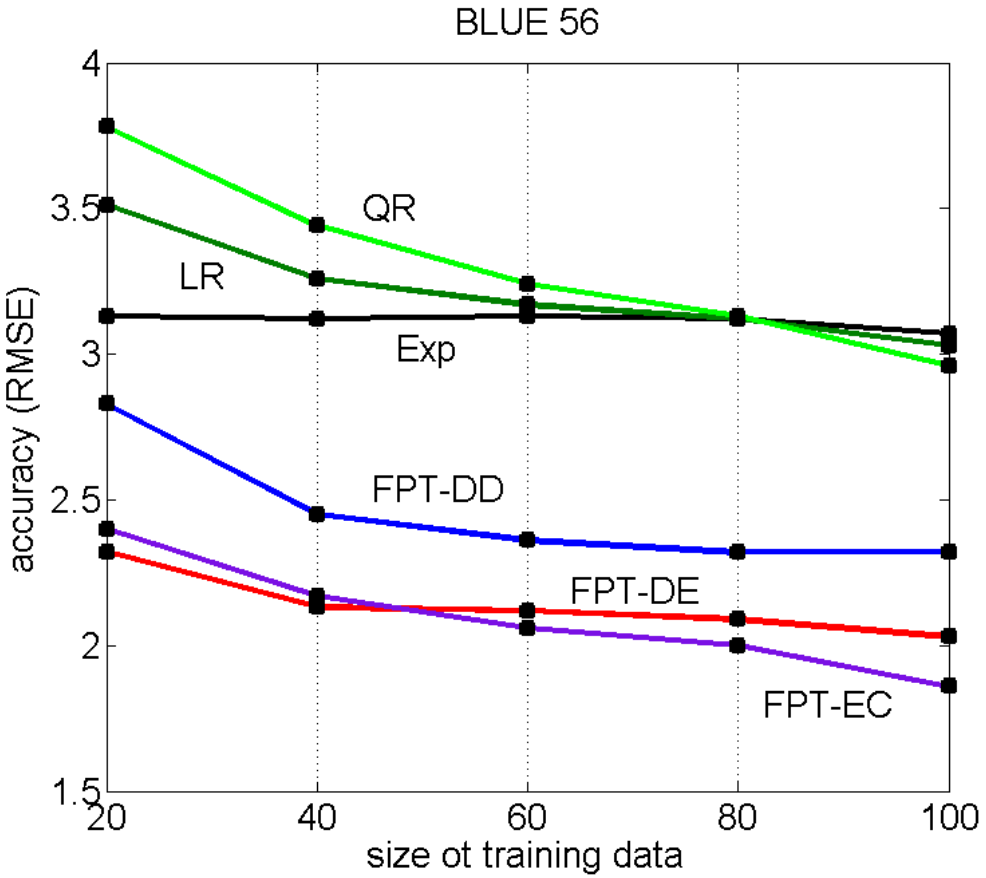


Size of TSK models

Dyes	#rules	Te	Co	Ti
Blue 266	13	3	4	4
Brown 1	8	2	2	3
Blue 56	10	3	2	4
Red 60	9	3	2	2
Yellow 7	9	2	3	2
Yellow 23	13	5	4	3
Mixture	12	4	4	3

In general, TSK (Takagi-Sugeno-Kang) models were judged to be much less comprehensible than the FPT models!

CASE STUDY: POLYESTER DYEING



Average generalization performance as a function of the sample size.

FPT data-driven

FPT data-driven with expert fuzzy sets

FPT model calibration

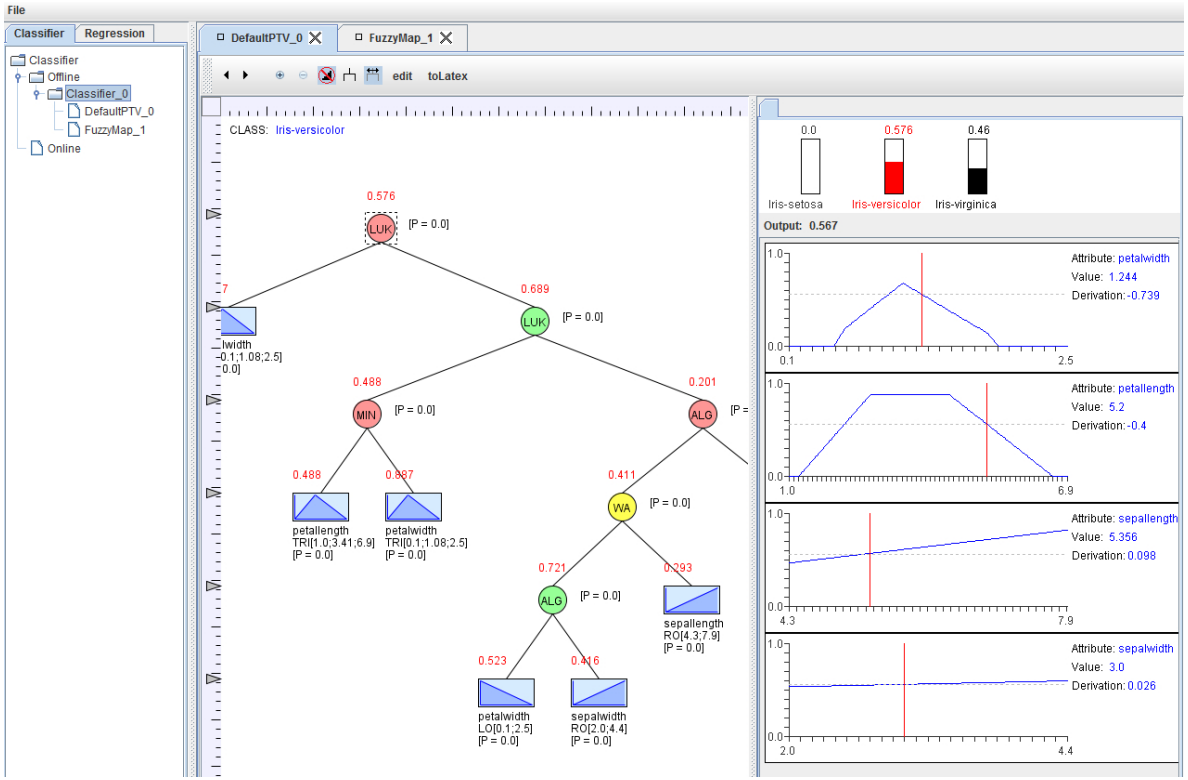
Mamdani models (purely knowledge-driven)

linear regression

quadratic regression

PATTERN TREE SOFTWARE

INTELLIGENT SYSTEMS



Check website of the Intelligent Systems Group @ UPB

OUTLINE

INTELLIGENT SYSTEMS

① Prelude

② Machine learning 101

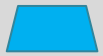
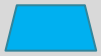
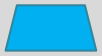
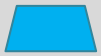
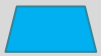
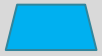
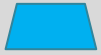
③ Potential contributions of fuzzy logic

④ Fuzzy pattern trees

⑤ Learning from fuzzy data

LEARNING FROM FUZZY DATA

INTELLIGENT SYSTEMS

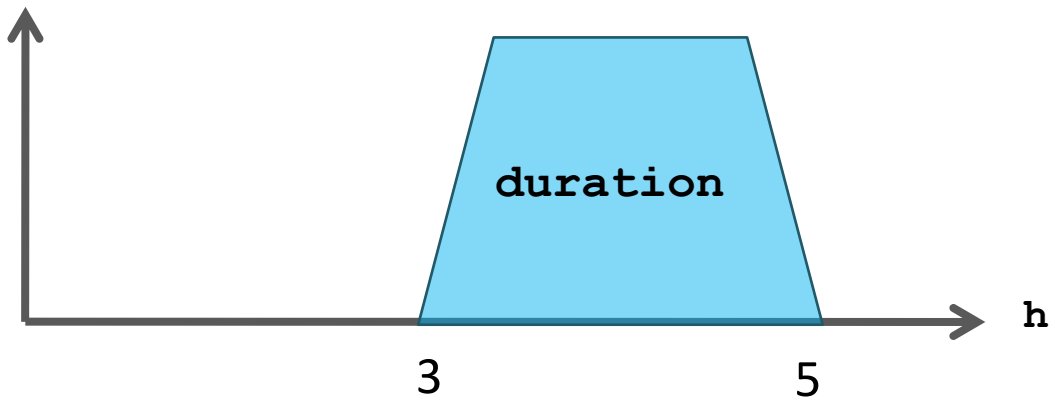
X_1	X_2	X_3	X_4	Y
10	0.42	0	132	10.5
12	0.90	1	154	
	0.61	1		
11		1		94.2
	0.66	0	654	12.6
19		0	127	
32	0.72	1		
15	0.12	0		62.5
...	...	...	...	...

How to analyze and learn from such data?

The „ontic“ view (conjunctive interpretation):

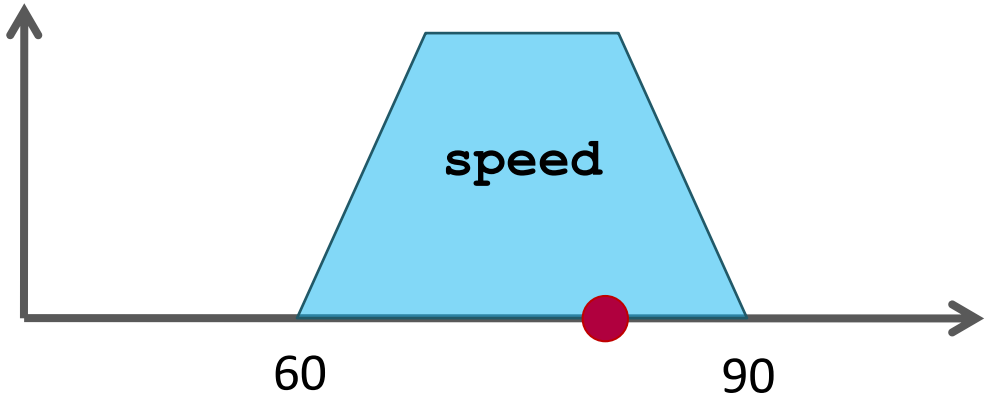
- a fuzzy set is a real data entity;
- an attribute can assume a fuzzy set as a „value“, i.e.,
- we have a (fuzzy set)-valued attribute.

EXAMPLE: X = duration of sunshine in Berlin today (with domain $\mathbb{F}([0, 24])$)



The „epistemic“ view (disjunctive interpretation):

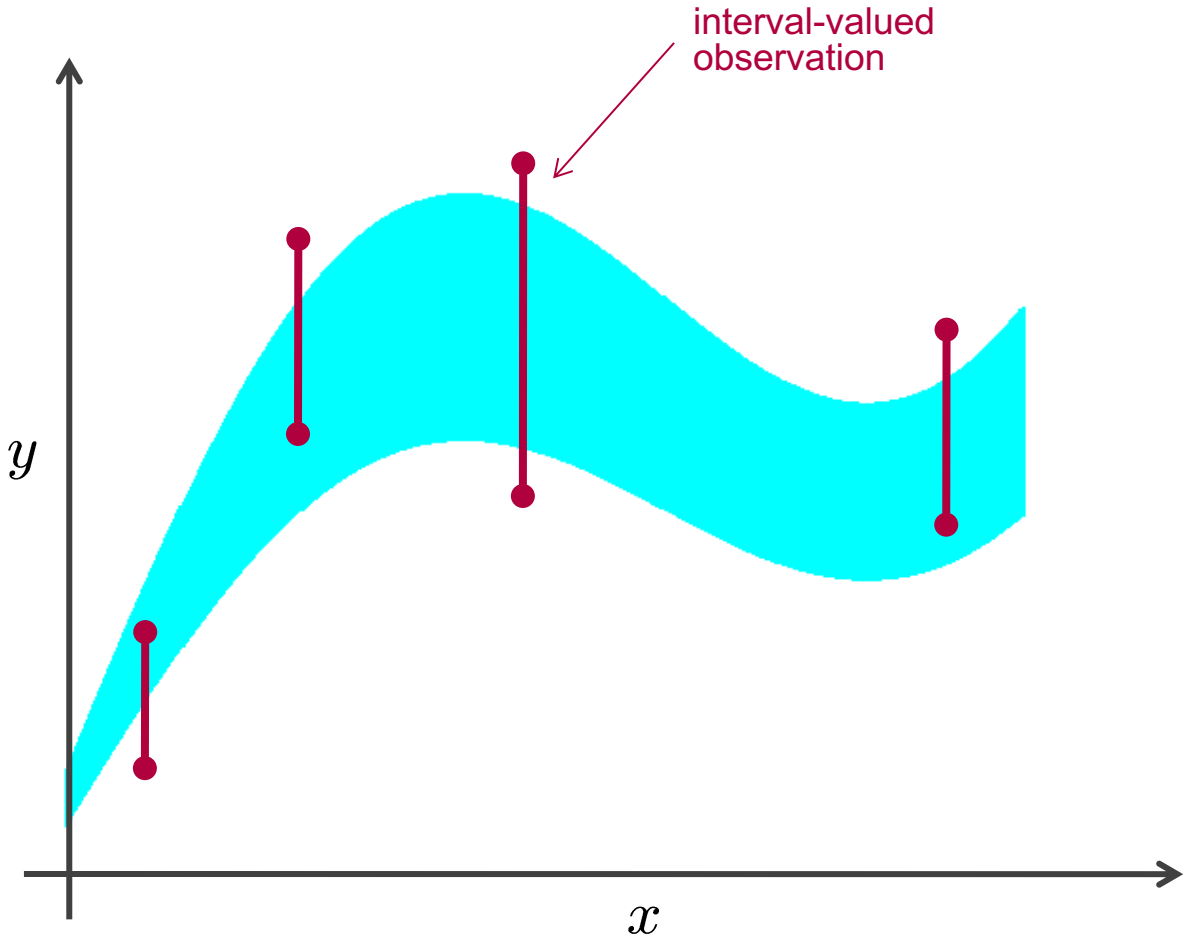
- The true value of the attribute is precise, and a fuzzy set is used to express imprecise knowledge about this value (possibility distribution).



A FUZZY SET IS NOT THE **DATA OBJECT**, BUT REPRESENTS **KNOWLEDGE** ABOUT THIS OBJECT!

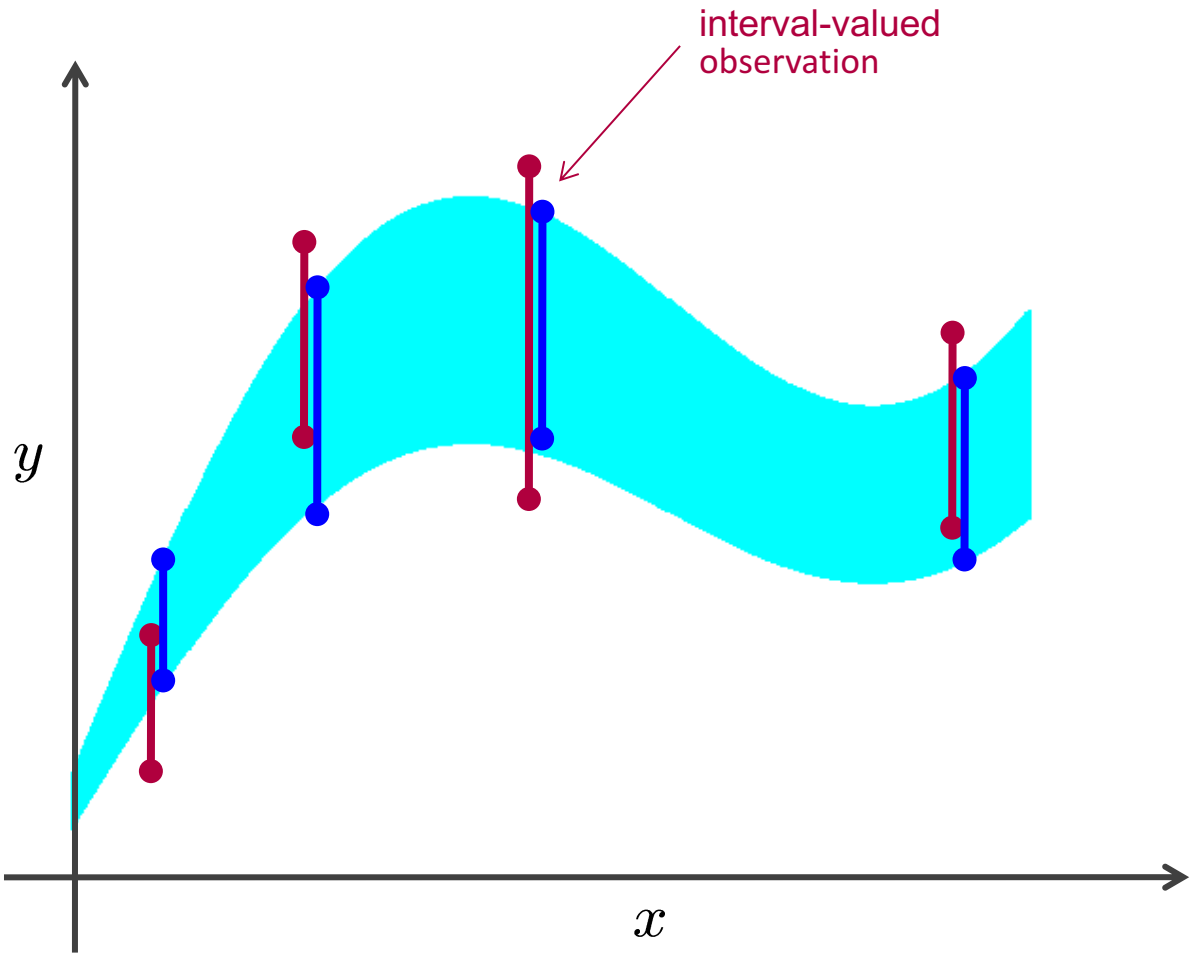
The two interpretations, ontic and epistemic, call for very different extensions of methods for data analysis!

THE ONTIC VIEW



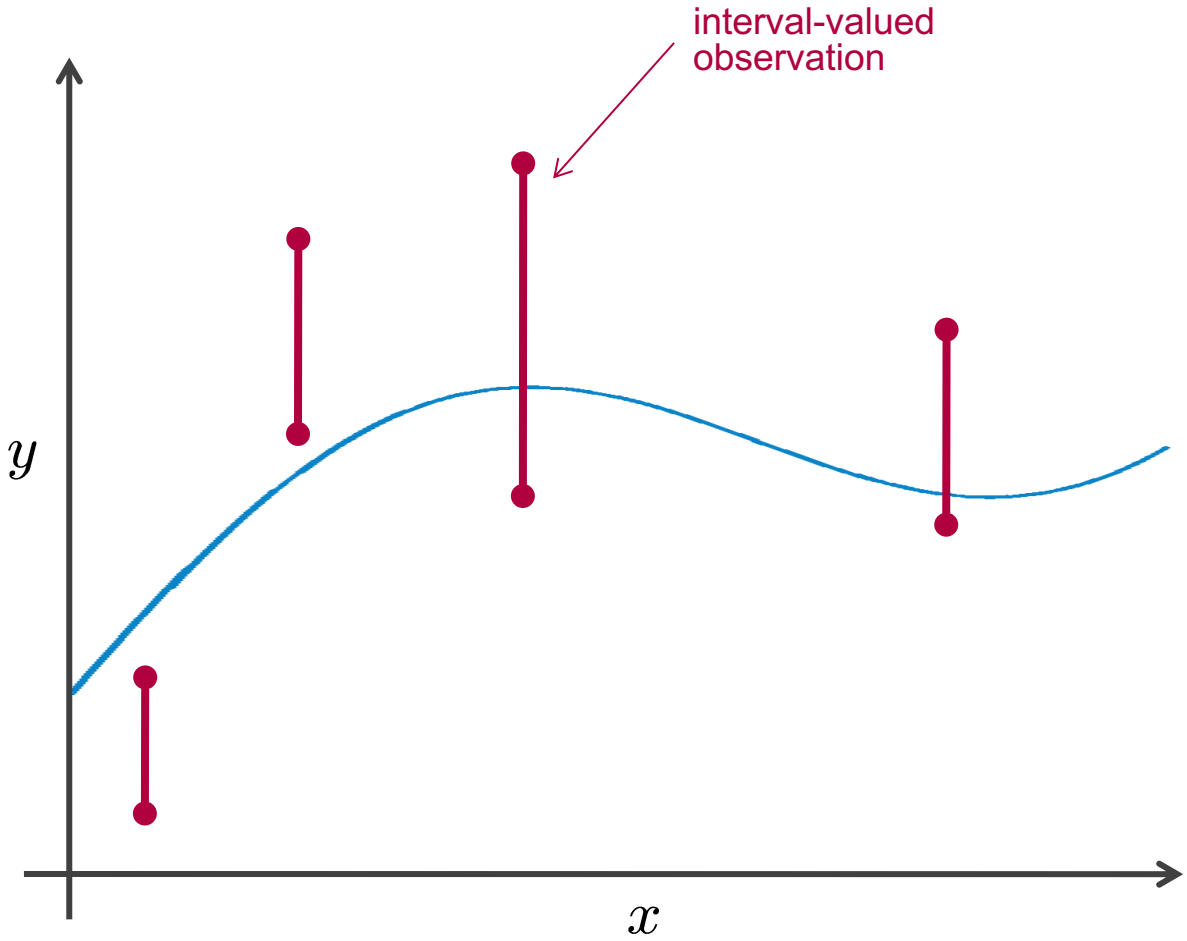
Reproducing imprecise observations by means of a set-valued function!

THE ONTIC VIEW



$$F^* \in \arg\min_{F \in \mathcal{F}} \sum_i D(Y_i,\ F(x_i))$$

THE EPISTEMIC VIEW



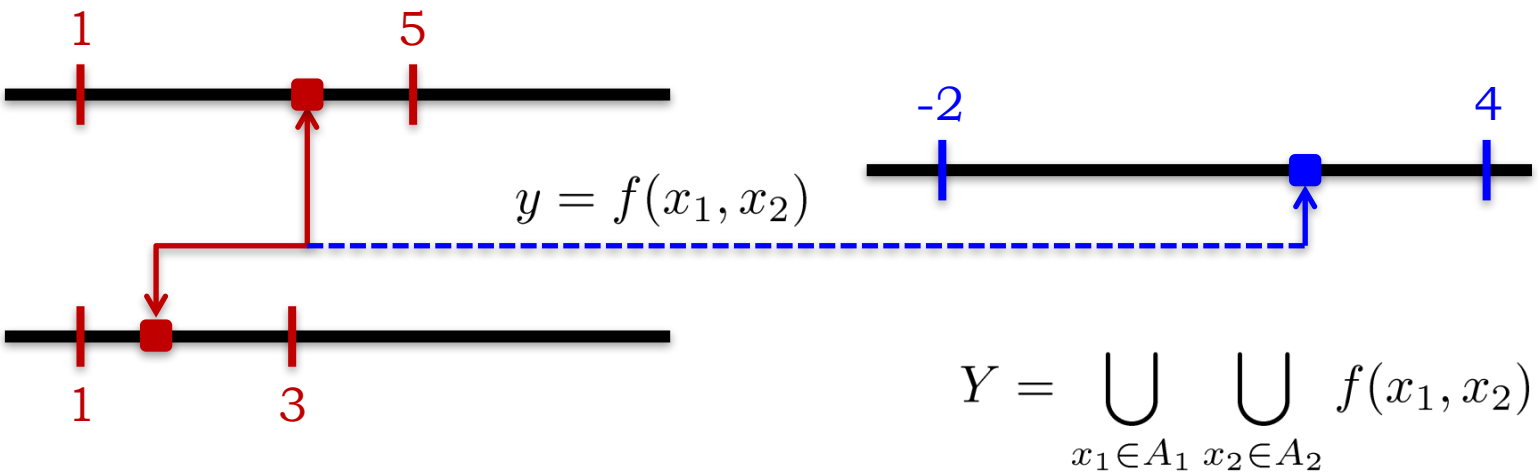
SET of REAL-VALUED functions instead of a single SET-VALUED function!

A model is deemed possible if there is an INSTANTIATION (a possible set of precise observations) for which it yields an optimal fit ...

→ EXTENSION PRINCIPLE (applied to a data analysis method) ?

INTELLIGENT SYSTEMS

- For example, interval arithmetics: $[1,5] \ominus [1,3] = [-2,4]$



$$Y = \bigcup_{x_1 \in A_1} \bigcup_{x_2 \in A_2} f(x_1, x_2)$$

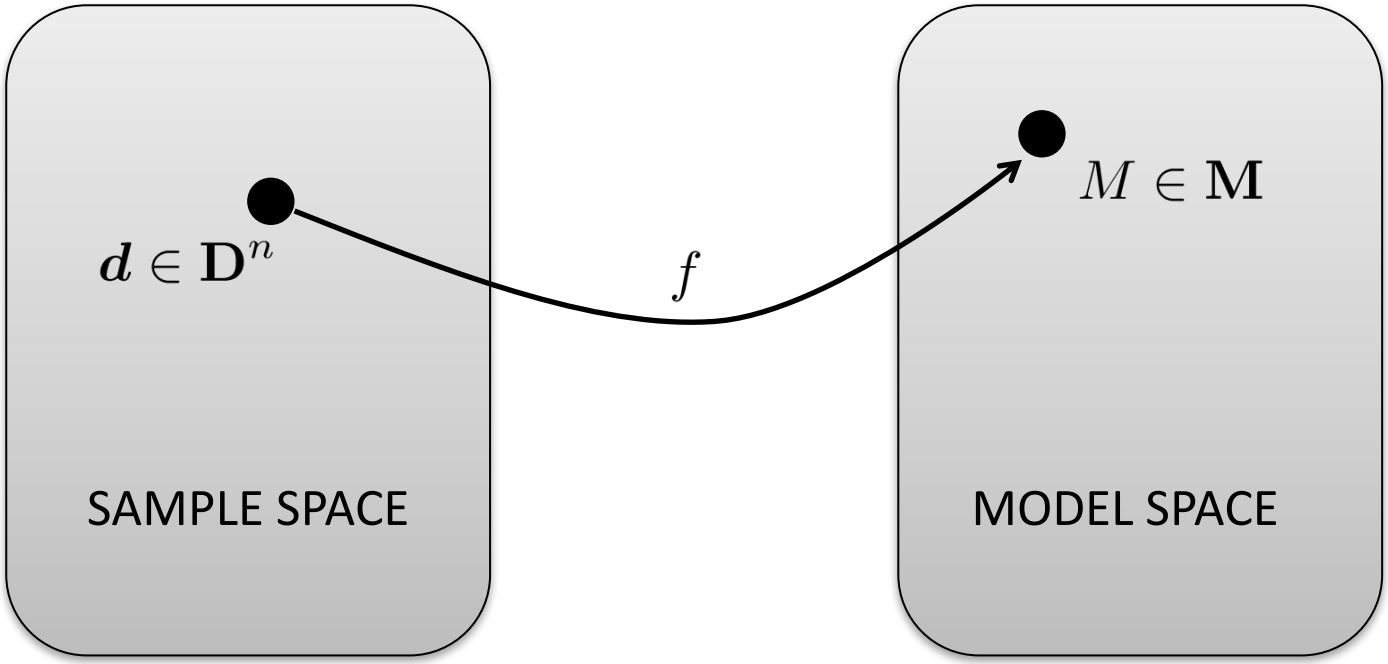
All instantiations of (single-valued) input values are treated the same and equally contribute to the output!

- A learning algorithm is a **mapping from data to models**:

$$f : \mathbf{D}^n \to \mathbf{M}, \boldsymbol{d} = (d_1, \ldots, d_n) \mapsto M$$

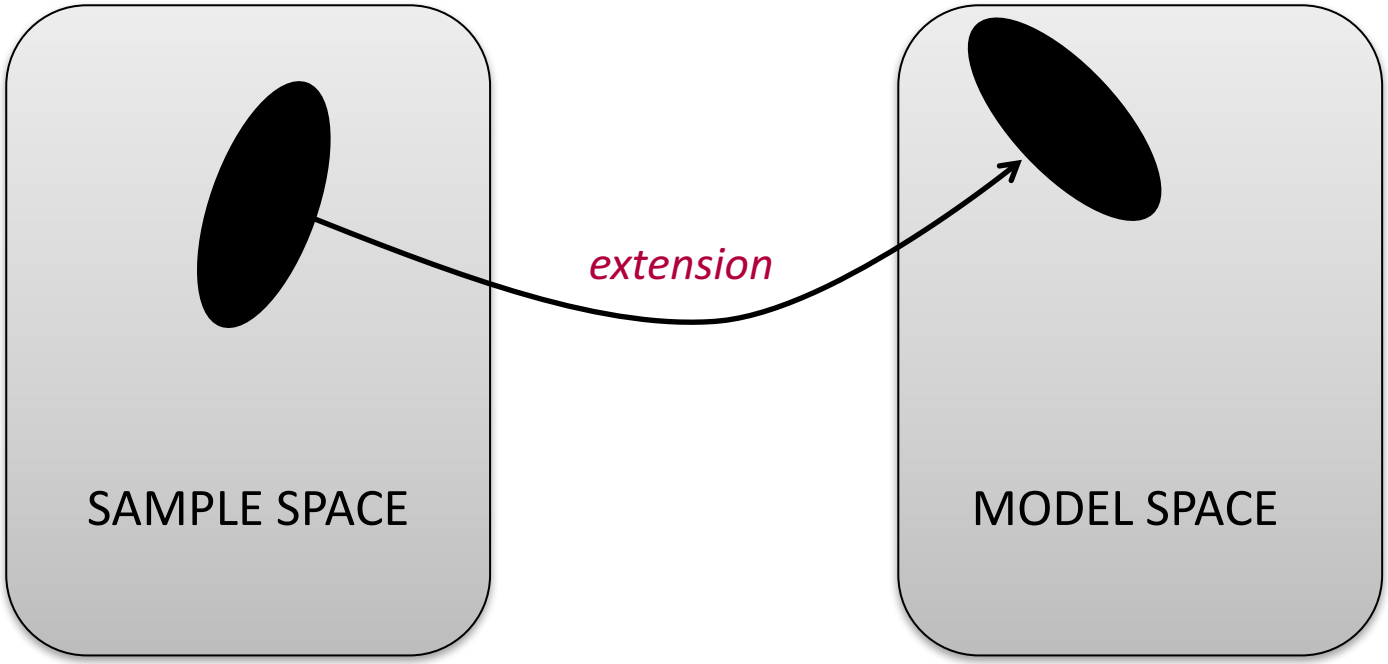
- A learning algorithm is a **mapping from data to models**:

$$f : \mathbf{D}^n \to \mathbf{M},\ \boldsymbol{d} = (d_1, \ldots, d_n) \mapsto M$$



- A learning algorithm is a **mapping from data to models**:

$$f : \mathbf{D}^n \to \mathbf{M}, \boldsymbol{d} = (d_1, \ldots, d_n) \mapsto M$$



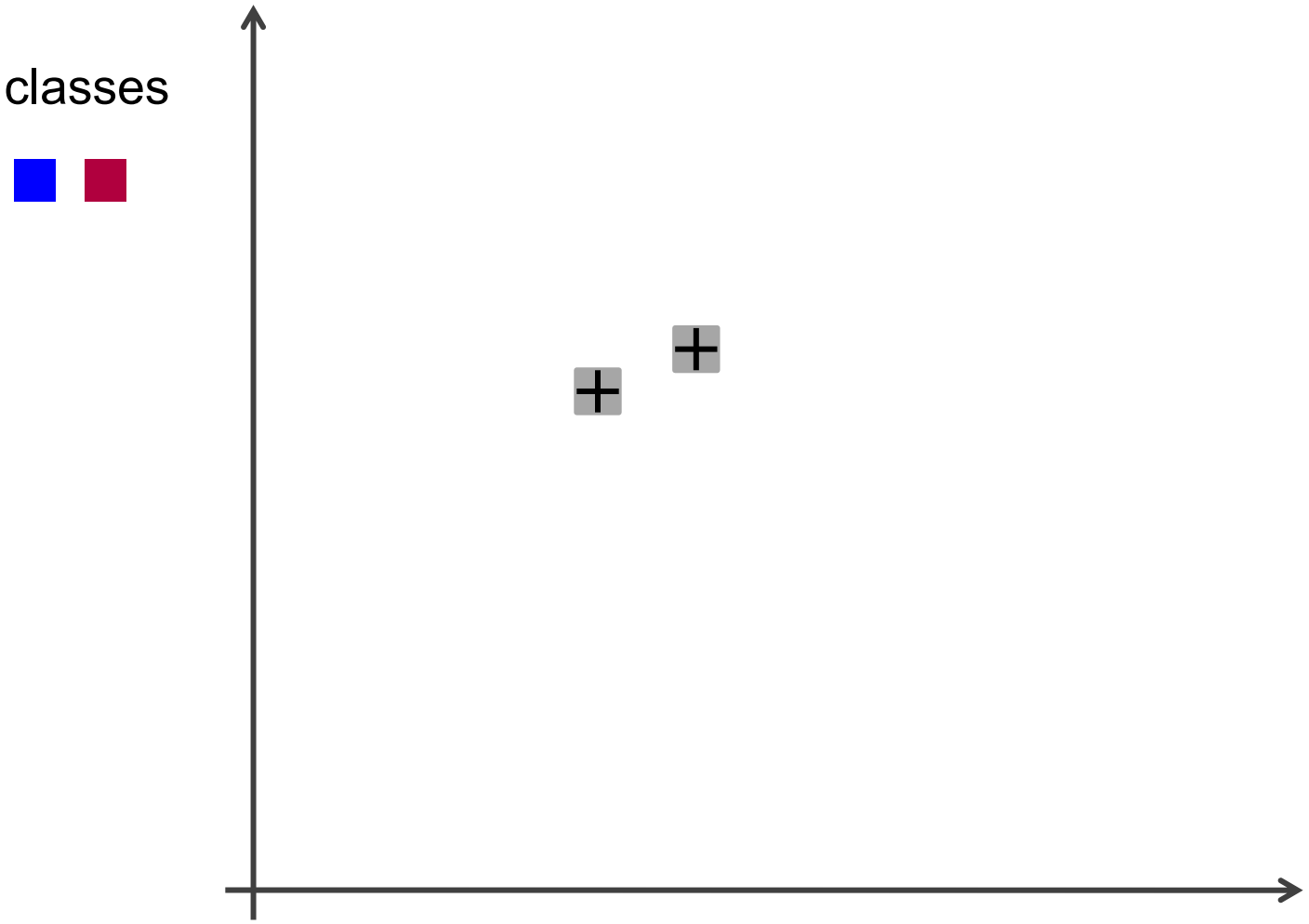
Questioning the equal treatment of all instantiations ...

In data analysis, a method inducing a model from a set of data always comes with certain **MODEL ASSUMPTIONS** (learning bias), and under these assumptions, specific instantiations may appear more plausible than others!

... to be explained through some simple examples.

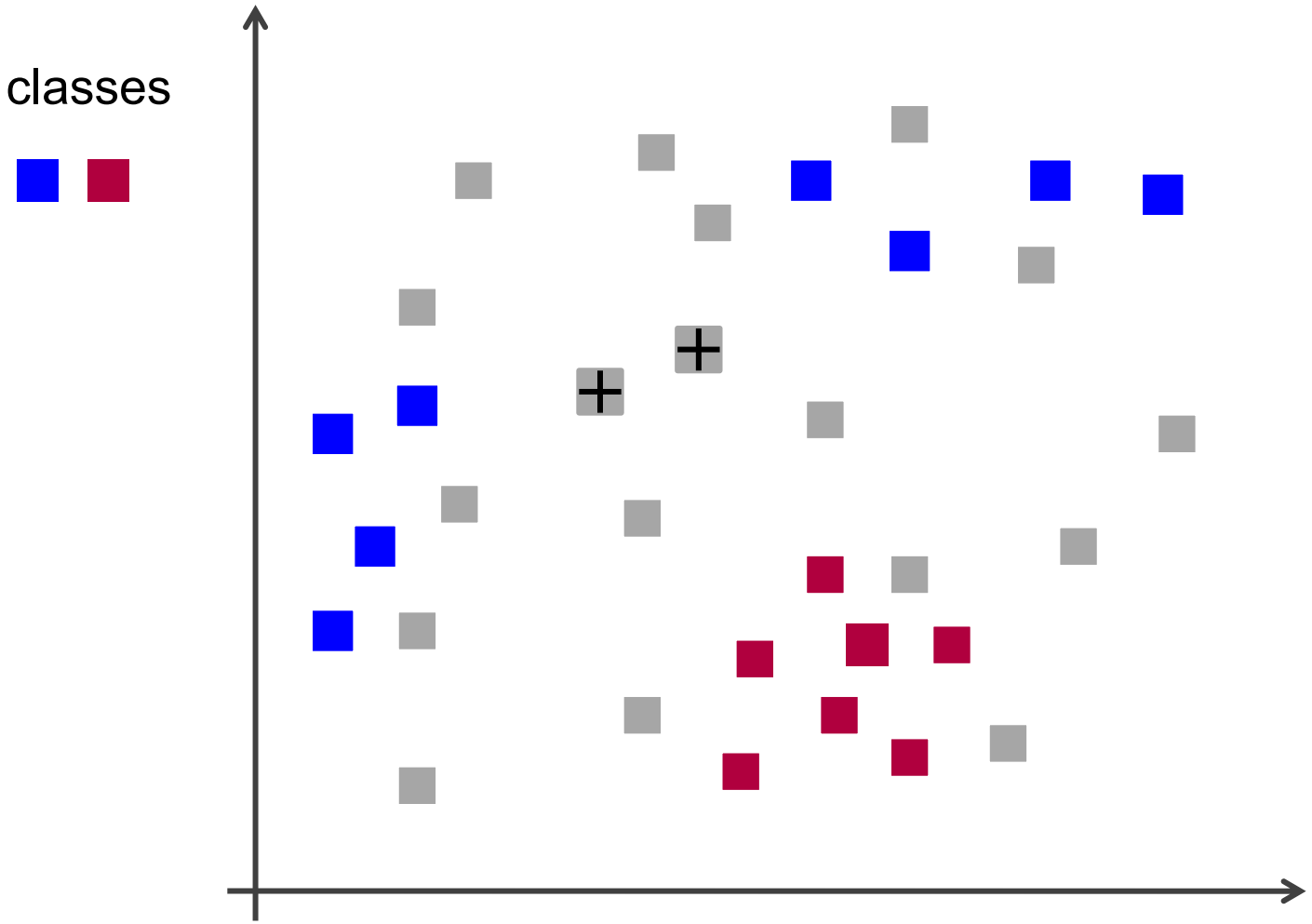
DATA DISAMBIGUATION

INTELLIGENT SYSTEMS



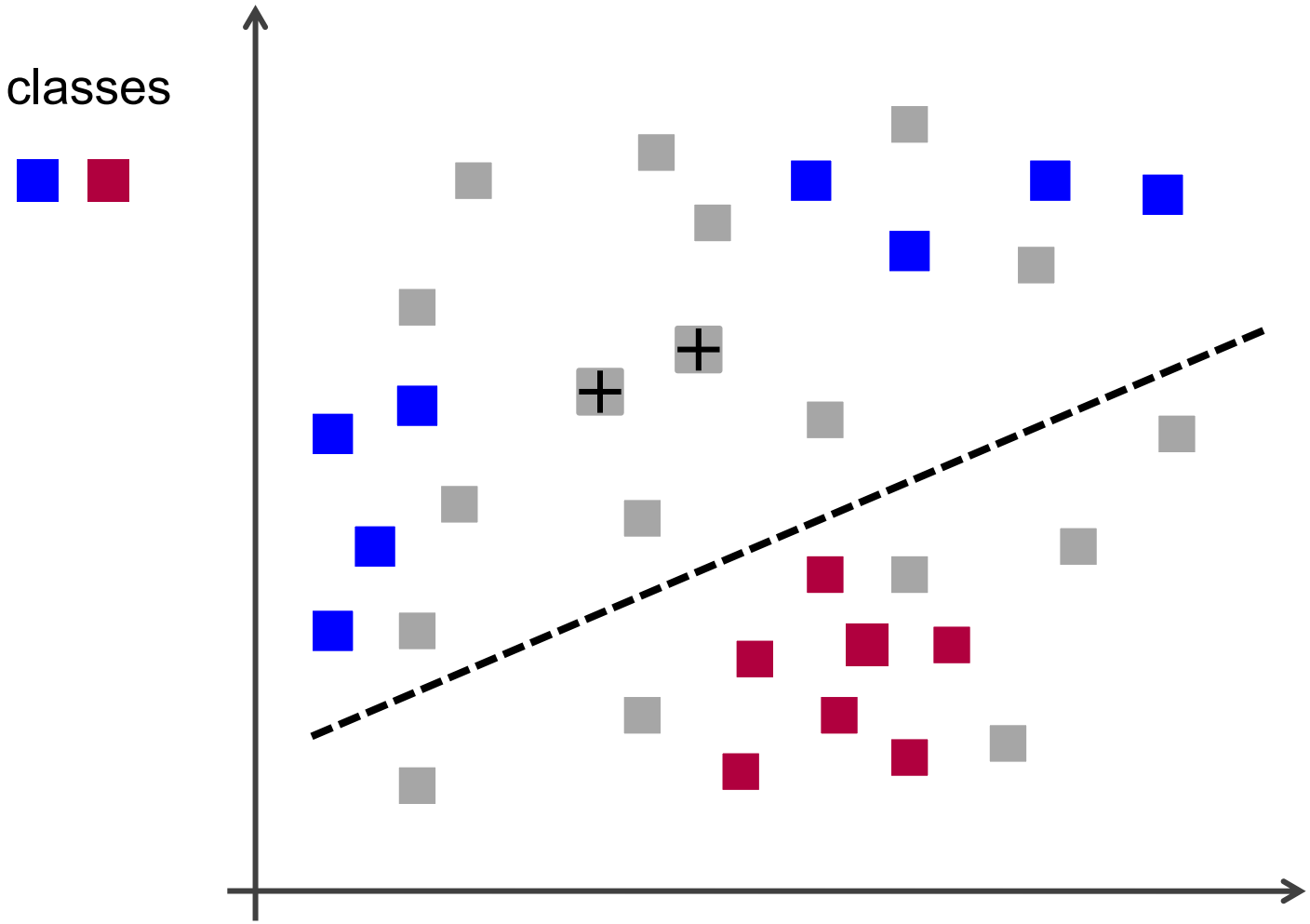
DATA DISAMBIGUATION

INTELLIGENT SYSTEMS



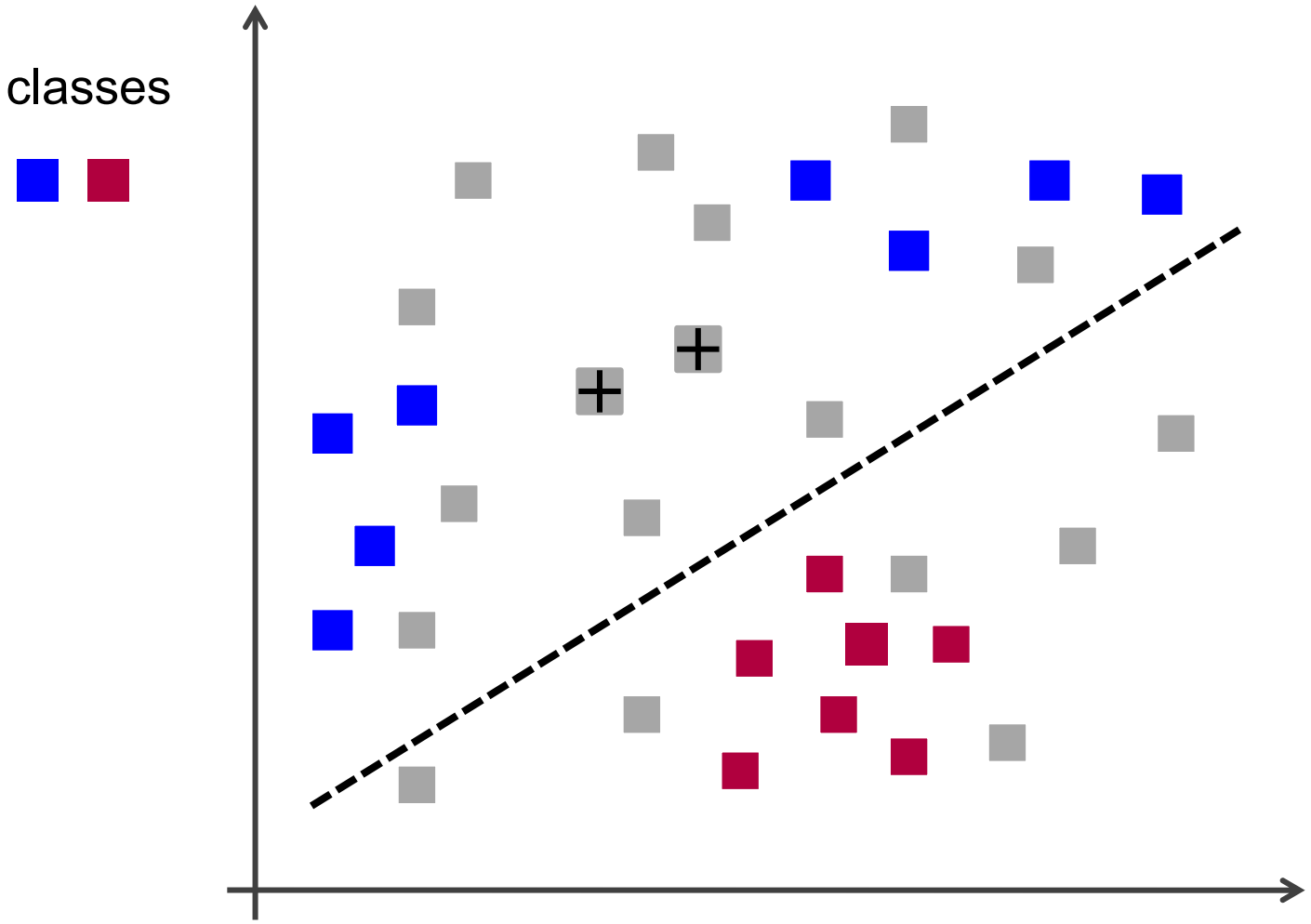
DATA DISAMBIGUATION

INTELLIGENT SYSTEMS



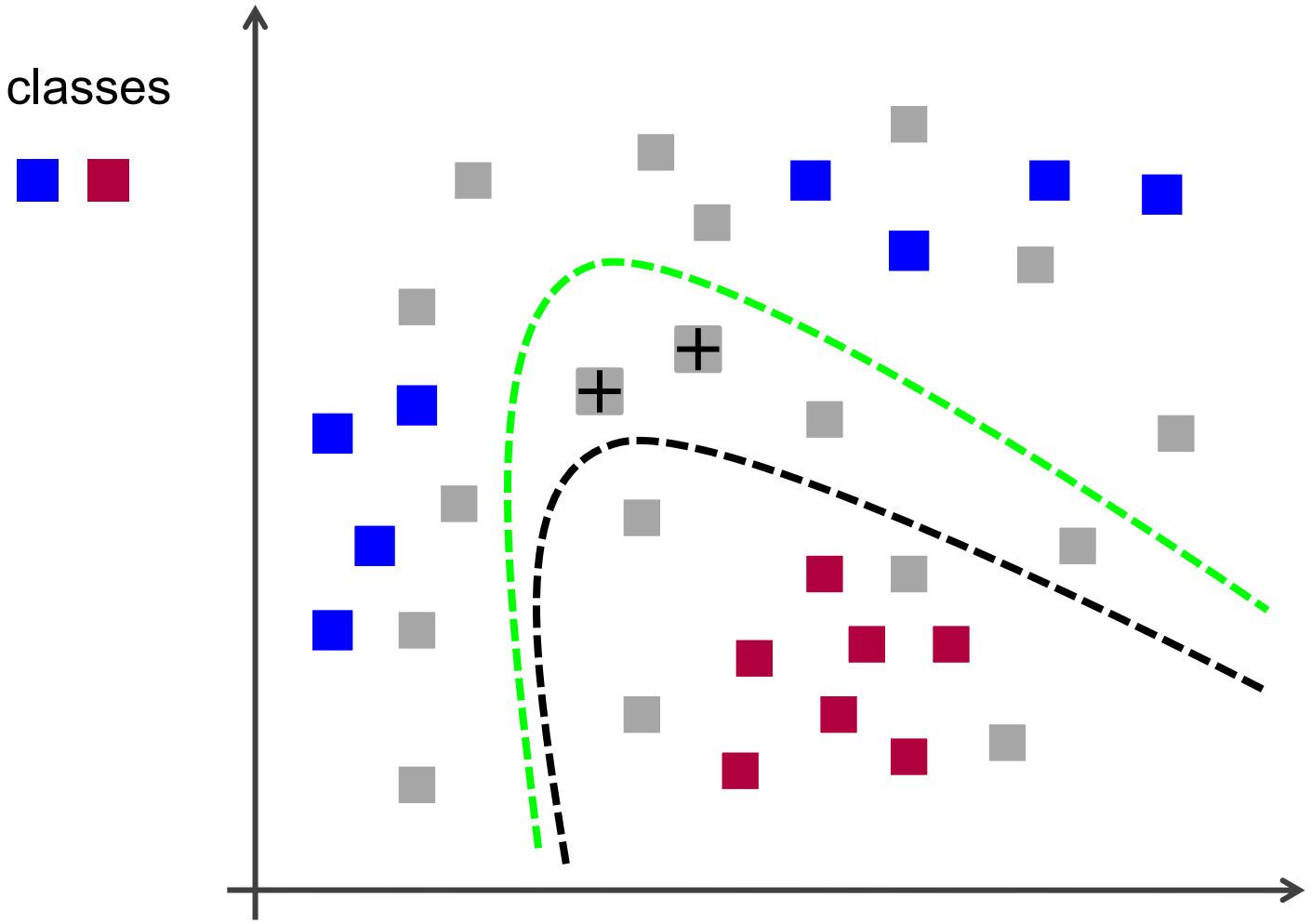
DATA DISAMBIGUATION

INTELLIGENT SYSTEMS



DATA DISAMBIGUATION

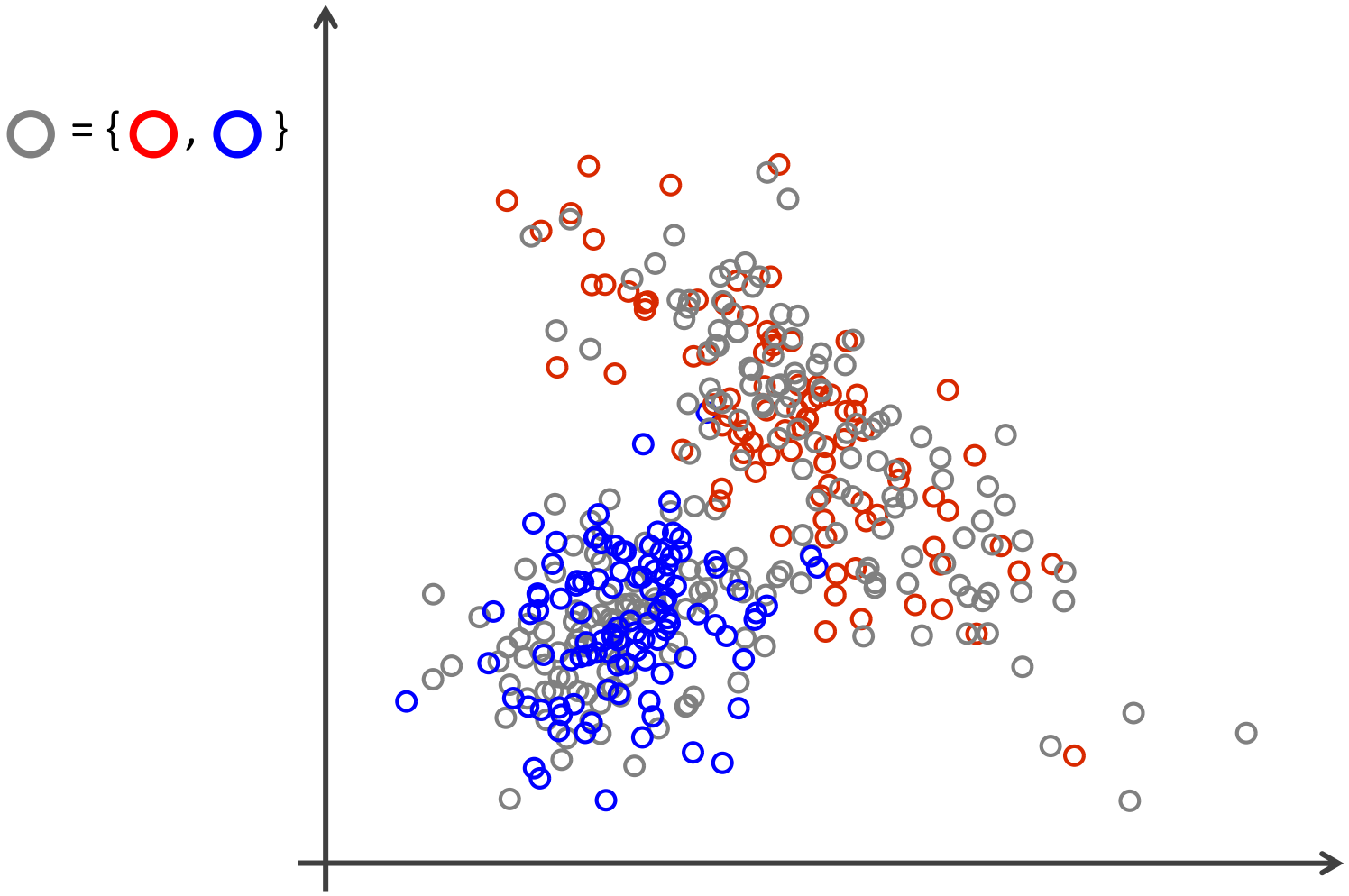
INTELLIGENT SYSTEMS



The more biased the view, the less ambiguous the data looks like.

DATA DISAMBIGUATION

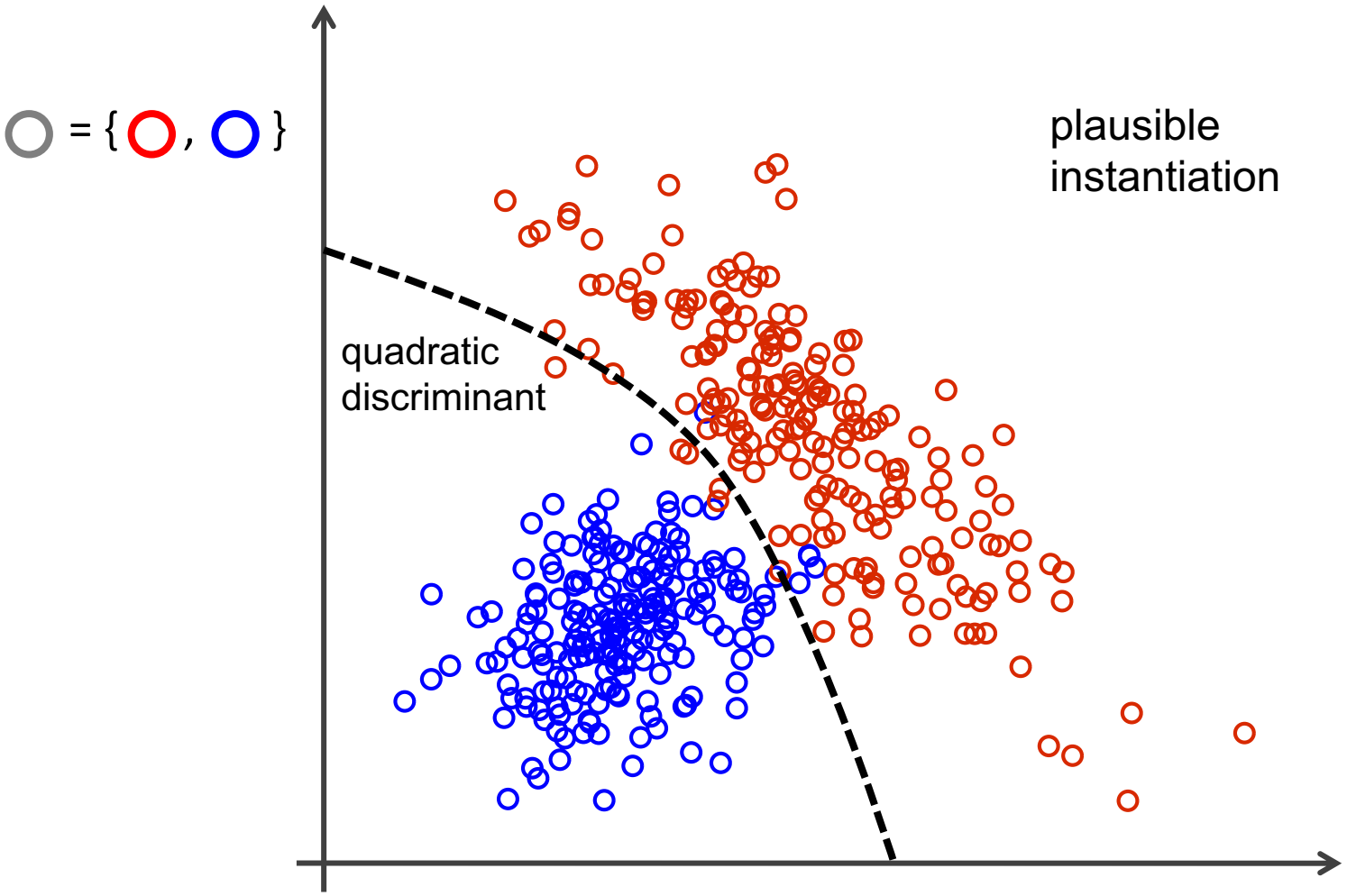
INTELLIGENT SYSTEMS



assume both class distributions to be Gaussian

DATA DISAMBIGUATION

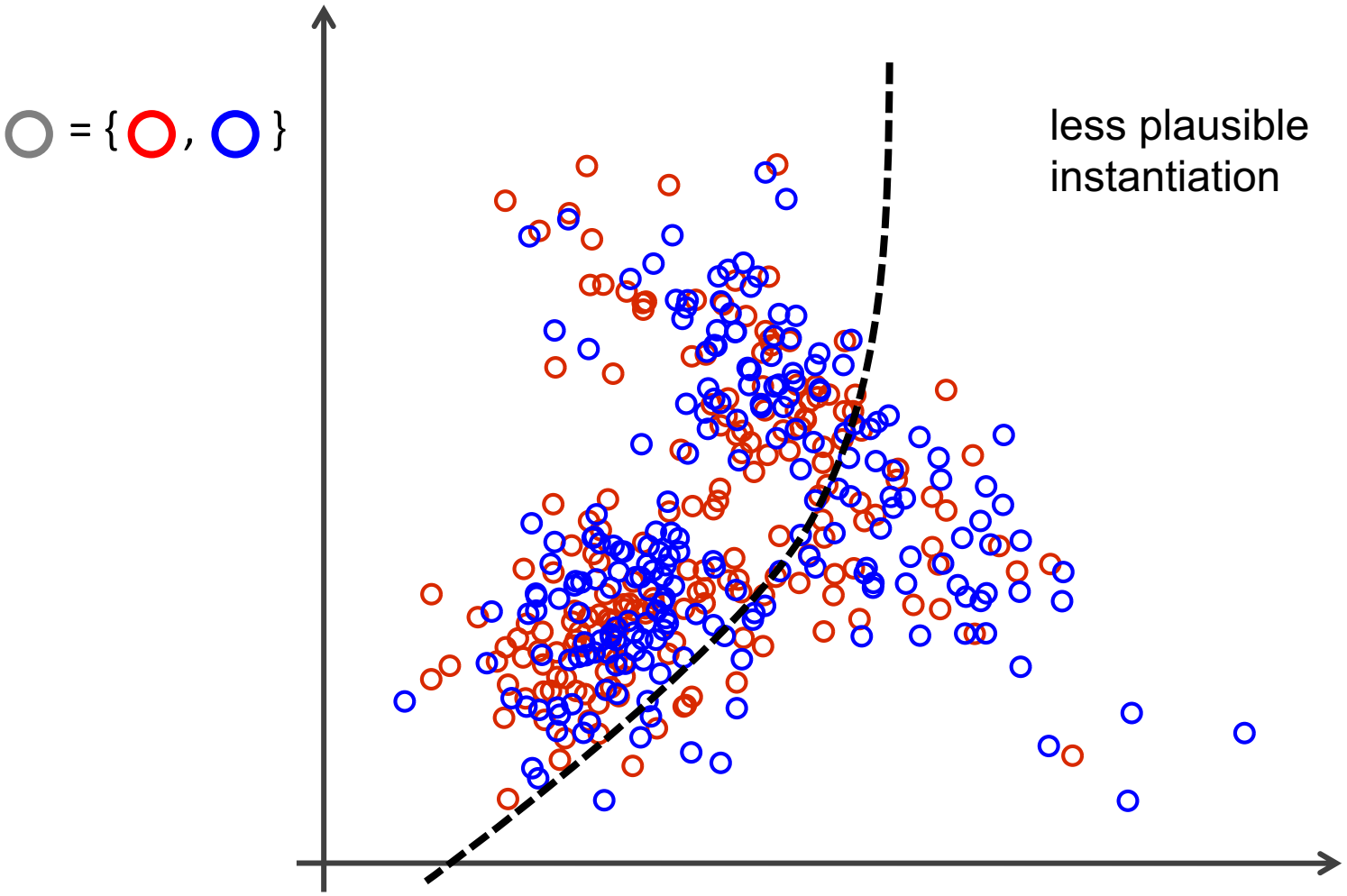
INTELLIGENT SYSTEMS



assume both class distributions to be Gaussian

DATA DISAMBIGUATION

INTELLIGENT SYSTEMS

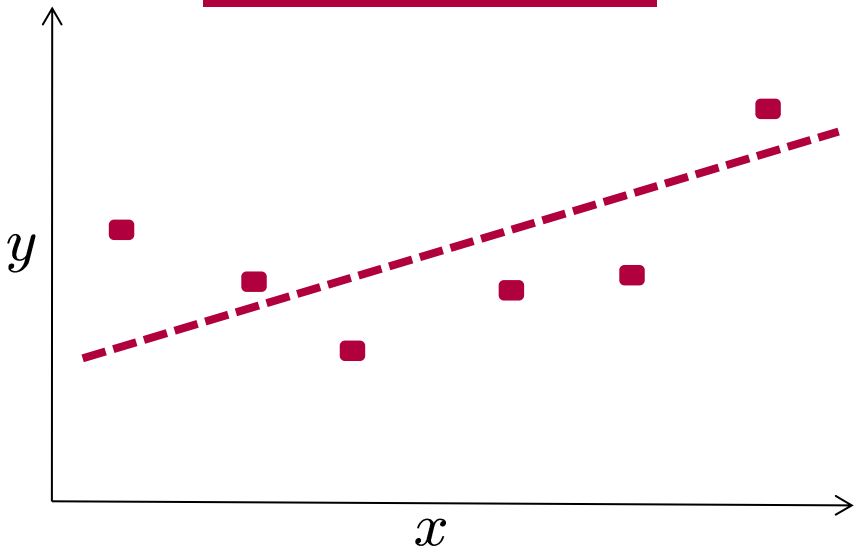


assume both class distributions to be Gaussian

DATA DISAMBIGUATION

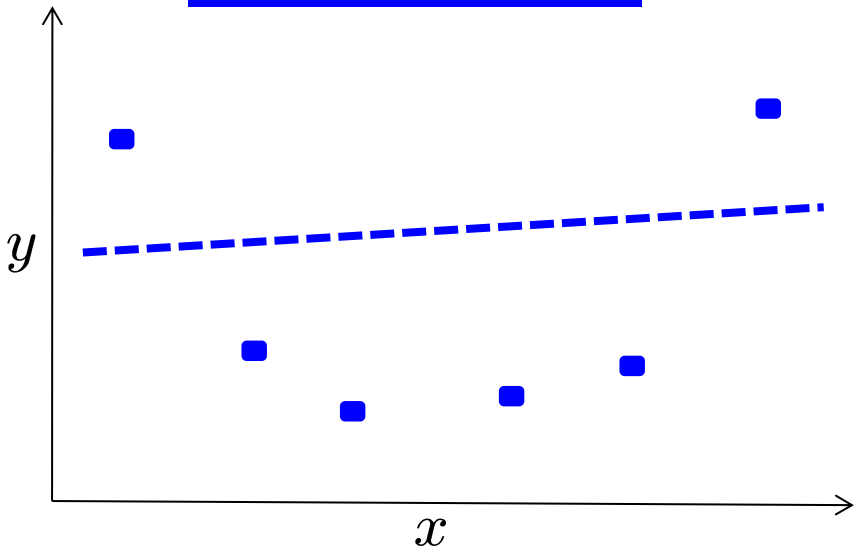
INTELLIGENT SYSTEMS

MORE PLAUSIBLE



A plausible instantiation that can be fitted reasonably well with a **LINEAR** model!

LESS PLAUSIBLE

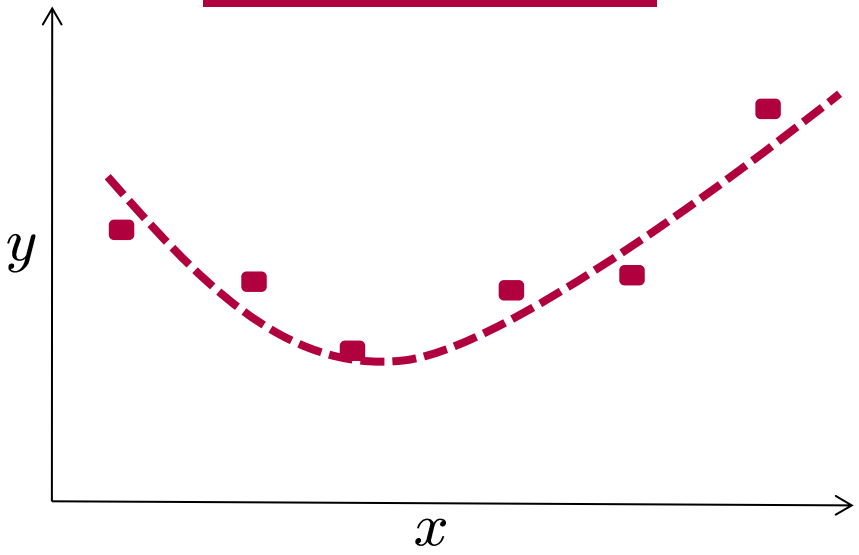


A less plausible instantiation, because there is no **LINEAR** model with a good fit!

DATA DISAMBIGUATION

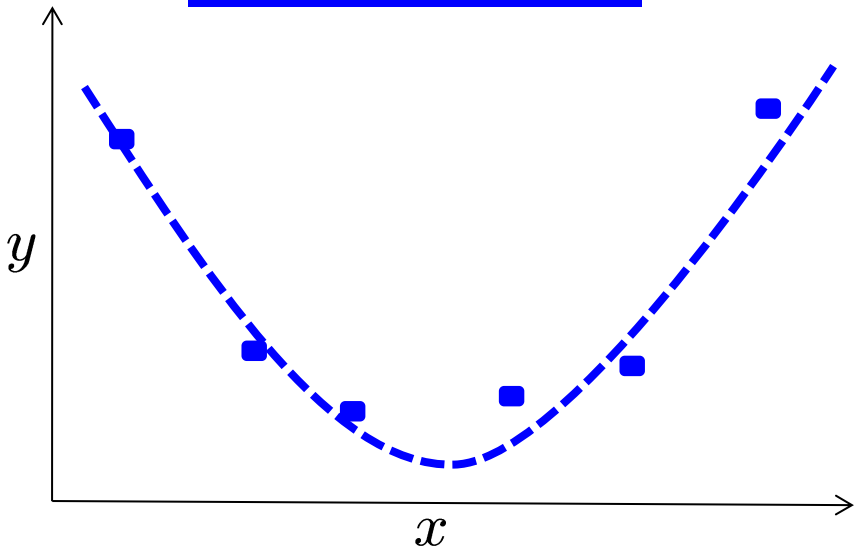
INTELLIGENT SYSTEMS

PLAUSIBLE



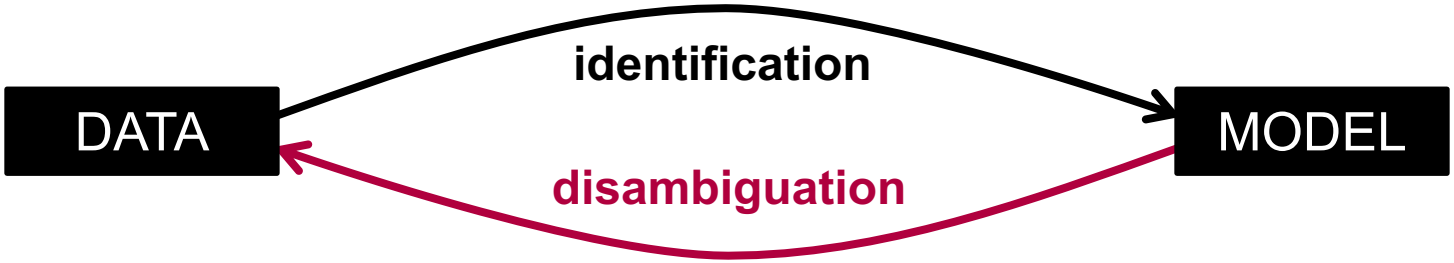
A plausible instantiation that can be fitted quite well with a **QUADRATIC** model!

PLAUSIBLE



A plausible instantiation that can be fitted quite well with a **QUADRATIC** model!

It all depends on how you look at the data!



- In the setting of supervised learning with discriminative models, we suggest that model identification and data disambiguation can support each other, and should be performed simultaneously.
- Not only the data is telling us something about the model, but also the model (assumptions) about the data.

SUPERSET LEARNING

INTELLIGENT SYSTEMS

... is a specific type of **weakly supervised learning**, studied under different names in machine learning:

- *learning from partial labels*
- *multiple label learning*
- *learning from ambiguously labeled examples*
- *...*

... also connected to learning from **coarse data** in statistics (Rubin, 1976; Heitjan and Rubin, 1991), missing values, **data augmentation** (Tanner and Wong, 2012),

... as well as data modeling based on **generalized sets and measures**, such as **fuzzy data** (Kwakernaak, 1978; Kruse and Meyer, 1987; Puri and Ralescu, 1986; Coppi et al., 2006; Bandemer and Näther, 2011; Viertl, 2011) and **belief functions** (Denoeux, 1995).

Given a set of (i.i.d.) **training data**

$$\mathcal{D} = \Big\{ (\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N) \Big\} \subset \mathcal{X} \times \mathcal{Y}$$

and a **hypothesis space** $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$, find a model with low **risk**

$$\mathcal{R}(h) = \int_{\mathcal{X} \times \mathcal{Y}} L\big(h(\boldsymbol{x}), y\big)\, d\,\mathbf{P}(\boldsymbol{x}, y)\,.$$

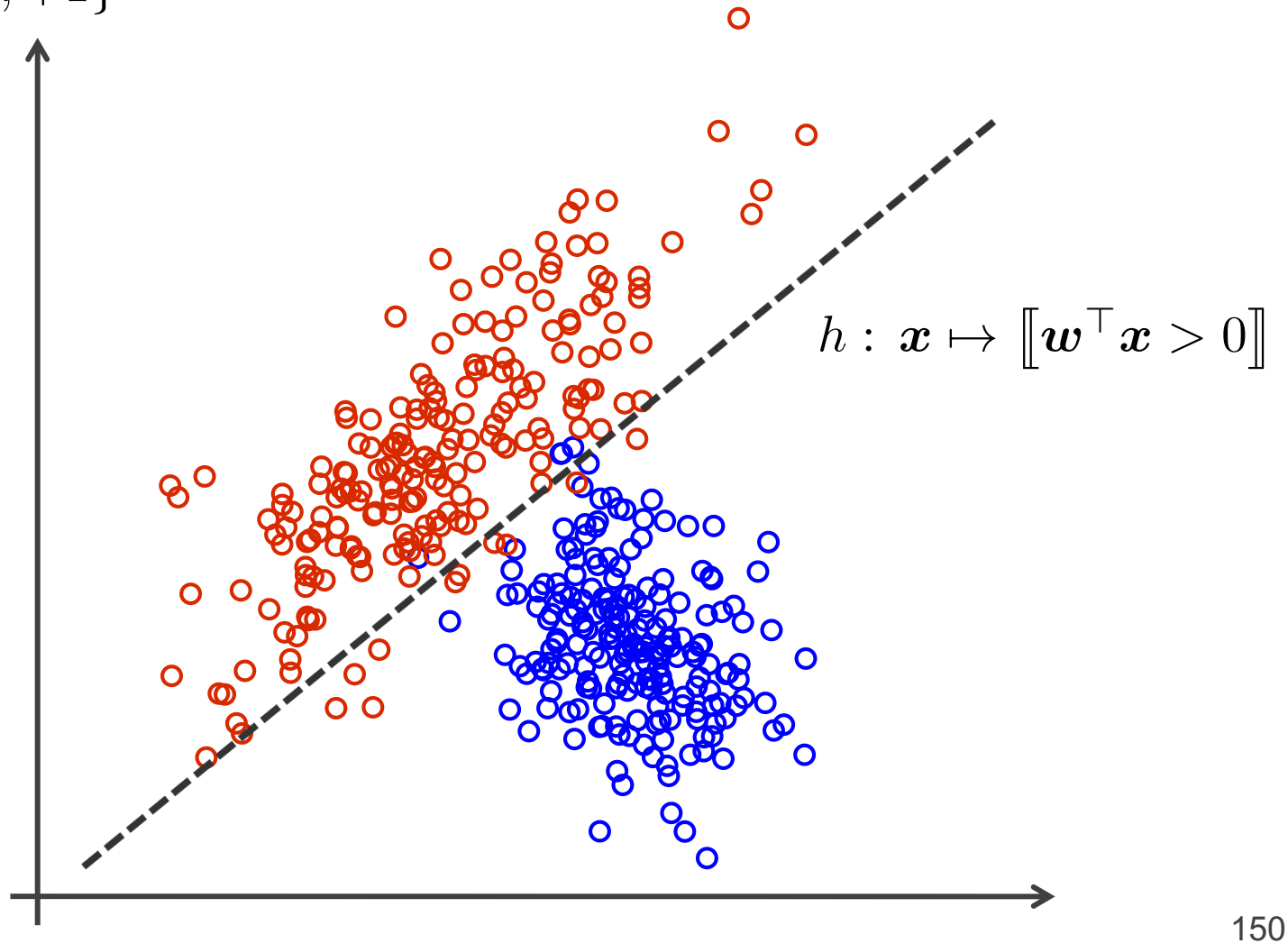
loss function

data generating process

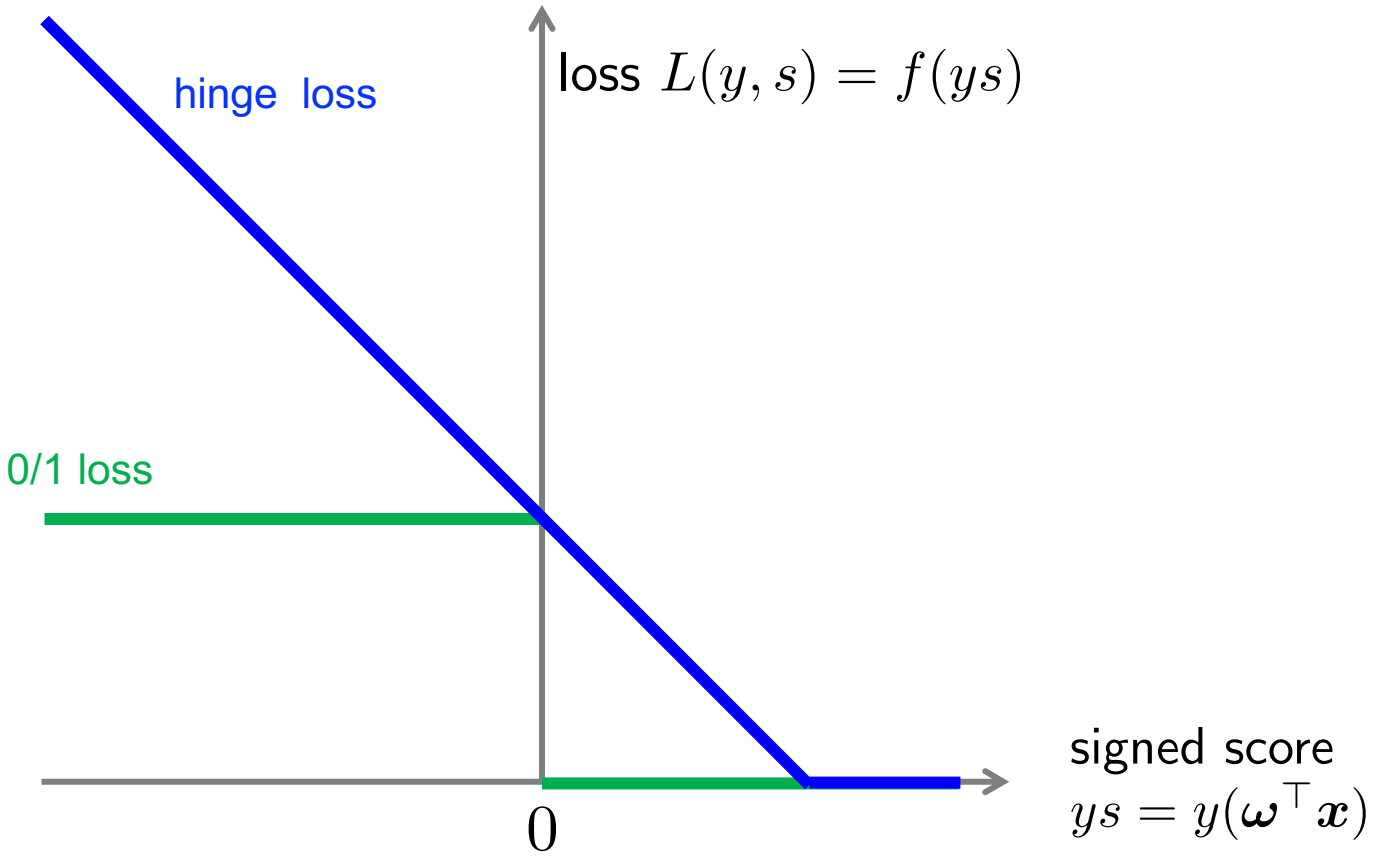
EXAMPLE: BINARY CLASSIFICATION

INTELLIGENT SYSTEMS

$\mathcal{X} = \mathbb{R}^d,$
$\mathcal{Y} = \{-1, +1\}$



EXAMPLE: BINARY CLASSIFICATION

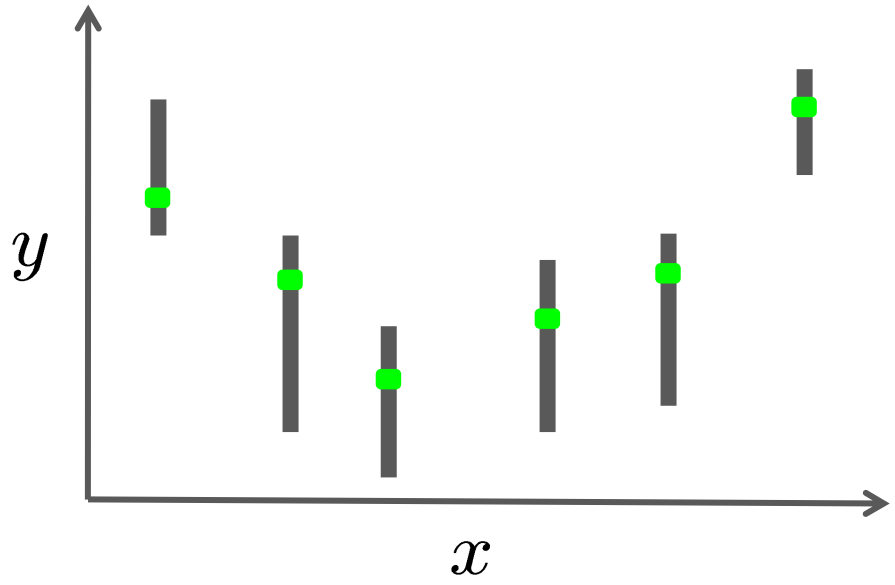


- Set of imprecise/ambiguous/coarse observations

$$\mathcal{O} = \{(\boldsymbol{x}_1, Y_1), \ldots, (\boldsymbol{x}_N, Y_N)\}$$

  with **supersets** $Y_n \ni y_n$.

- An **instantiation** of $\mathcal{O}$, denoted $\mathcal{D}$, is obtained by replacing each Y_n with a candidate $y_n \in Y_n$.

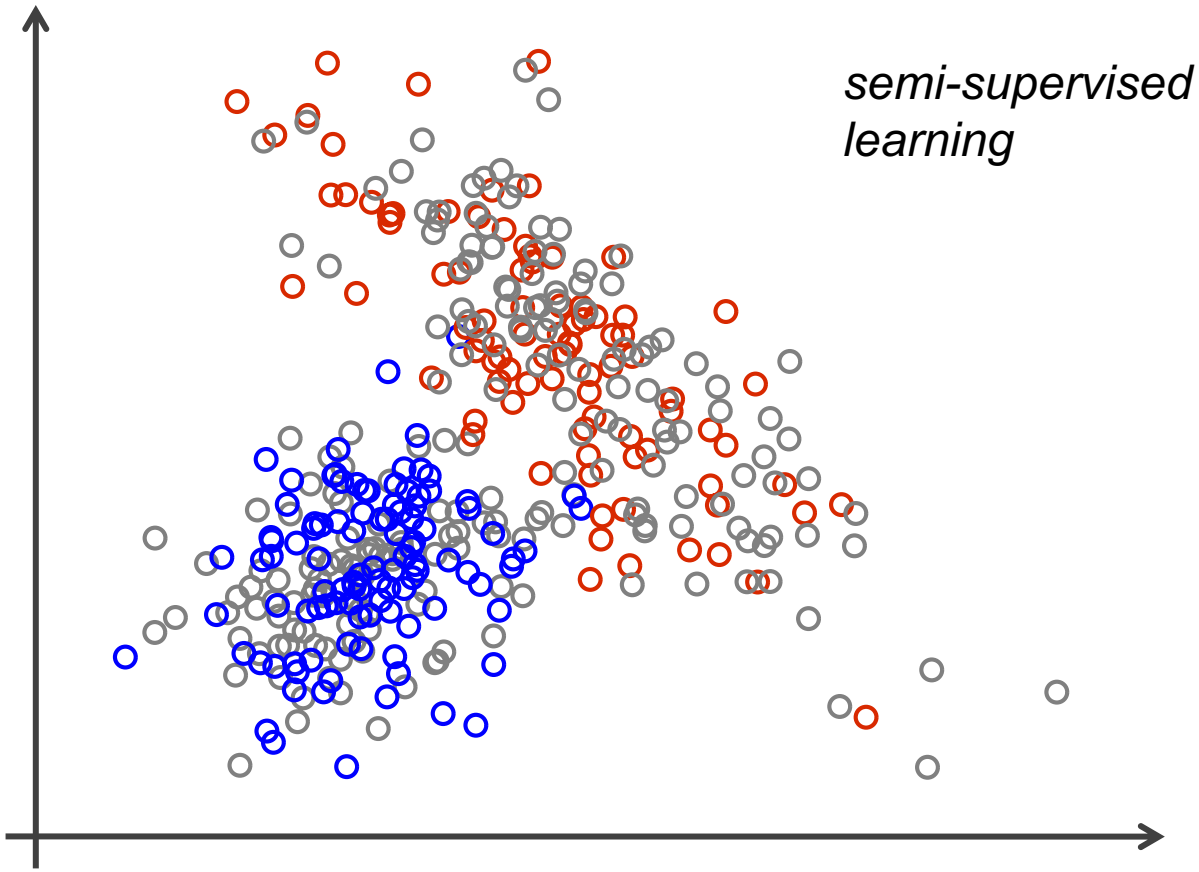


one of infinitely many instantiations

EXAMPLE: BINARY CLASSIFICATION

INTELLIGENT SYSTEMS

= {, }



EXAMPLE: CLASSIFICATION

INTELLIGENT SYSTEMS

x_1	x_2	x_3	y_1	y_2	y_3	y_4
21.9	0	154.3				
43.2	1	133.2				
53.3	1	163.5				
…	…	…	…	…	…	…
42.7	0	142.8				

EXAMPLE: CLASSIFICATION

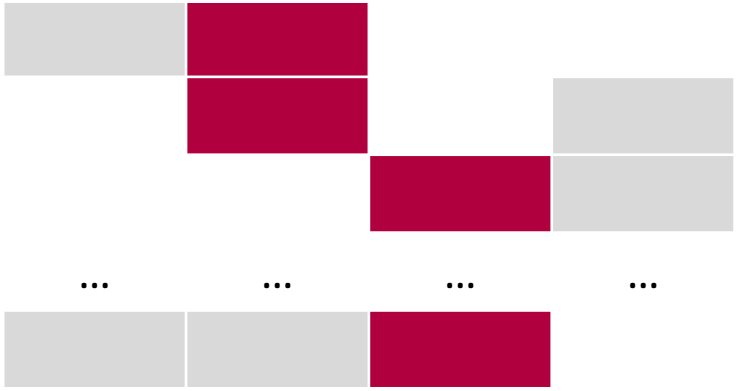
INTELLIGENT SYSTEMS

x_1	x_2	x_3	y_1	y_2	y_3	y_4
21.9	0	154.3				
43.2	1	133.2				
53.3	1	163.5				
…	…	…	…	…	…	…
42.7	0	142.8				

EXAMPLE: CLASSIFICATION

INTELLIGENT SYSTEMS

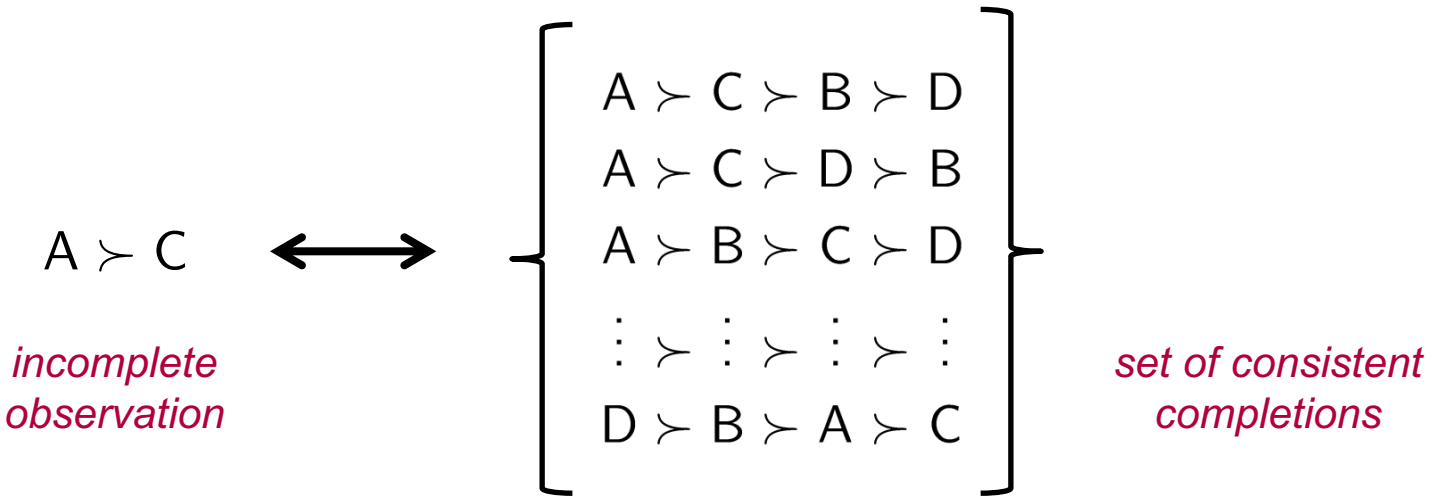
x_1	x_2	x_3	y_1	y_2	y_3	y_4
21.9	0	154.3				
43.2	1	133.2				
53.3	1	163.5				
…	…	…	…	…	…	…
42.7	0	142.8				

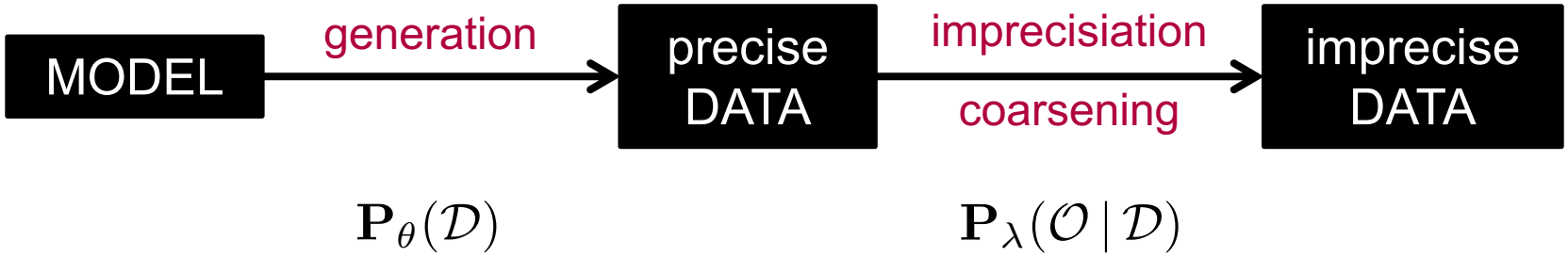


INTELLIGENT SYSTEMS

In label ranking, we learn mappings from instances to rankings:

$$\boldsymbol{x} \quad \mapsto \quad \mathrm{A} \succ \mathrm{C} \succ \mathrm{D} \succ \mathrm{B}$$





- We are interested in learning with **weak assumptions** about the coarsening process, and learning algorithms ought to be **robust** with respect to these assumptions.
- Similar to **epistemic random set setting** (Ω, P, Y), but with little knowledge about multi-valued mapping $Y : \Omega \to 2^{\mathcal{Y}}$.
- **Discriminative learning**, not generative.

EMPIRICAL RISK MINIMIZATION

Given a set of (i.i.d.) training data and a **hypothesis space** $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$, find a model with minimal **empirical risk**

$$\mathcal{R}_{emp}(h) = \frac{1}{N} \sum_{i=1}^{N} L\big(h(\boldsymbol{x}_i), y_i\big) \,.$$

In general, ERM won't work well (unless N is large)…

We propose a principle of **generalized empirical risk minimization** with the empirical risk

$$\mathcal{R}^*_{emp}(h) = \frac{1}{N} \sum_{n=1}^{N} L^*\big(Y_n, h(\boldsymbol{x}_n)\big)$$

and the **optimistic superset loss** (OSL) function

$$L^*(Y, \hat{y}) = \min\big\{L(y, \hat{y}) \,|\, y \in Y\big\}\,.$$

how well the (precise) model fits the imprecise data

We propose a principle of **generalized empirical risk minimization** with the empirical risk

$$\mathcal{R}^{**}_{emp}(h) = \frac{1}{N}\sum_{n=1}^{N} L^{**}\big(Y_n, h(\boldsymbol{x}_n)\big)$$

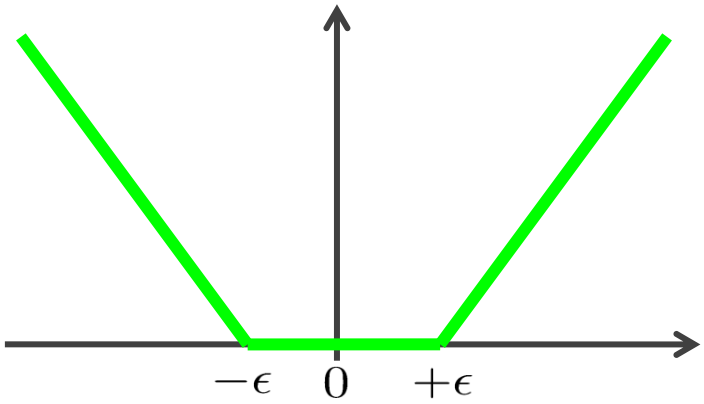
and the **optimistic fuzzy superset loss** (OFSL) function

$$L^{**}(Y, \hat{y}) = \int_0^1 L^*\Big([Y]_\alpha, \hat{y}\Big)\, d\alpha$$

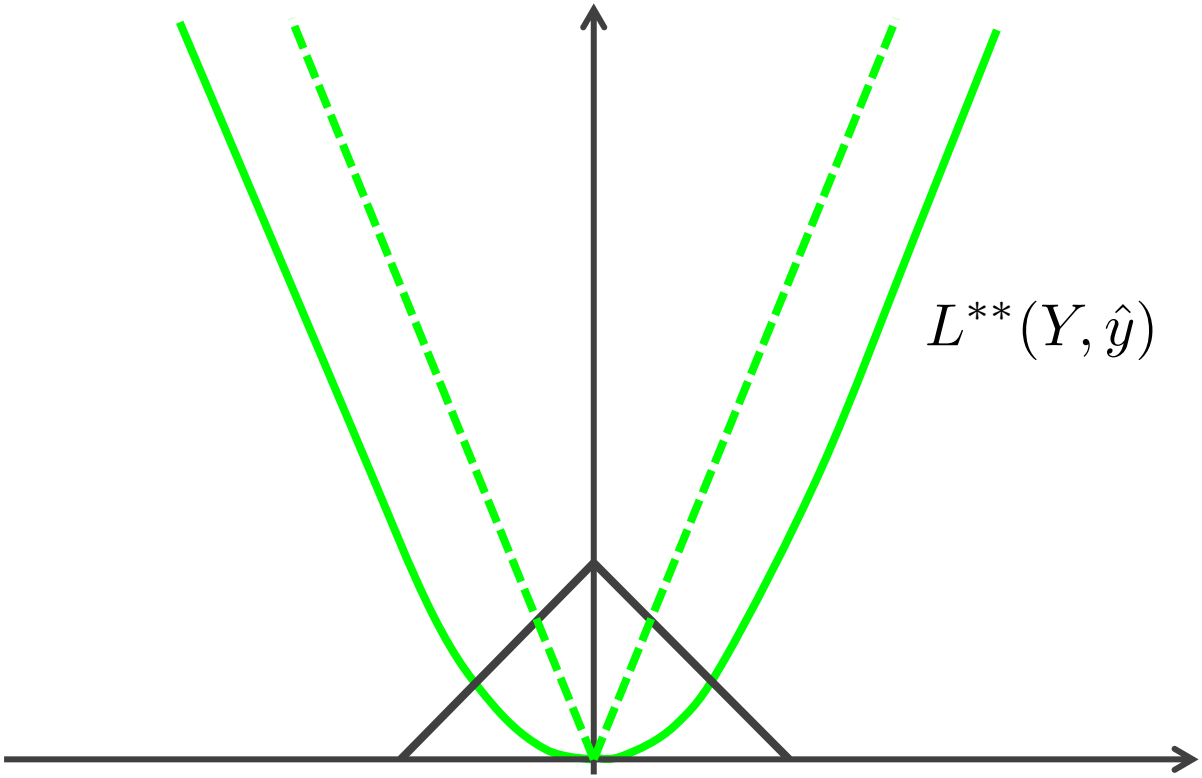
.

- Generalized ERM derives from a likelihood-based approach, which proceeds from $\mathbf{P}(\mathcal{D}, \mathcal{O} \mid h)$,
- and makes (weak) assumptions about the coarsening $\mathbf{P}(\mathcal{O} \mid \mathcal{D}, h)$.
- Further, it exploits additivity of the loss.
- Finally, the logistic loss is replaced by any other loss function.

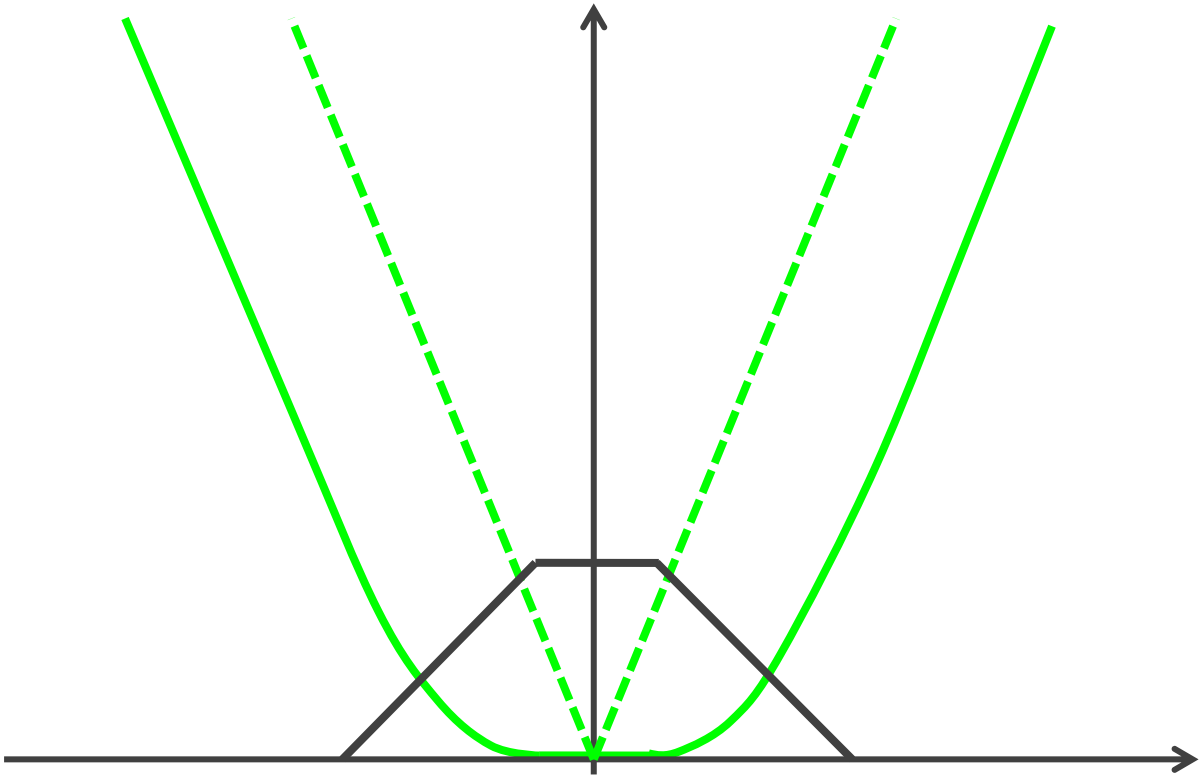
Why should generalized ERM actually work?



The ϵ-insensitive loss $L(y, \hat{y}) = \max(|y - \hat{y}| - \epsilon, 0)$ used in support vector regression corresponds to L^* with L the standard L_1 loss $L(y, \hat{y}) = |y - \hat{y}|$ and precise data y_n being replaced by interval-valued data $Y_n = [y_n - \epsilon, y_n + \epsilon]$.



Huber loss



(generalized) Huber loss

The Kendall loss used in label ranking:

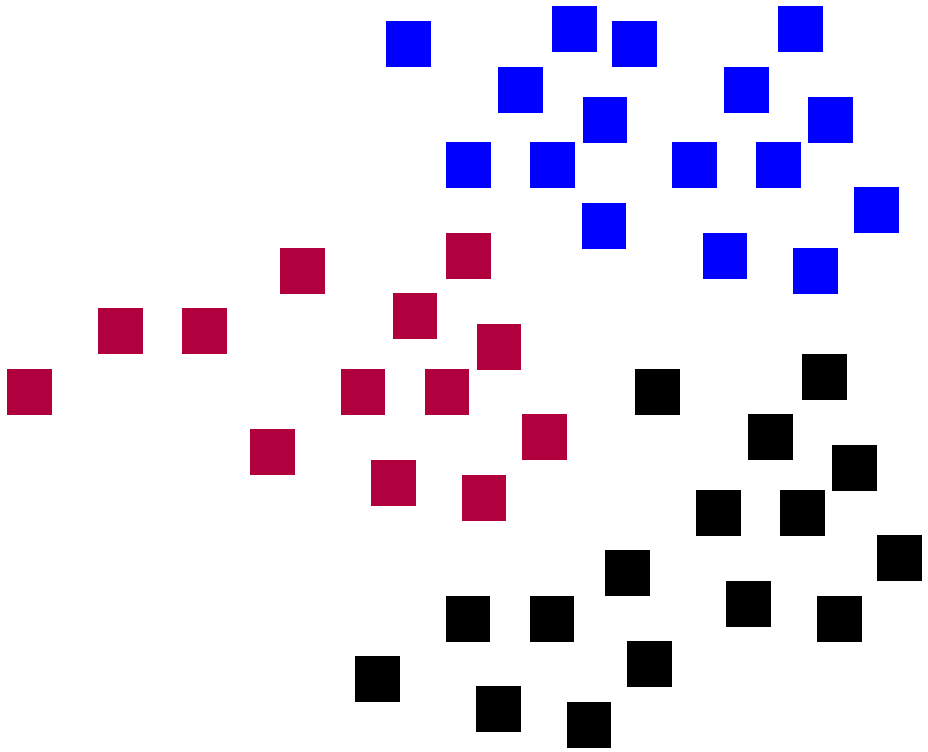
$$L(\pi, \hat{\pi}) = \sum_{i<j} \left[\!\left[\operatorname{sign}(\pi(i) - \pi(j)) \neq \operatorname{sign}(\hat{\pi}(i) - \hat{\pi}(j)) \right]\!\right]$$

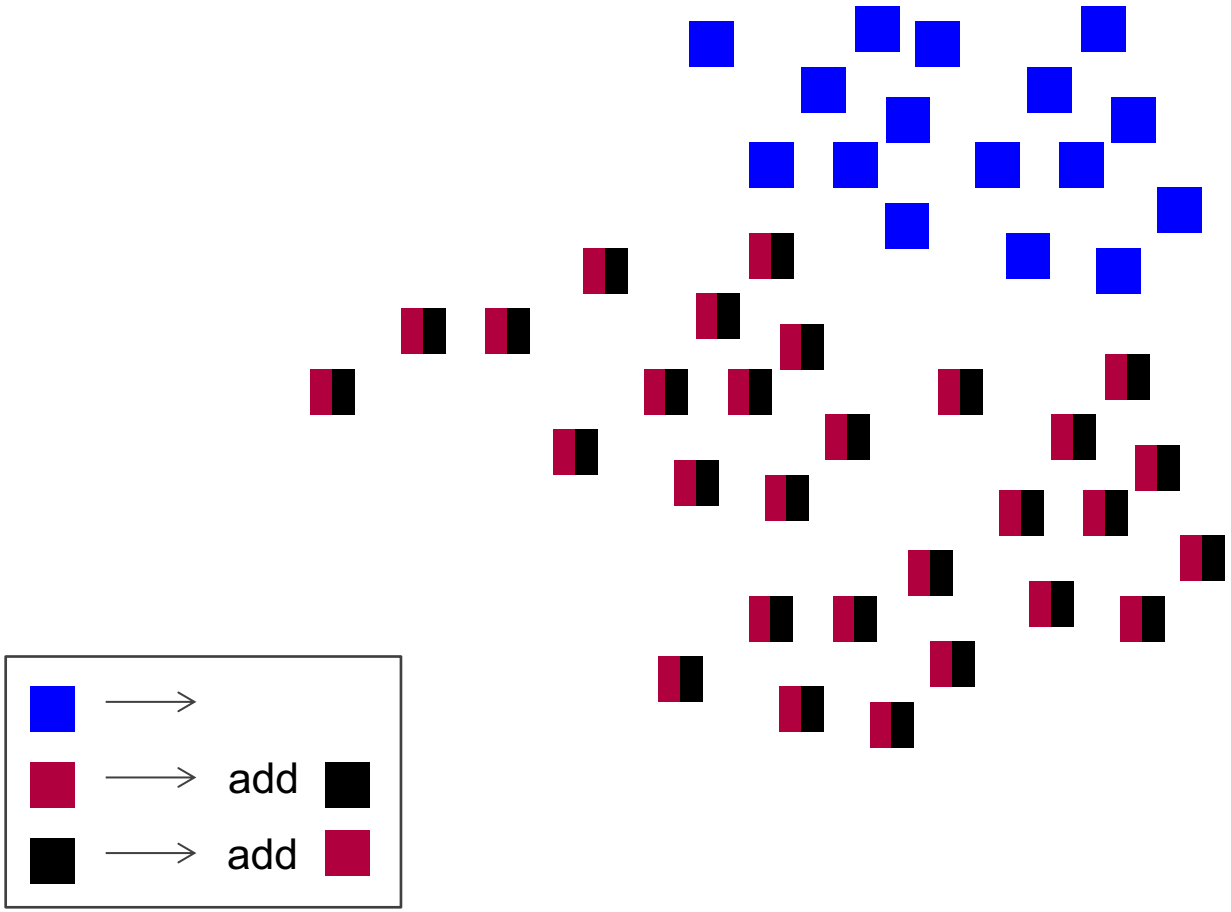
- Cheng and H. (2015) compare an approach to label ranking based on superset learning with state-of-the-art approaches.
- Very strong performance, more robust toward incompleteness.

New methods as natural instantiations of the generalized ERM framework!

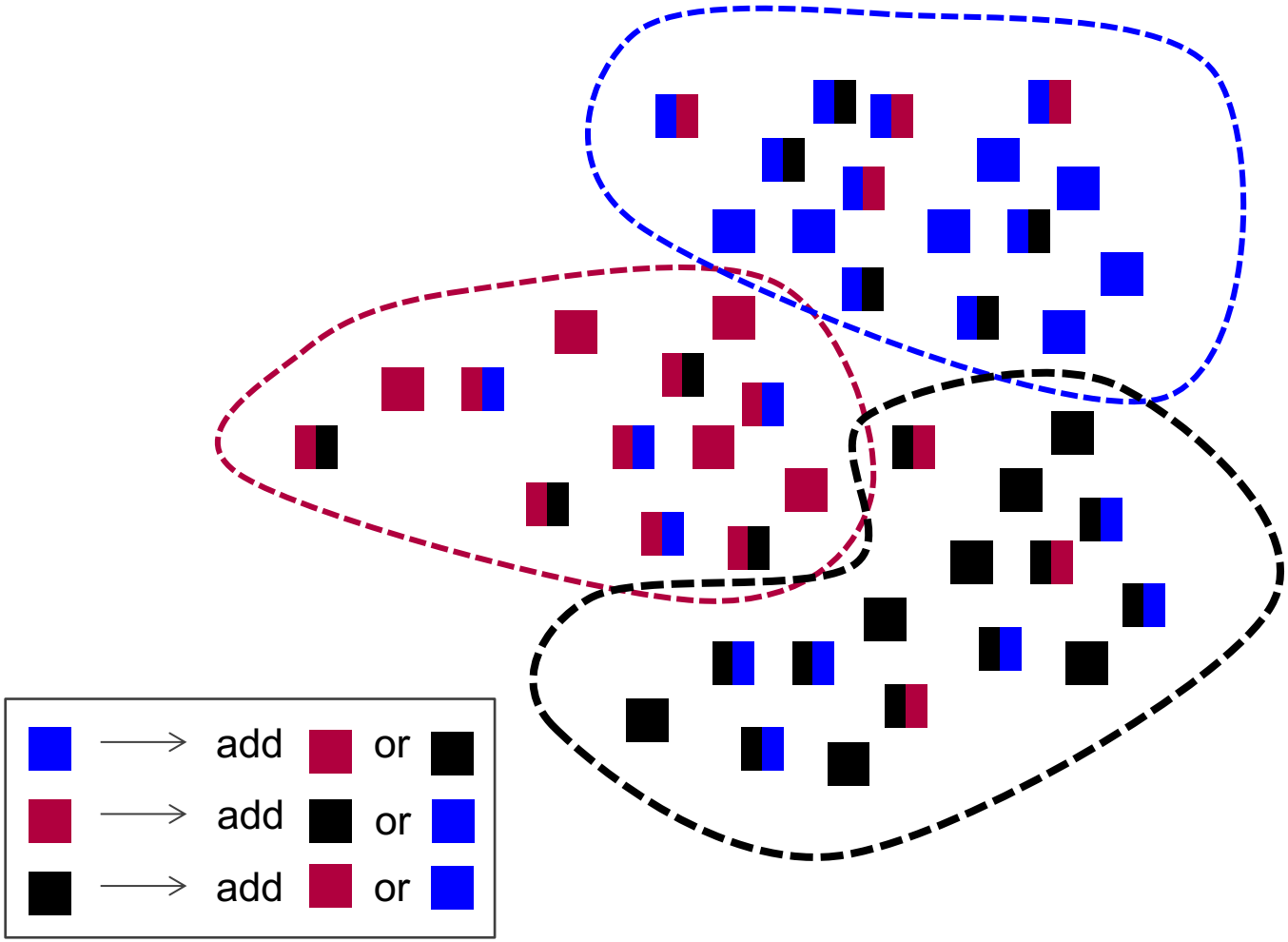
- Under what conditions is (successful) learning in the superset setting actually possible?
- Specifically, under what conditions does generalized ERM work?
- Couldn't the optimism induce a strong bias?
- Might other principles be better?

$$L^*(Y, \hat{y}) = \min \left\{ L(y, \hat{y}) \,|\, y \in Y \right\}$$
$$L^*(Y, \hat{y}) = \text{avg} \left\{ L(y, \hat{y}) \,|\, y \in Y \right\}$$
$$L^*(Y, \hat{y}) = \max \left\{ L(y, \hat{y}) \,|\, y \in Y \right\}$$





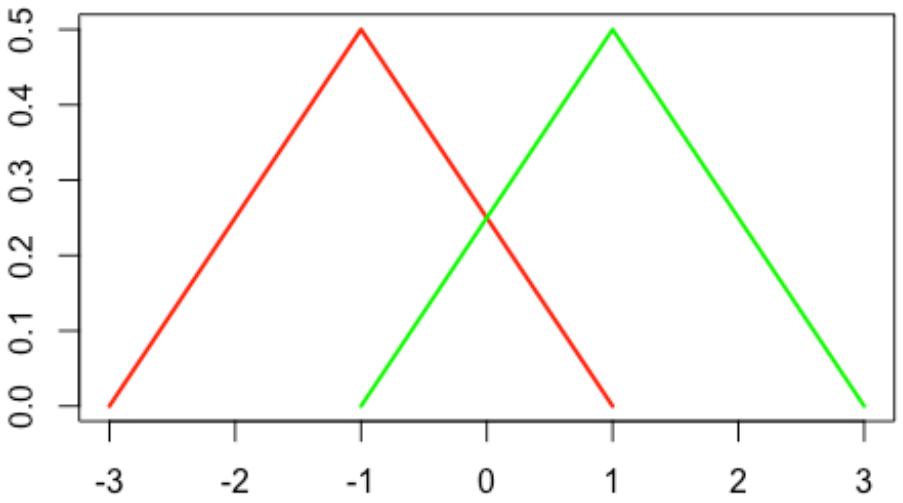
systematic (adversarial) coarsening



non-systematic (random) coarsening

AN EXAMPLE

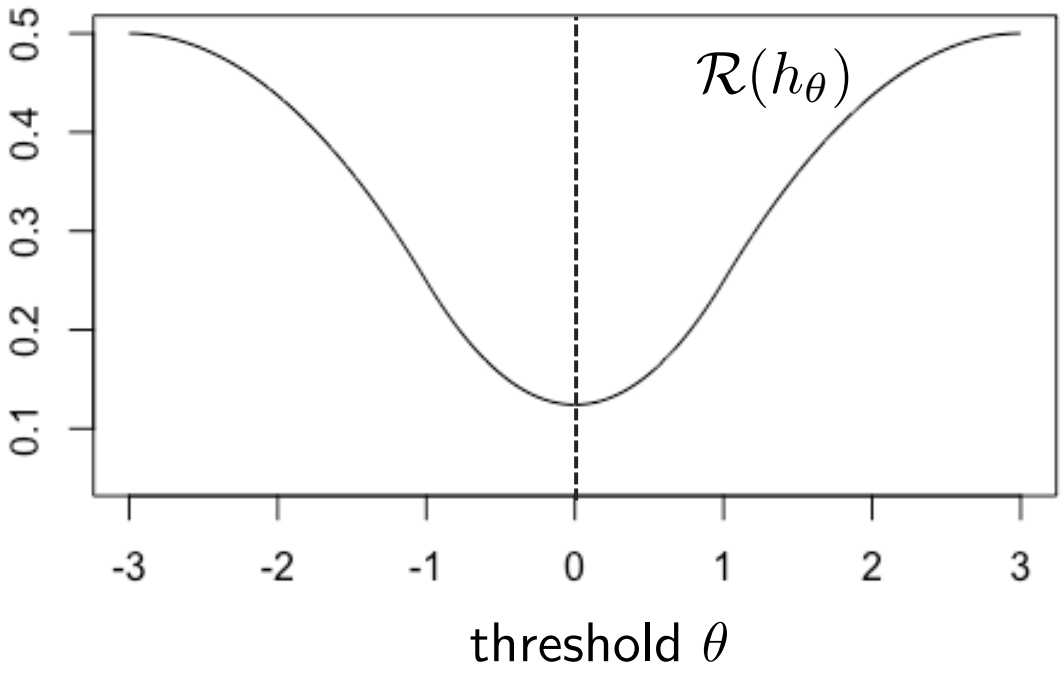
INTELLIGENT SYSTEMS



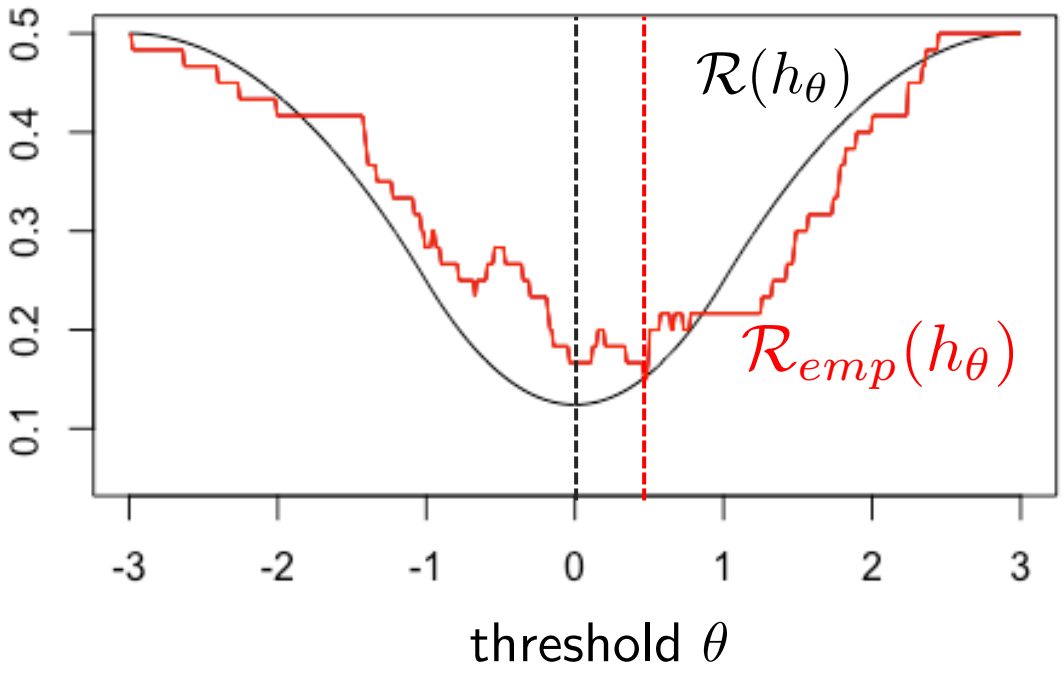
positive class

negative class

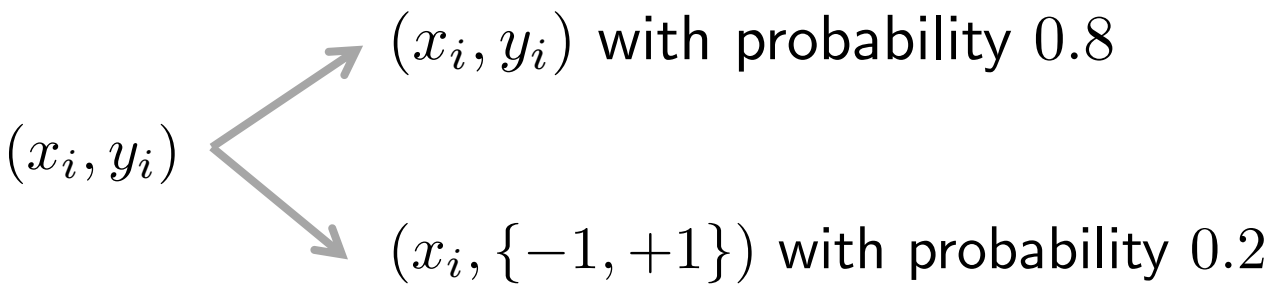
$$h_\theta(x) = \begin{cases} +1\,, & x \geq \theta \\ -1\,, & x < \theta \end{cases}$$


$\mathcal{R}(h_\theta)$
0.1
0.2
0.3
0.4
0.5
-3
-2
-1
0
1
2
3
threshold θ

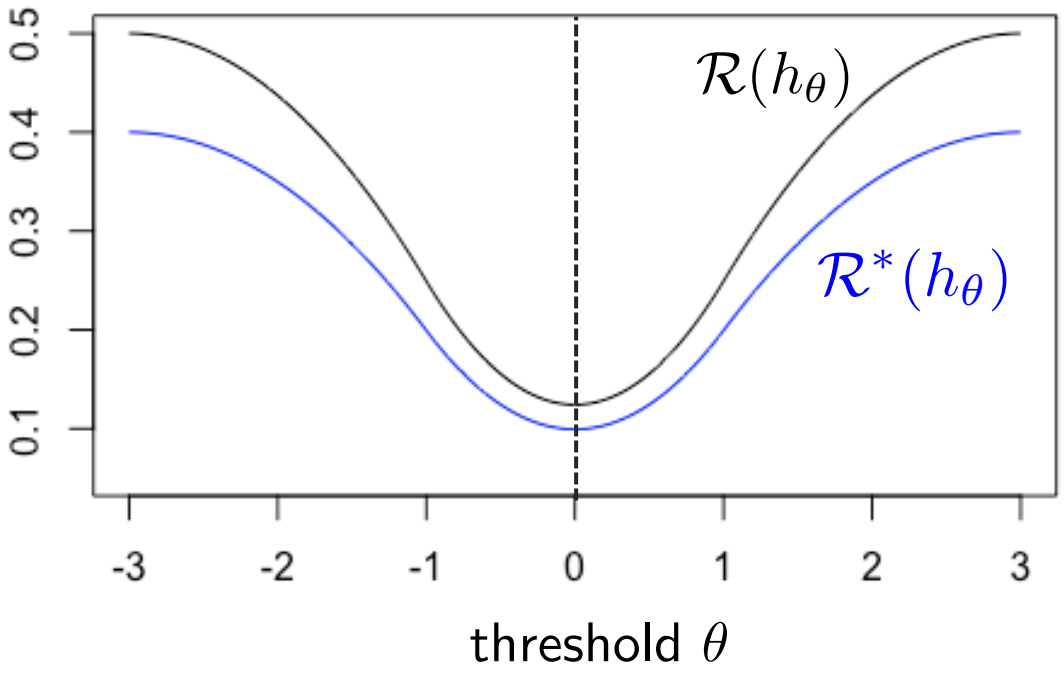
AN EXAMPLE


0.1
0.2
0.3
0.4
0.5
-3
-2
-1
0
1
2
3
$\mathcal{R}(h_\theta)$
$\mathcal{R}_{emp}(h_\theta)$
threshold θ

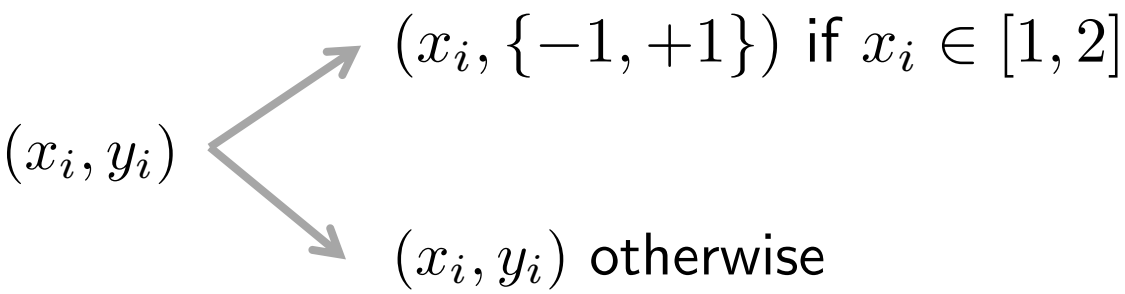
All examples are coarsened with probability 0.2.



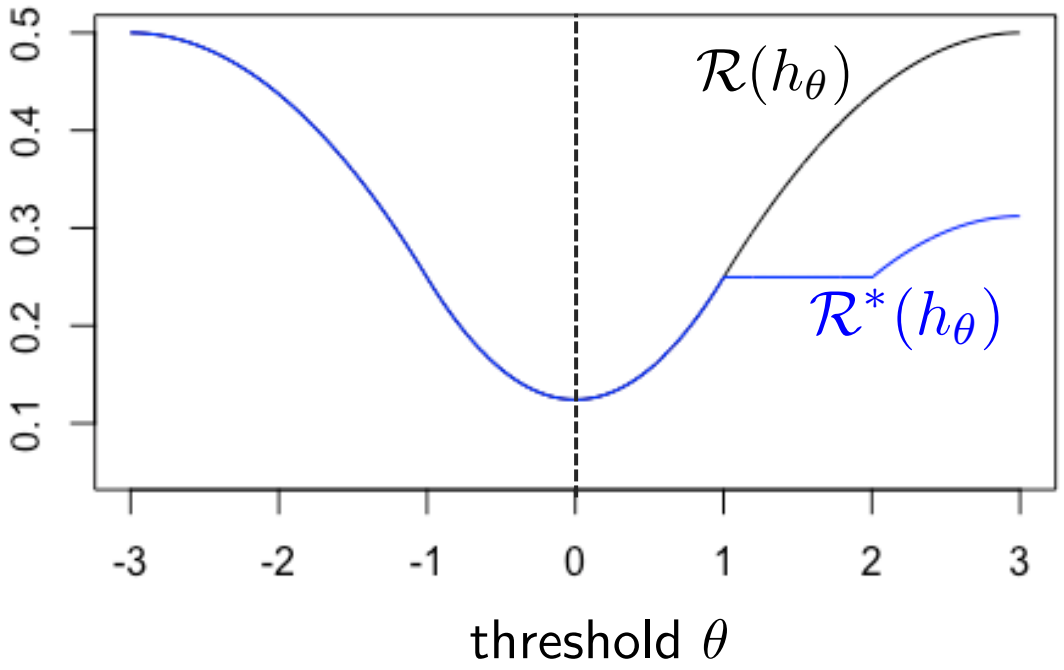
All examples are coarsened with probability 0.2.



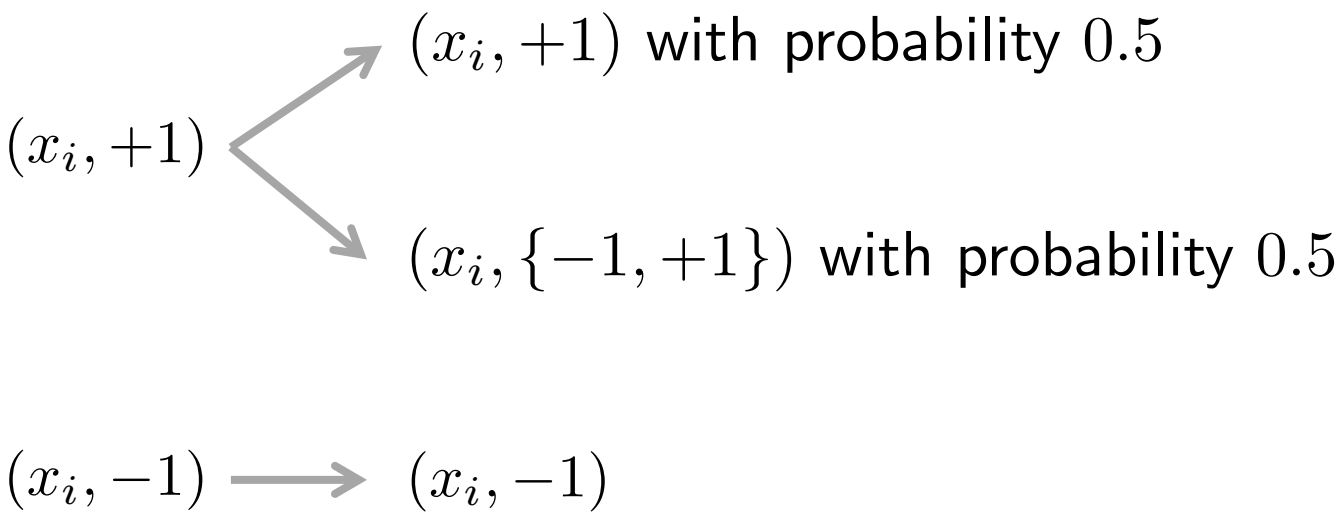
Examples with x between 1 and 2 are coarsened.



Examples with x between 1 and 2 are coarsened.

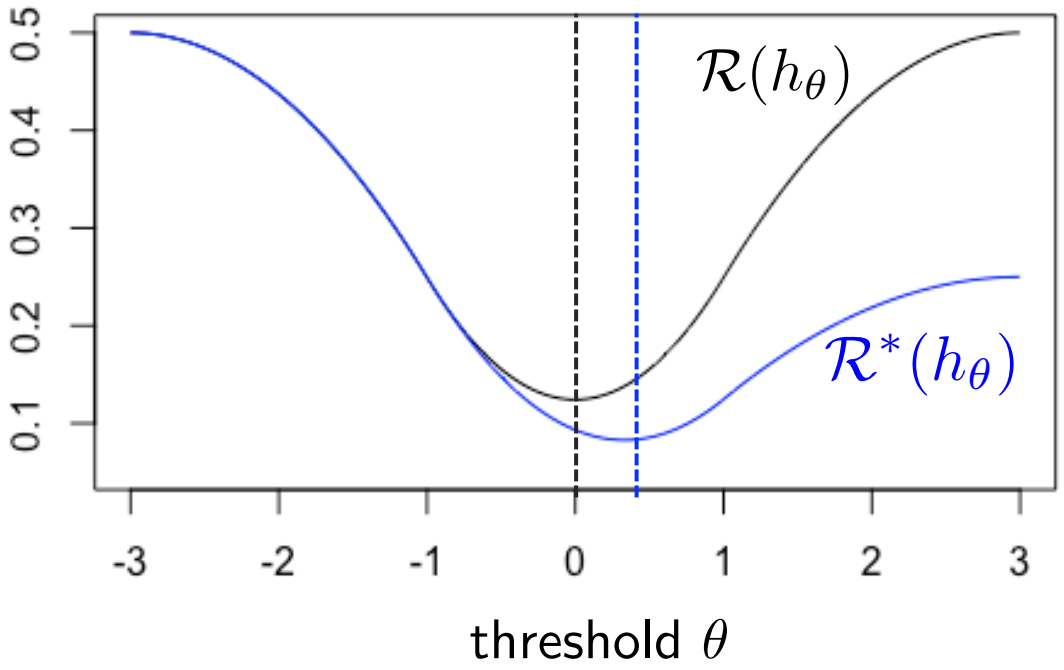


Positive examples are coarsened with probability $1/2$.



INTELLIGENT SYSTEMS

Positive examples are coarsened with probability $1/2$.



The **balanced benefit condition**:

$$0 \leq \eta_1 \leq \inf_{h \in \mathcal{H}} \frac{\mathcal{R}^*(h)}{\mathcal{R}(h)} \leq \sup_{h \in \mathcal{H}} \frac{\mathcal{R}^*(h)}{\mathcal{R}(h)} \leq \eta_2 \leq 1 \ ,$$

where $\mathcal{R}^*(h)$ is the expected superset loss of h.

For sufficiently large sample size,

$$\mathcal{R}(\hat{h}) \leq \mathcal{R}(h^*) + \Delta(d_{\mathcal{H}}, \epsilon, \delta, \eta_1, \eta_2) \ ,$$

with probability $1 - \delta$, where $d_{\mathcal{H}}$ is the Natarajan dimension of $\mathcal{H}$, h^* the Bayes predictor and $\hat{h}$ the minimizer of $\mathcal{R}^*_{emp}$.

Liu and Dietterich (2014) consider the **ambiguity degree**, which is defined as the largest probability that a particular **distractor** label co-occurs with the true label in multi-class classification:

$$\gamma = \sup \left\{ \mathbf{P}_{Y \sim \mathcal{D}^s(\boldsymbol{x},y)}(\ell \in Y) \,|\, (\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}, \ell \in \mathcal{Y}, p(\boldsymbol{x}, y) > 0, \ell \neq y \right\}$$

Let $\theta = \log(2/(1+\gamma))$ and $d_{\mathcal{H}}$ the Natarajan dimension of $\mathcal{H}$. Define

$$n_0(\mathcal{H}, \epsilon, \delta) = \frac{4}{\theta\epsilon}\left(d_{\mathcal{H}}\left(\log(4d_{\mathcal{H}} + 2\log L + \log\left(\frac{1}{\theta\epsilon}\right)\right) + \log\left(\frac{1}{\delta}\right) + 1\right).$$

Then, in the realizable case, with probability at least $1-\delta$, the model with the smallest **empirical superset loss** on a set of training data of size $n > n_0(\mathcal{H}, \epsilon, \delta)$ has a **generalisation error** of at most ϵ.

So far: Imprecision as a necessary evil

Observations are imprecise/incomplete, and we have to deal with that!

Now: Imprecision as a means for modeling

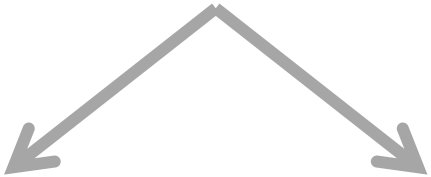
Deliberately turn precise into imprecise data, so as to modulate the influence of an observation on the learning process!

Motivated by the following monotonicity property:

$$Y \subset Y' \quad \Rightarrow \quad L^*(Y, \cdot) \geq L^*(Y', \cdot)$$

We suggest an alternative way of **weighing examples**, namely, via **„data imprecisiation“** ...

$$L(y_i, \cdot)$$



$$w_i \, L(y_i, \cdot) \qquad\qquad L^*(Y_i, \cdot)$$

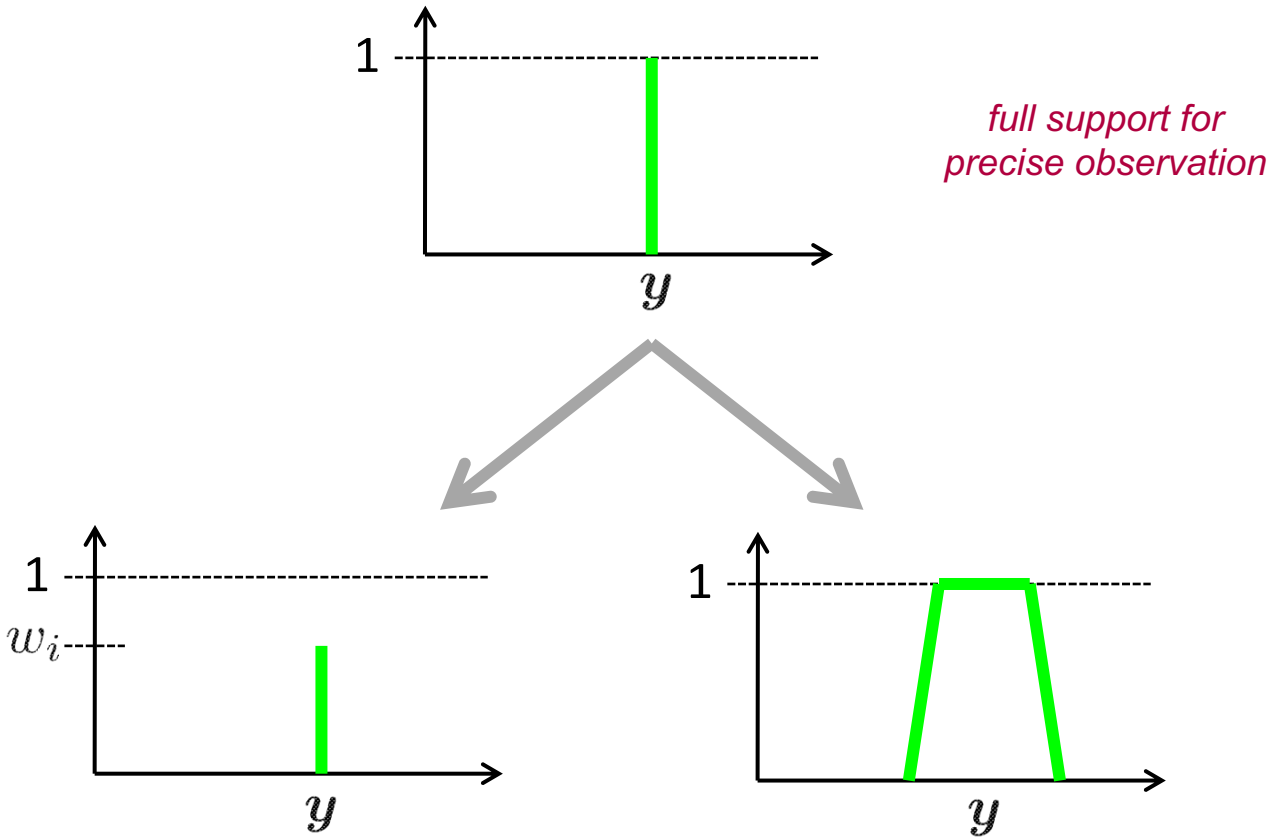
modulating the influence of a training example $(\boldsymbol{x}_i, y_i)$ by multiplying the loss with a constant w_i.

modulating the influence of a training example $(\boldsymbol{x}_i, y_i)$ by coarsening the observation y_i.

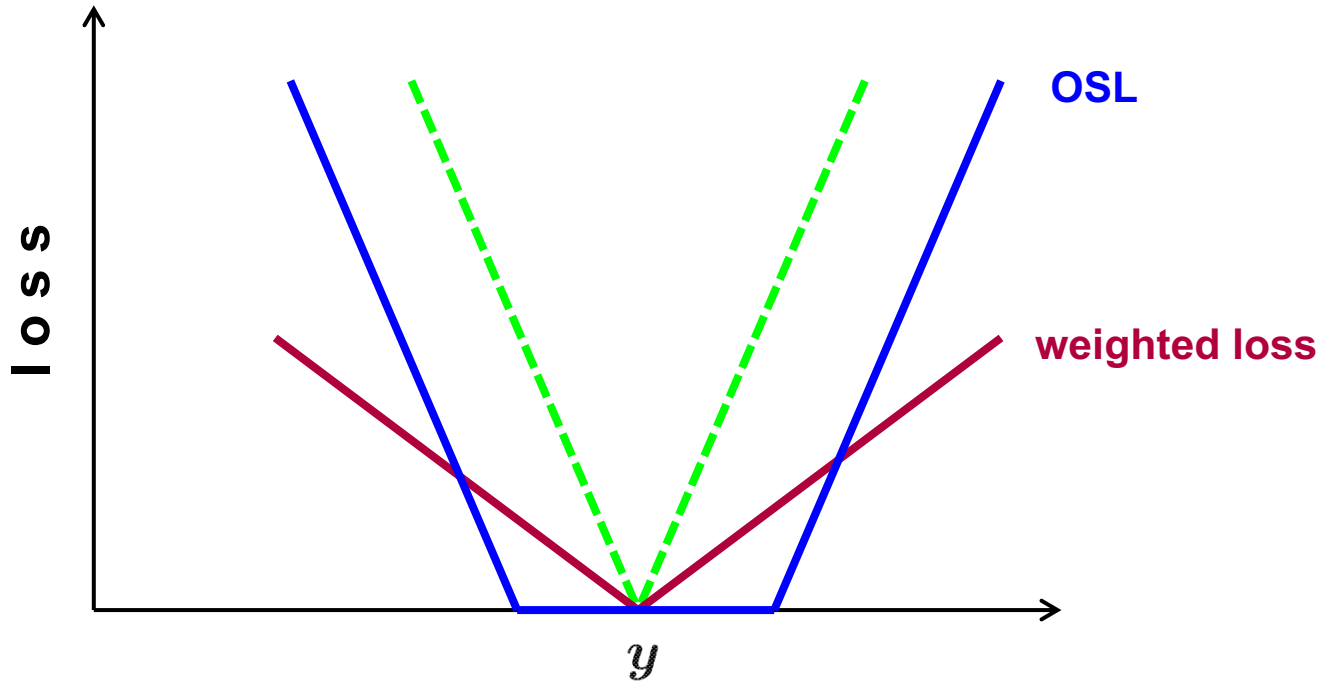
EXAMPLE WEIGHING

INTELLIGENT SYSTEMS

We suggest an alternative way of **weighing examples**, namely, via **„data imprecisiation“** ...



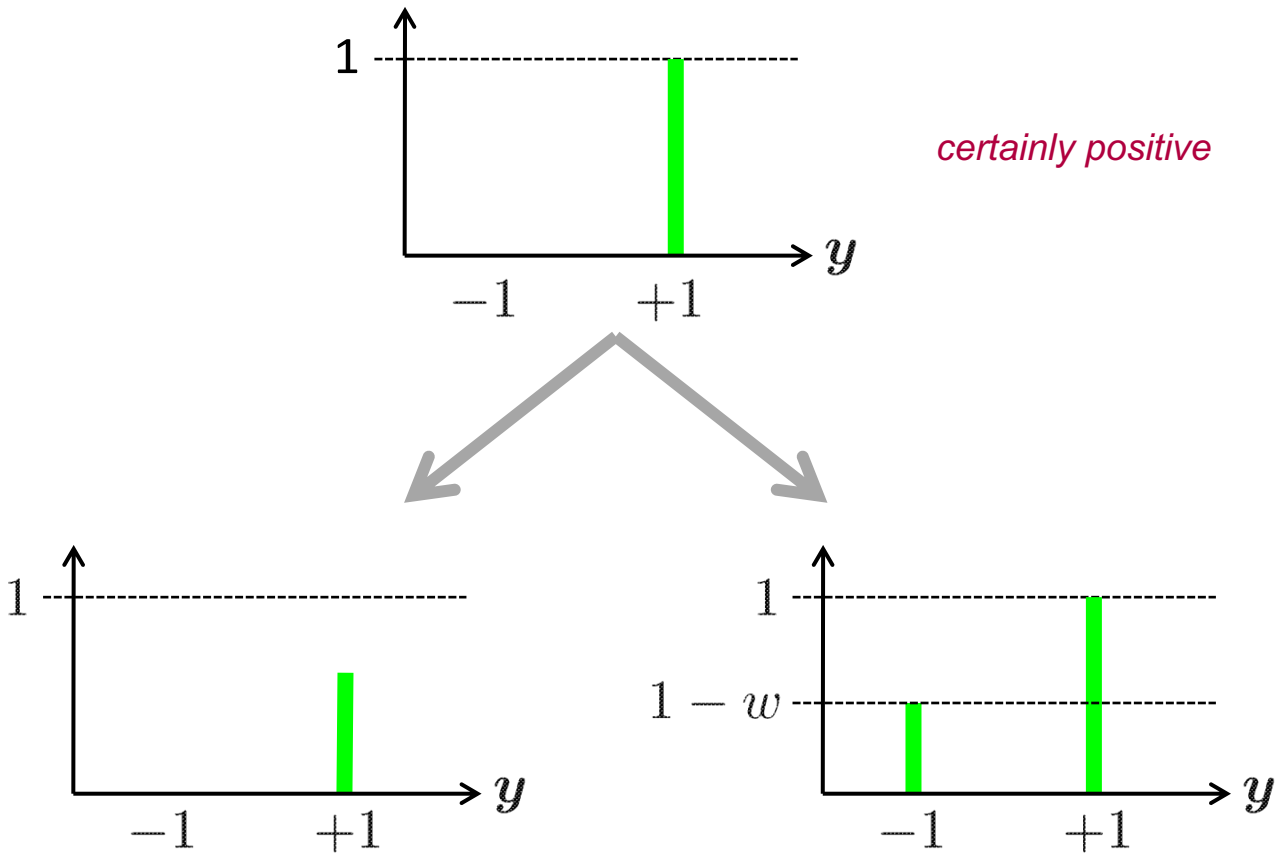
EXAMPLE WEIGHING



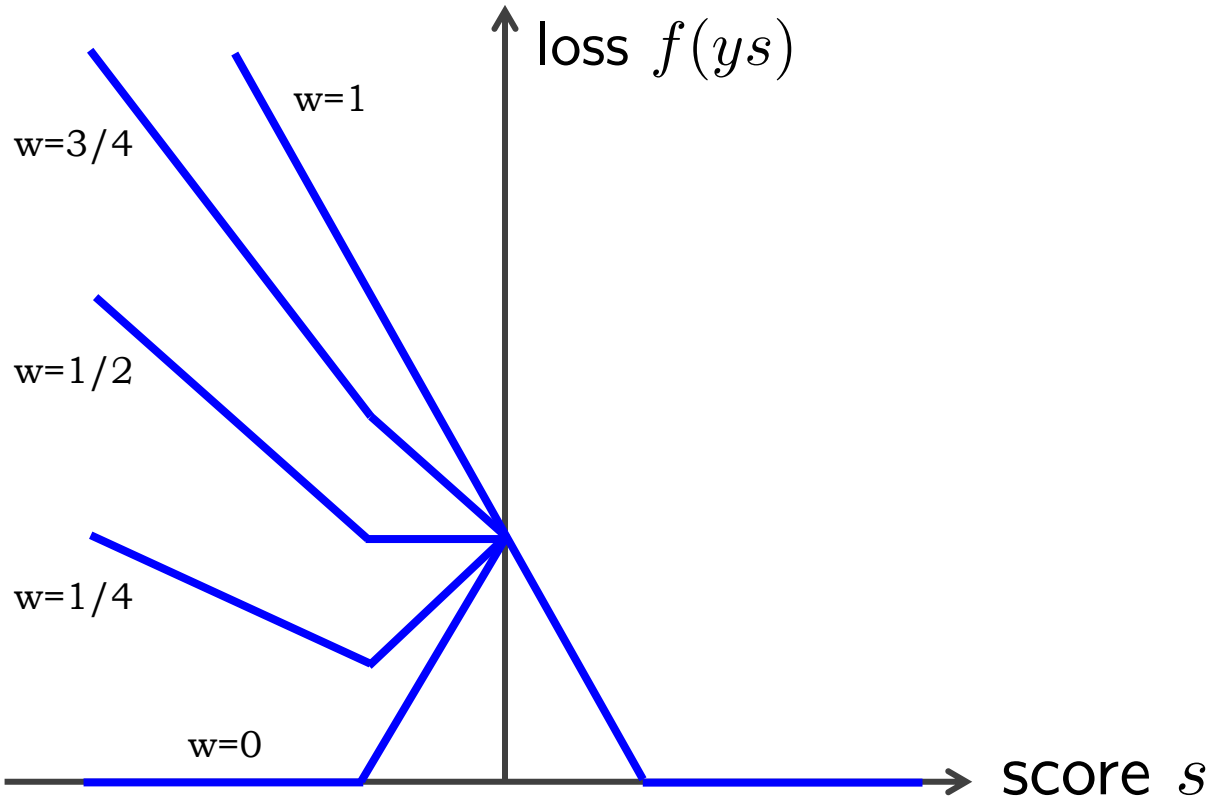
Different ways of (individually) discounting the loss function.

In (Lu and H., 2015), we empirically compared standard **locally weighted linear regression** with this approach and essentially found no difference.

We suggest an alternative way of weighing examples, namely, via **„data impreciſiation“** ...

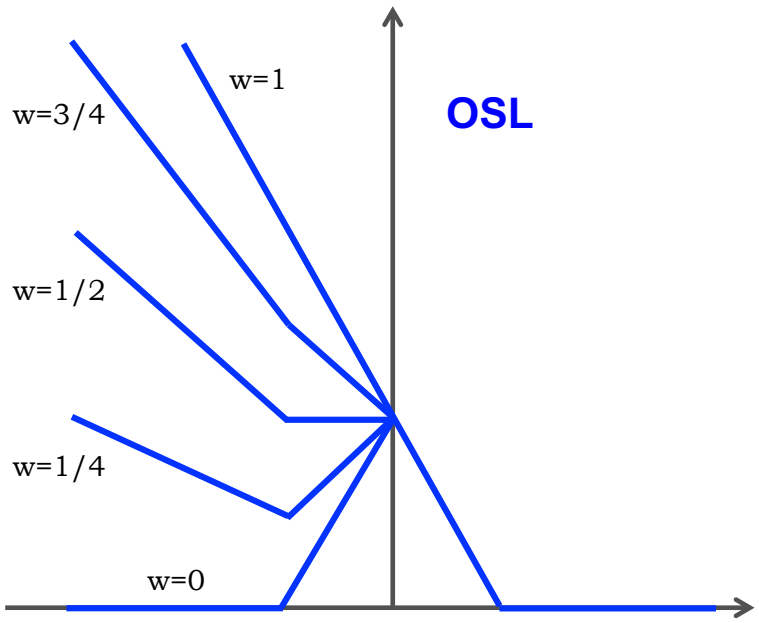


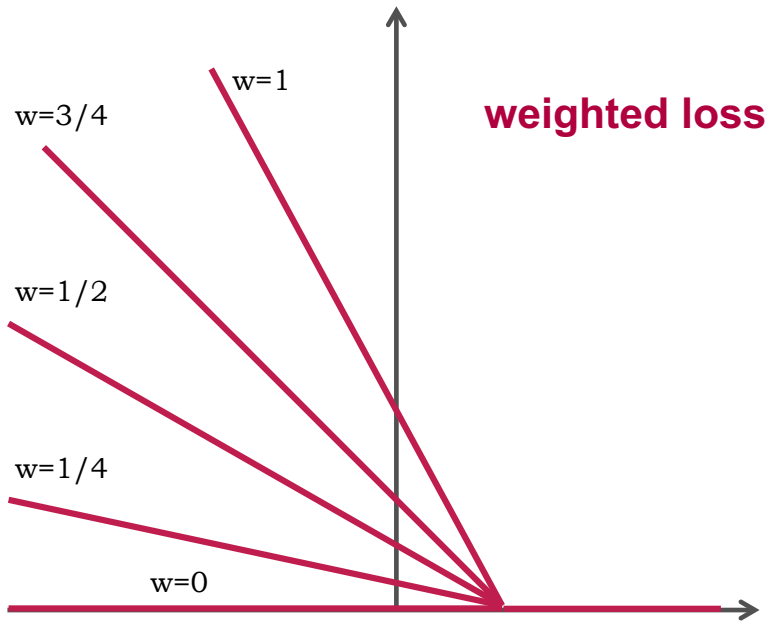
FUZZY MARGIN LOSSES



GENERALIZED HINGE LOSS

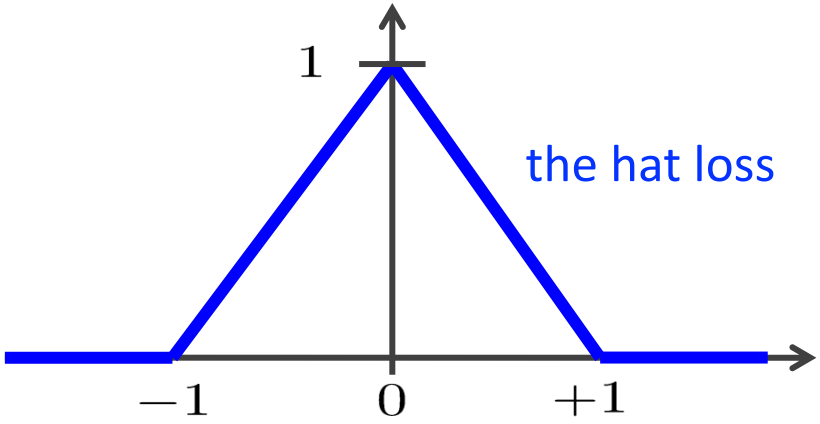
FUZZY MARGIN LOSSES





Different ways of (individually) discounting the loss function.

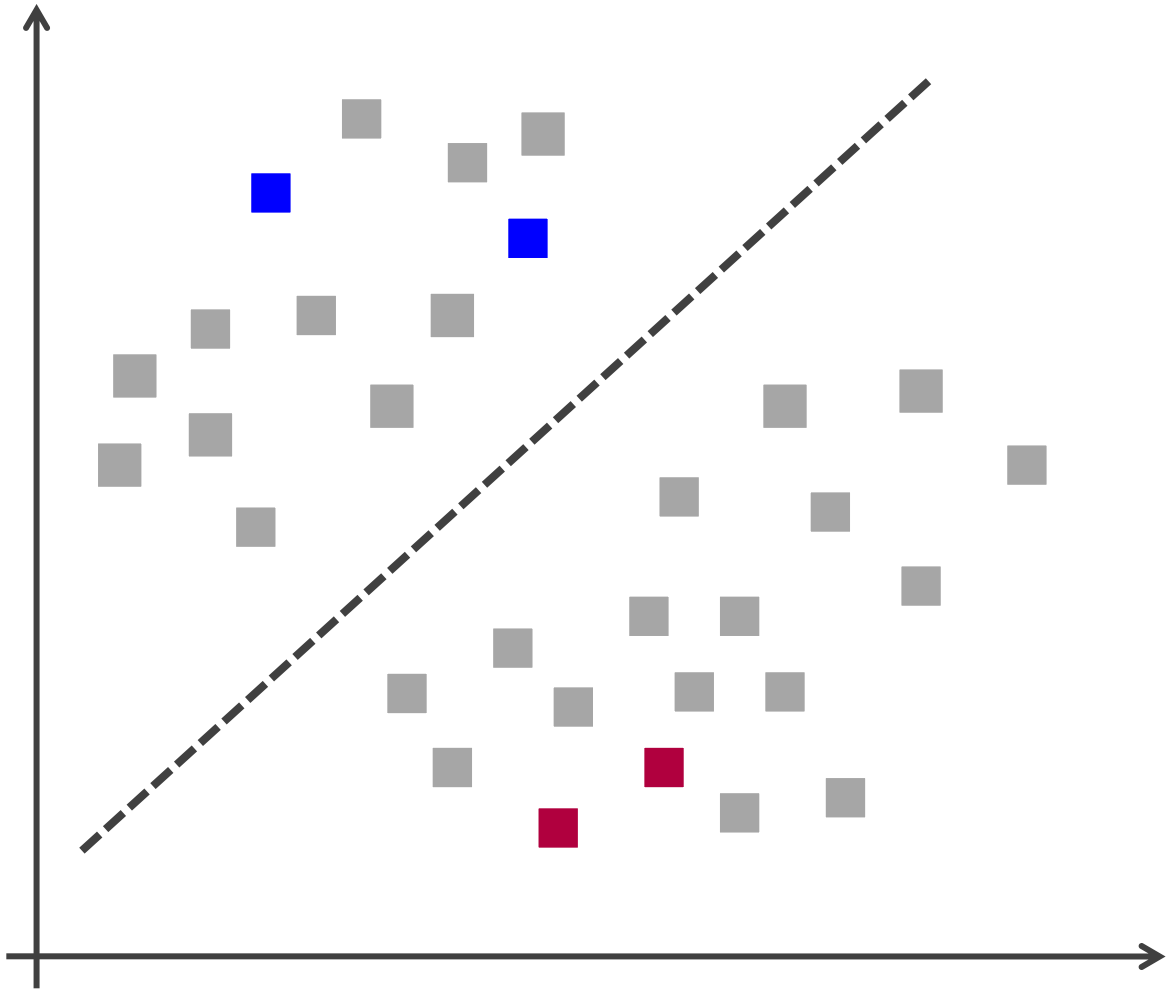
THE HAT LOSS



Semi-supervised learning with SVMs: Consider unlabeled data as instances labeled with the superset $\{-1, +1\}$. The generalized loss L^* with L the standard hinge loss then corresponds to the (non-convex) "hat loss".

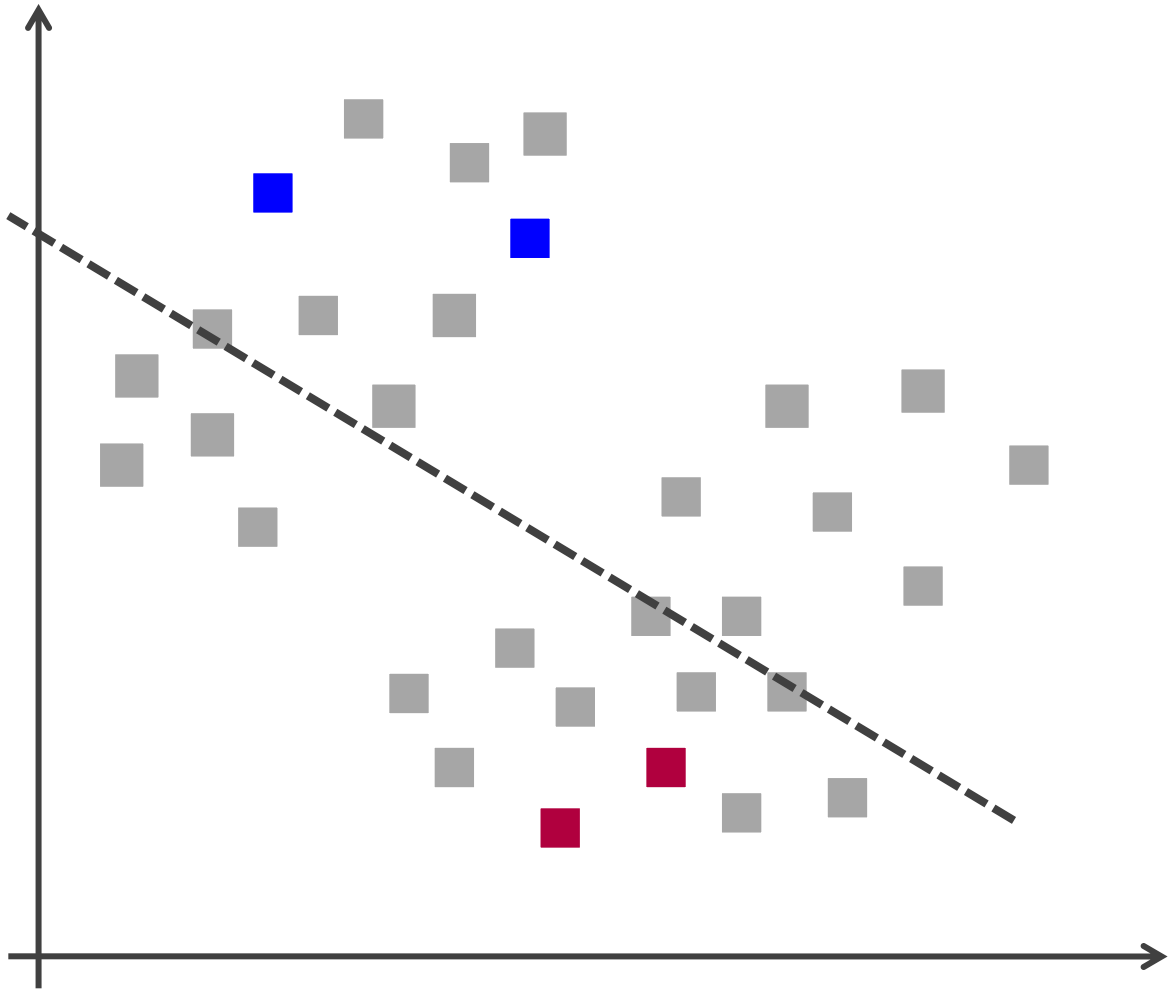
DATA DISAMBIGUATION

INTELLIGENT SYSTEMS



DATA DISAMBIGUATION

INTELLIGENT SYSTEMS



CONCLUSION

- **Machine learning** is a flourishing field, at the core of data science, and is apparently doing well without fuzzy logic.
- Yet, interesting contributions to **fuzzy machine learning** have already been made, and even more significant ones are conceivable.
- Going beyond straightforward fuzzy extensions of conventional ML methods, we need to focus on the right topics, correctly appraise the (complementary) role of fuzzy sets in learning from data, and avoid unwarranted claims.
- As an example of using **fuzzy systems** for modeling, we presented f**uzzy pattern trees** as a novel model class that nicely combines expressivity and transparency.
- In addition to modeling functional dependencies, fuzzy sets can also be used for **modeling data**; we addressed this issue and presented basic ideas of (fuzzy) **superset learning**.

Fuzzy logic and machine learning:

- E.H. Fuzzy Sets in Machine Learning and Data Mining: Status and Prospects. Fuzzy Sets and Systems, 156(3), 2005.
- E.H. Fuzzy Machine Learning and Data Mining. WIREs Data Mining and Knowledge Discovery, 2011.
- E.H. Does machine learning need fuzzy logic? Fuzzy Sets and Systems, 28:292--299, 2015.
- E.H. From knowledge-based to data-driven fuzzy modeling: Development, criticism, and alternative directions, Informatik Spektrum, 38(6):500—509, 2015.

INTELLIGENT SYSTEMS

Fuzzy pattern trees:

- R. Senge and E.H. Fast Fuzzy Pattern Trees Learning for Classification. IEEE TFS, 23(6), 2015.
- Huang, TD. Gedeon, and M. Nikravesh. Pattern trees induction: A new machine learning approach. IEEE TFS 16(4), 2008.
- Y. Yi, T. Fober and E.H. Fuzzy Operator Trees for Modeling Rating Functions. Int. J. Comp. Intell. and Appl. 8(1), 2009.
- R. Senge and E.H. Pattern Trees for Regression and Fuzzy Systems Modeling. Proc. WCCI-2010, Barcelona, Spain, 2010.
- R. Senge and E.H. Top-Down Induction of Fuzzy Pattern Trees. IEEE TFS, 19(2), 2011.
- A. Shaker and E.H. Evolving Fuzzy Pattern Trees for Binary Classification on Data Streams. Information Sciences, 220:34-45, 2013.
- M. Nasiri, E. Hüllermeier, R. Senge and E. Lughofer. Comparing Methods for Knowledge-Driven and Data-Driven Fuzzy Modeling: A Case Study in Textile Industry. Proc. IFSA-2011.

Learning from fuzzy data:

- E. Hüllermeier (2014). Learning from Imprecise and Fuzzy Observations: Data Disambiguation through Generalized Loss Minimization. International Journal of Approximate Reasoning, 55(7):1519-1534, 2014.
- .E. Hüllermeier and W. Cheng (2015). Superset Learning Based on Generalized Loss Minimization. Proc. ECML/PKDD 2015.
- S. Lu and E. Hüllermeier. Locally Weighted Regression through Data Imprecisiation. Workshop Computational Intelligence, Dortmund, 2015.
- S. Lu and E. Hüllermeier. Support Vector Classification on Noisy Data using Fuzzy Superset Losses. Workshop Computational Intelligence, Dortmund, 2016.