Understanding Machine Learning – A theory Perspective

Shai Ben-David

University of Waterloo

MLSS at Okinawa, Japan, 2024

Disclaimer – Warning ….

This talk is NOT about how amazing machine learning is.

I am sure you are already convinced of that.

I am NOT going to show any videos of fancy applications of ML.

I will talk more about the "WHY" of ML than about the "HOWTO".

I wish to focus on understanding the principles underlying Machine Learning.

High level view of (Statistical) Machine Learning

"The purpose of science is

to find meaningful simplicity

in the midst of

disorderly complexity"

Herbert Simon

More concretely

- Statistical learning is concerned with algorithms that detect ***meaningful regularities*** in large complex data sets.
- We focus on data that is ***too complex*** for humans to figure out its meaningful regularities.
- We consider the task of finding such regularities from ***random samples*** of the data population.

How is learning handled in nature (1)?

Bait Shyness in rats

Successful animal learning

The *Bait Shyness* phenomena in rats:

Poisoned baits are not effective against rats.

When rats encounter poisoned food,
they learn very fast the causal relationship
*between the **taste and smell** of the food*
*and **sickness** that follows a few hours later.*

How is learning handled in nature (2)?Pigeon Superstition (Skinner 1948)

What is the source of difference?

- In what way are the rats “smarter” than the pigeons?

Bait shyness and inductive bias

Garcia et al (1989) *:*

Replace the stimulus associated with the poisonous baits by making a sound when they taste it (rather than having a slightly different taste or smell).

How well do the rats pick the relation of sickness to bait in this experiment?

Surprisingly (?)

The rats fail to detect the association!

They do not refrain from eating when the same warning sound occurs again.

What about "improved rats"?

- Why aren't there rats that will also pay attention to the noise when they are eating?
- And to light, and temperature, time-of-day, and so on?
- Wouldn't such "improved rats" survive better?

Second thoughts about our improved rats

- But then every tasting of food will be an "outlier" in some respect….
- How will they know which tasting should be blamed for the sickness?

The crucial component –

- The rats are “genetically engineered” to pay attention to the taste and smell of food but not to the noise they hear while eating.

- *Leaners need of some* **prior knowledge** *about the task they are trying to solve.*

The Basic No Free Lunch principle

No learning algorithm can be guaranteed to succeed on *all learnable* tasks.

Any learning algorithm has a limited scope of phenomena that it can capture, (an inherent *inductive bias*).

No learning is possible without **applying prior knowledge.**

There can be no *best learner.*

The No Free Lunch theorem

For every learning algorithm,

there exists a perfectly predictable labelling rule such that,

if training data is labelled according to that rule,

the expected *error of the algorithm (having access to examples labeled by that rule*)

is as bad as that of a random coin toss.

Naive user view of machine learning

"I'll give you my data, you'll crank up your machine and return meaningful insight"

"It does not work?"

"I can give you more data"

"Still doesn't work?"

" I'll try another advisor" ....

The missing component

The necessity of domain-specific prior knowledge.

The modeling of prior knowledge

A central challenge for machine learning:
How should one model prior knowledge?

We need tools that are, at the same time,

- ✓ Understandable to the domain expert

And

- ✓ Useful for the design of learning paradigms.

Common tools for modeling PK (1)

Hypothesis classes – restrict the set of potential predictors.

For example, commit to predicting with a linear decision rule. Namely,

Assign weights to each attribute

(Blood Pressure, BMR, Age, Exercise, etc.)

Predict probability of heart attack as a weighted sum of these attribute values.

Common tools for modeling PK (2)

Apply regularization principles:

- Description length
- Margins
- Sparsity
- Low norm

Common tools for modeling PK (3)

Kernels, or, similarity functions-

Aimed to express prior knowledge regarding how likely are two domain element to have the same label.

Example:

Define a *Facebook kernel* over people by

$$K(p,q)=\frac{2\#(\text{Common Facebook friends})}{(\#(\text{Facebook friends of p})+\#(\text{Facebook friends of q}))}$$

Common tools for modeling PK (4)

- Architecture of a Neural Network
- Prior ``likelihood probability" over the set of possible models (Bayesian Learning).

Some formal discussion

Some typical classification prediction tasks

- Medical Diagnosis (*Patient info → High/Low risk*).
- Sequence-based classifications of proteins.
- Detection of fraudulent use of credit cards.
- Stock market prediction (*today's news → tomorrow's market trend*).

The formal setup (for label prediction tasks)

- Domain set – X
- Labe set - Y (often {0,1})

- Learner's input -

Training sample $S=((x_1,y_1), \ldots(x_m, y_m))$

- Learner's output –

prediction rule h: X ➔ Y

Data generation and measures of success

- An unknown distribution D generates instances (x_1, x_2, …) independently.
- An unknown function f: X ➔ Y labels them.
- The error of a classifier h is the probability (over D) that it will fail, h(x) ≠ f(x)

Empirical Risk Minimization (ERM)

Given a labeled sample

$$S=((x_1,y_1), \ldots (x_m, y_m))$$

and some candidate classifier h,

Define the empirical error of h as

$$L_S(h) = |\{i : h(x_i) \neq f(x_i)\}|/m$$

(the proportion of sample points on which h errs)

ERM – find h the minimizes $L_S(h)$.

Not so simple – Risk of Overfitting

- Given any training sample

  $S=((x_1,y_1), \ldots (x_m, y_m))$

- Let,

  $h(x)=y_i$ if $x=x_i$ for some $i \leq m$

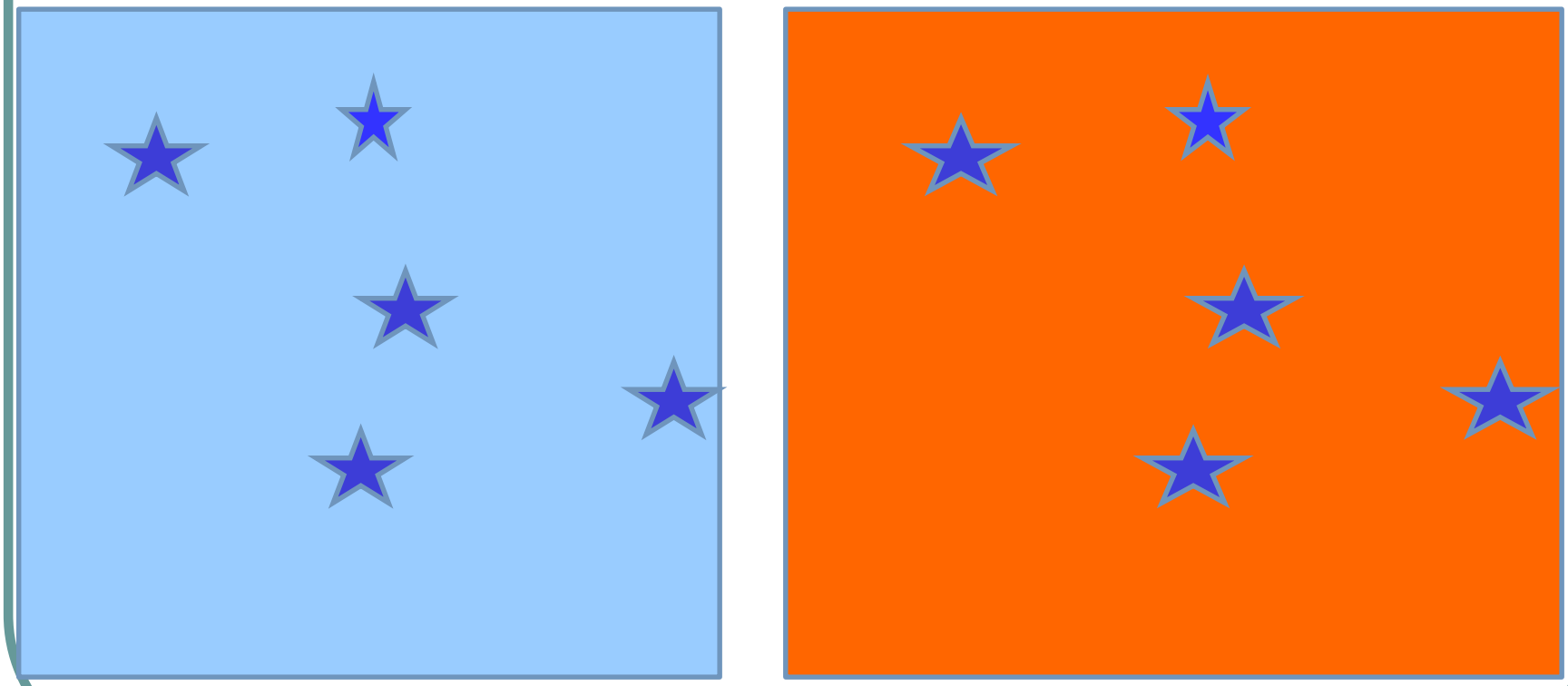
  and $h(x)=0$ for any other x.

- Clearly $L_S(h) =0$.
- It is also pretty obvious that in many cases this h has high error probability.

Failure of ERM

Ground truth f

ERM generated h



First type of prior knowledge – Hypothesis classes

- A *hypothesis class H* is a set of hypothesis. We re-define the ERM rule by searching only inside such a prescribed H.

- *$ERM_H(S)$ picks a classifier* h in H that minimizes the empirical error over members of H

Our first theorem

Theorem: (Guaranteed success for ERM_H)

Let H be a finite class, and assume further that the unknown labeling rule, f, is a member of H.

Then for every $\varepsilon, \delta > 0$, if $m > \log(|H|/\delta)/\varepsilon$,

With probability $> 1 - \delta$ over S of size m samples i.i.d. by D and labeled by f,

Any $ERM_H(S)$ has error below ε.

Not only finite classes

- The same holds for the case that X is the real line and H is the class of all intervals.
- More generally – the guarantee holds for every H of ***finite VC-dimension***

A formal definition of learnability

H is ***PAC Learnable*** if

there is a function $m_H : (0,1)^2 \rightarrow N$

and a learning algorithm A,

such that for every distribution D over X,

every $\varepsilon, \delta > 0$ and every f in H,

for samples S of size $m > m_H(\varepsilon, \delta)$

generated by D and labeled by f,

$$Pr[L_D(A(S)) > \varepsilon] < \delta$$

More realistic setups

Relaxing the realizability assumption.

- We wish to model scenarios in which the learner does not have a priori knowledge of a class to which the true classifier belongs.

- Furthermore, scenarios in which the labels are not fully determined by the instance attributes.

General loss functions

Our learning formalism applies well beyond counting classification errors.

Let Z be any domain set.

and l : H x Z ➔ *R* quantify the loss of a "model" h on an instance z.

Given a probability distribution P over Z

Let $L_P(h) = Ex_{z\sim P}(l\ (h, z))$

Examples of such losses

- The 0-1 classification loss:

  l *(h, (x,y))= 0 if h(x)=y and 1 otherwise.*

- *Regression square loss:*

  l $(h, (x,y)) = (y-h(x))^2$

- *K-means clustering loss:*

  l $(c_1, \ldots c_k), z) = min_i (c_i - z)^2$

confusion matrix loss

Real / predict	Cat	Dog	Horse	Car
Cat	0	0.2	0.4	1
Dog	0.2	0	0,3	1
Horse	0.4	0.3	0	1
Car	1	1	1	0

Agnostic PAC learnability

H is **Agnostic *PAC Learnable*** if

there is a function $m_H : (0,1)^2 \rightarrow N$

and a learning algorithm A,

such that for every distribution P over XxY

and every $\varepsilon, \delta > 0$,

for samples S of size $m > m_H(\varepsilon, \delta)$

generated by P,

$$Pr[L_P(A(S)) < \mathbf{Inf_{[h \text{ in } H]}\, L_P(h)} + \varepsilon] < \delta$$

Absolute vs “Regret” loss

Note the difference in setup:

- Realizable setup:
  - We make assumptions (a model in our class is 100% correct)
  - We get strong results (small loss)
- Agnostic setup:
  - We make no assumptions.
  - We get weaker results (small added loss)

Can such learnability be guaranteed?

Once again, the crucial factor is the VC-dimension of the class H.

The fundamental theorem of statistical learning:

A class H is (agnostic) PAC learnable if and only if its VC-dimension is finite.

General Empirical loss

- For any loss l : H x Z ➔ *R as above and a finite domain subset S, define the empirical loss w.r.t. S=(z_1, …z_m) as* $L_S(h) = \Sigma_i$ l $(h, z_i)/m$.

ERM_H – minimizing the empirical loss over the class H is guaranteed to achieve optimal error bounds.

Success guarantee for ERM_H

- The fundamental theorem also states that for any learnable class H,

  the simple ERM_H paradigm achieves best possible sample size guarantees.

A quantitative version of the fundamental theorem

The number of random labeled samples required to guarantee ε, δ successful learning for a class of predictors H is

$$(VCdim(H) + \log(1/\delta))/\varepsilon^2$$

In terms of loss guarantees:

$$L_P(A(S)) < \mathbf{Inf_{[h \text{ in } H]}}\, \mathbf{L_P(h)} + \mathbf{(}VC(H) + \log(1/\delta)\mathbf{)/}|S|$$

It follows that we need the training sample size to be at least VC(H).

The role of VC-dimension

The bound above is tight in worst case:

Such error bound always holds

But

There may be cases (tasks/data) for which we get smaller error.

The VCdim of some hypotheses classes

As a rule of thumb – VC(H) is the number of parameters needed to `zoom in' on a specific h in H:

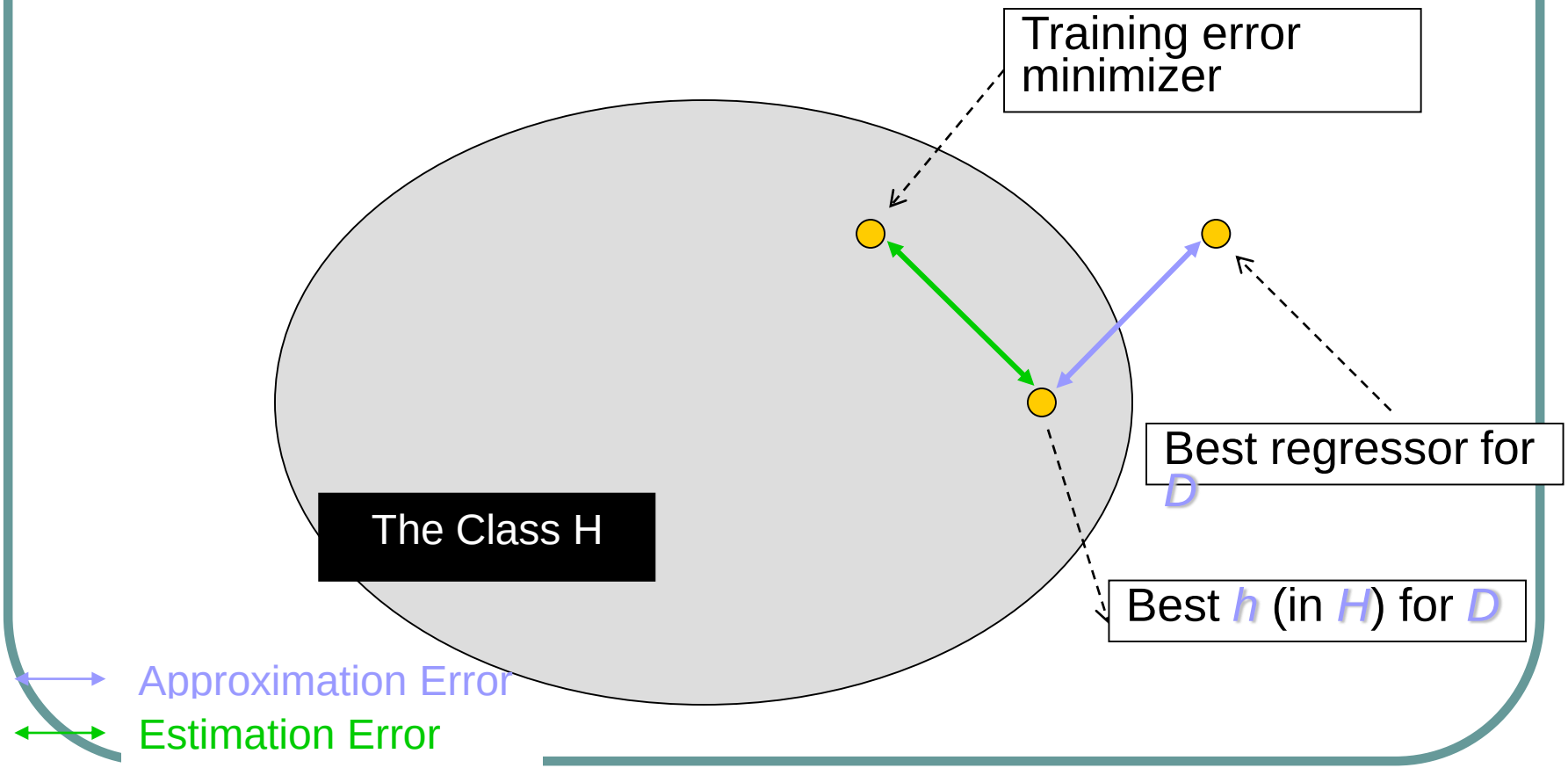
- For ***linear half-spaces*** in R^d it is (d+1)
- For (binary) ***decision trees*** it is the tree depth
- For a ***Neural Network*** it is the number of tunable edges.

Consideration in Picking an ML tool

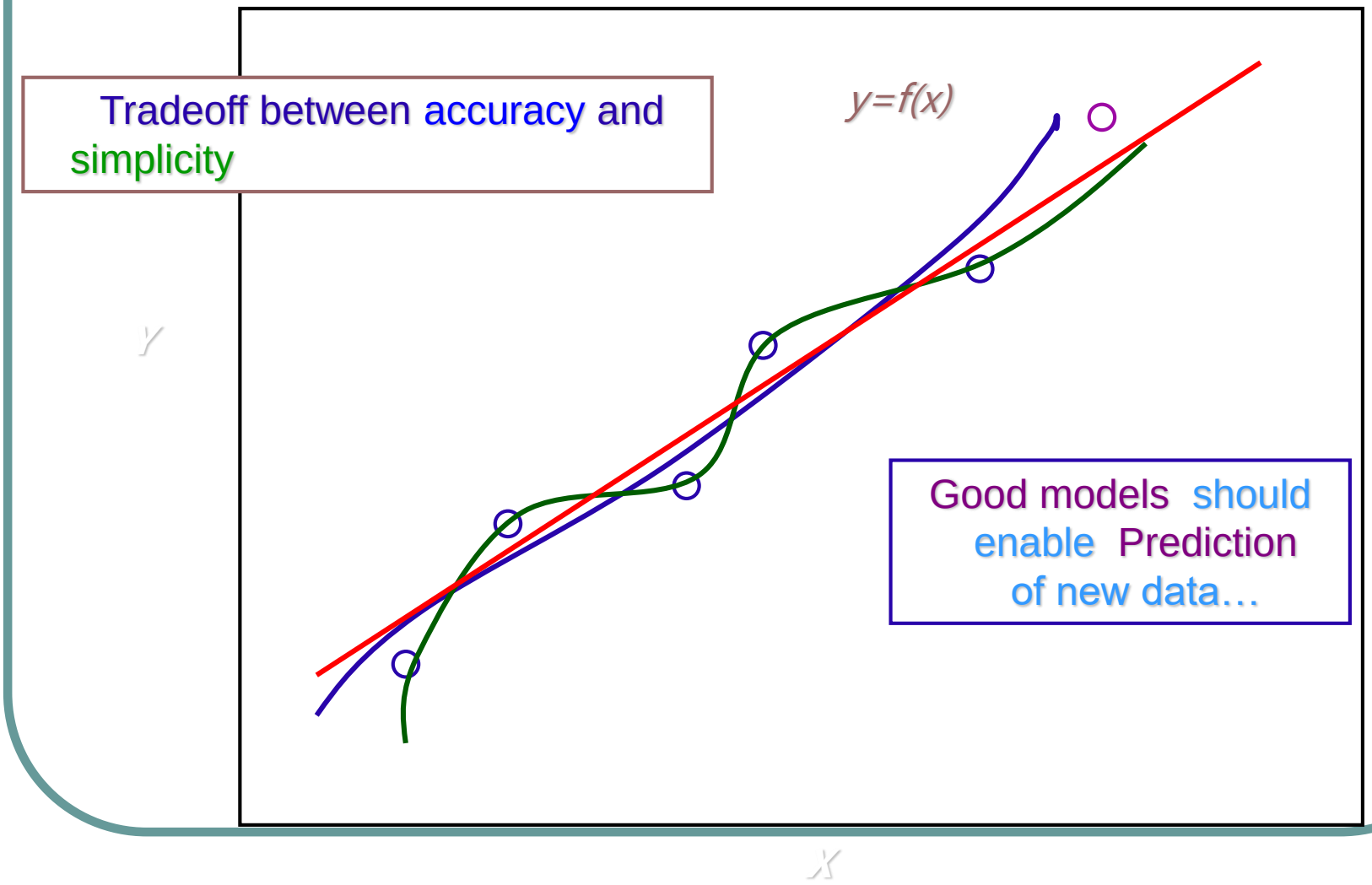
- Expressiveness
- Statistical validity

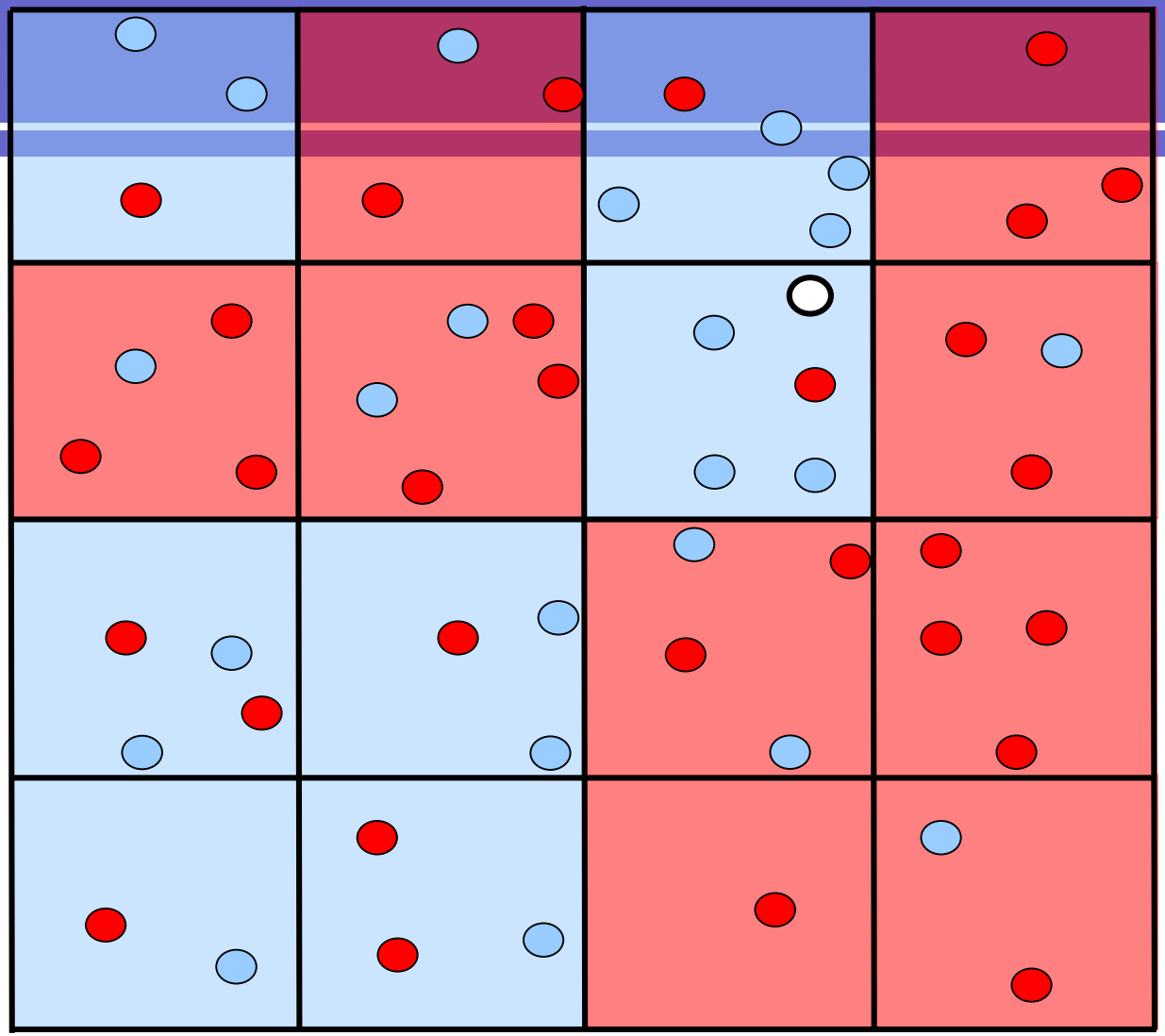
The larger/more-complex the class the more expressive it is, but we pay in statistical validity via a growing VCdim.

The Types of Errors to be Considered

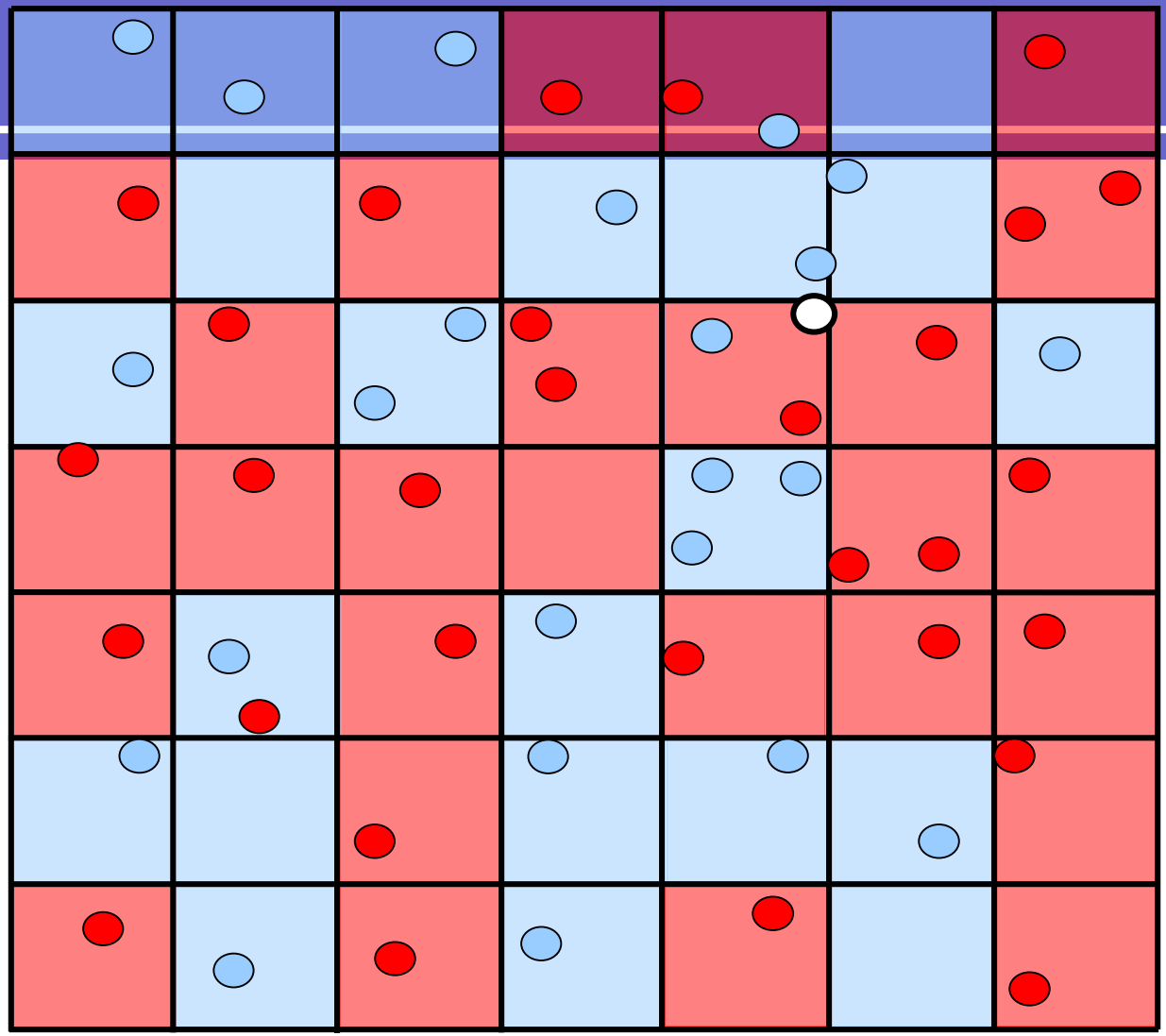


Learning Theory: The fundamental dilemma…



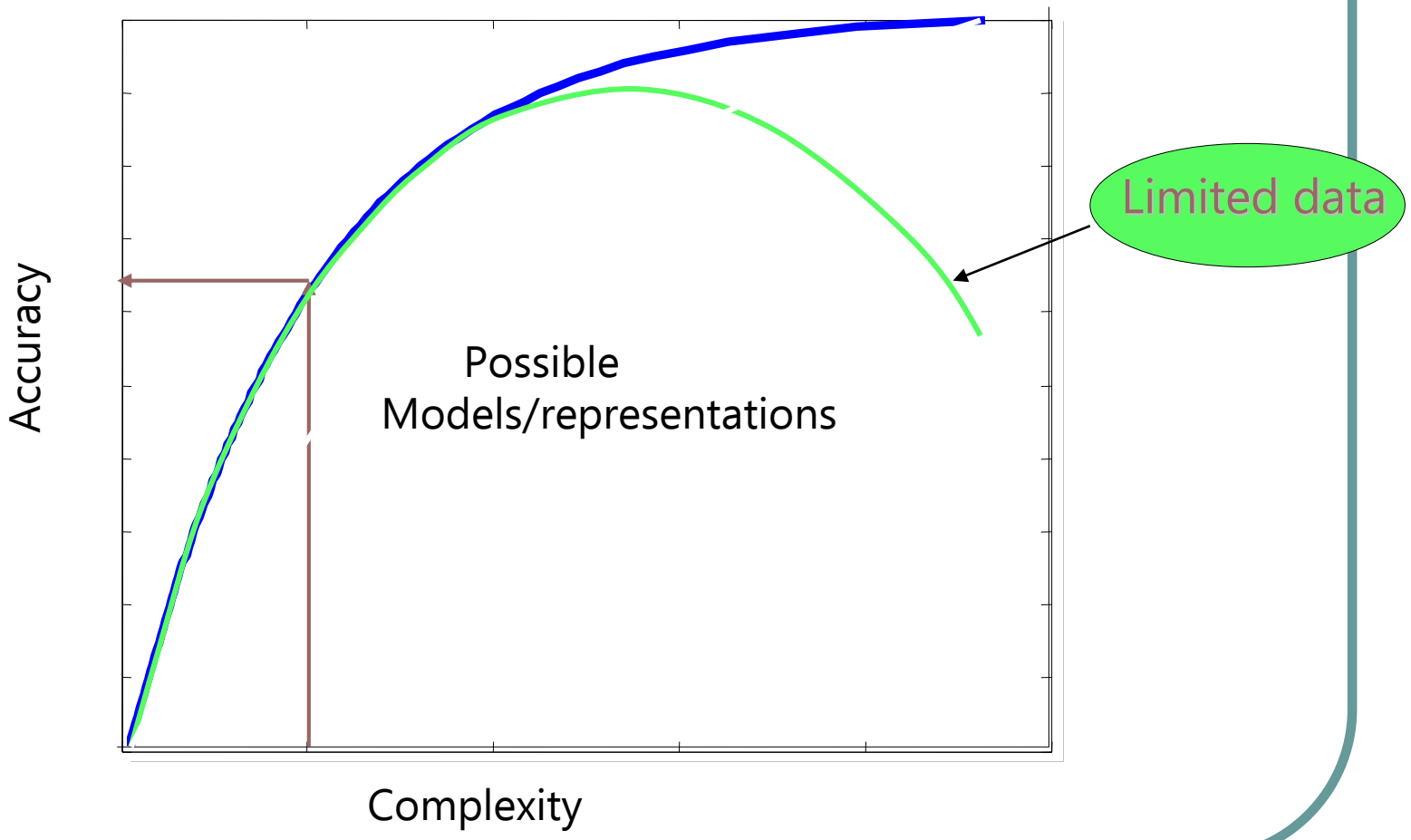


Training error = 16/47



Training error = 2/47

A Fundamental Dilemma of Science:
Model Complexity vs Prediction Accuracy



The model selection problem

Expanding *H*

will *lower* the approximation error

BUT

it will *increase* the estimation error

(lower statistical validity)

Are we done? All solved?

So far we discussed the information complexity of learning - size of training samples.

However, there is another crucial resource:

The runtime of a learner

Different important considerations

Three crucial aspects of learning algorithms:

- Expressiveness
- Statistical validity
- Computational complexity

A crucial resource

Often, enough training examples are available. We still need to process them to get a model/hypothesis.

The simplest learning algorithm is ERM_H

How much computing does it require?

Hardness-of-Optimization Results

For each of the following classes the empirical risk minimization problem is *NP-hard* :

Monomials
Monotone Monomials
Half-spaces
Balls
Axis aligned Rectangles
Neural Networks
k-means clustering

Hardness-of-Approximation Results

For each of the following classes *there exist some constant s. t.* approximating the best agreement rate for *h in H* (on a given input sample *S*) *up to this constant ratio*, is *NP-hard* :

BD-Eiron-Long
- Monomials
- Monotone Monomials
- Half-spaces
- Balls
- Axis aligned Rectangles

Bartlett- BD
- Threshold Neural Networks

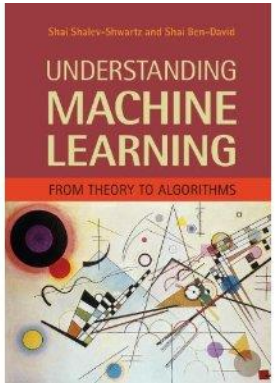
Some other classes are “easy”

- The class of Boolean Conjunctions over binary features.
- Decision stumps in R^d

Directions for addressing computational hardness

- Proper VS “improper” learning.
- Boosting
- Surrogate losses.
- Algorithms with no guarantees….
  (for example –SGD for DNN’s)

Want to know more?



See all 1 image(s)

Publisher: learn how customers can search inside this book.

Tell the Publisher!
I'd like to read this book on Kindle

Don't have a Kindle? Get your Kindle here, or download a **FREE** Kindle Reading App.

Understanding Machine Learning: From Theory to Algorithms [Hardcover]

Shai Shalev-Shwartz (Author), Shai Ben-David (Author)

List Price: ~~CDN$ 62.95~~
Price: CDN$ 50.36 Prime
You Save: CDN$ 12.59 (20%)

Pre-order Price Guarantee. Learn more.

This title has not yet been released.
You may pre-order it now and we will deliver it to you when it arrives.
Ships from and sold by **Amazon.ca**. Gift-wrap available.

Save Up to 90% on Textbooks
Hit the books in Amazon.ca's Textbook Store and save up to 90% on used textbooks and 35% on new textbooks. Learn more.

Vous voulez voir cette page en français ? Cliquez ici.

Quantity: 1

or
Sign in to turn on 1-Click ordering.

Share

Book Description

Publication Date: **May 31 2014** | ISBN-10: **1107057132** | ISBN-13: **978-1107057135**

Machine learning is one of the fastest growing areas of computer science, with far-reaching applications. The aim of this textbook is to introduce machine learning, and the algorithmic paradigms it offers, in a principled way. The book provides an extensive theoretical account of the fundamental ideas underlying machine learning and the mathematical derivations that transform these principles into practical algorithms. Following a presentation of the basics of the field, the book covers a wide array of central topics that have not been addressed by previous textbooks. These include a discussion of the computational complexity of learning and the concepts of convexity and stability;; important algorithmic paradigms including stochastic gradient descent, neural networks, and structured output learning;; and emerging theoretical concepts such as the PAC-Bayes approach and compression-based bounds. Designed for an advanced undergraduate or beginning graduate course, the text makes the fundamentals and algorithms of machine learning accessible to students and non-expert readers in statistics, computer science, mathematics, and engineering.

End of first part