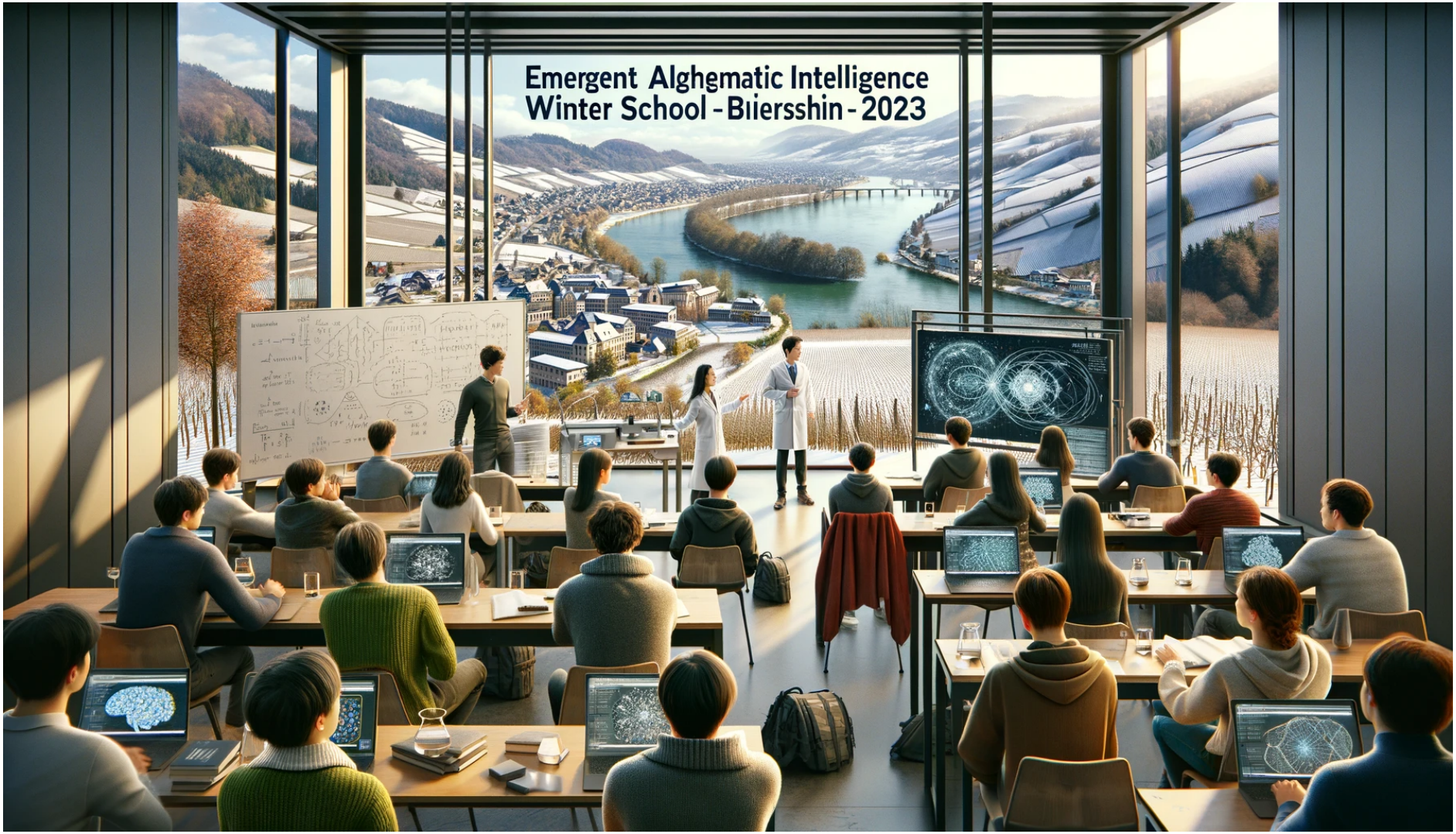
Machine Learning of Dynamic Processes with Applications to Time Series Forecasting

Lyudmila Grigoryeva

University of St. Gallen, Switzerland

Emergent Algorithmic Intelligence Winter School 2023
JGU Research Center for Algorithmic Emergent Intelligence
Mainz (Nierstein), 2023


Emergent Alghematic Intelligence
Winter School - Biiersshin - 2023

Outline

Outline for section 1

Machine Learning

- Classical definition (1959, Arthur Samuel, [Mit97])
- Machine learning as an input/output problem:
  - **Input z** contains available information for the solution of the problem (historical data, explanatory factors, features of the individuals that need to be classified, observations of a dynamical system).
  - **Output y** contains the solution of the problem (forecast, explained variables, classification results, clustered observations).
- The supervised machine learning problem consists in determining (learning) **function(al)s** from **z** to **y** out of *finite sample realizations* of an input and target pairs.
- Important overlap with statistics (parametric and non-parametric) and other fields all claiming leadership.
- Distinctive feature: (mostly) **agnostic** setup.
- Simultaneous interest for stochastic (processes) and deterministic (non-autonomous dynamical) systems.

Classical methods for solving (O/P)DEs[1]

- finite difference
- finite element
- spectral methods

These have completely changed the way we do science, and to an even greater extend, engineering.

- gas dynamics
- structural analysis
- radar, sonar, optics
- control of flight vehicles, satellites

If the finite difference method were invented today, the shock wave that it would generate would be just as strong as the one generated by deep learning.

[1] talk by Weinan E

Many difficult problems remain

- many-body problems (classical and quantum, in molecular science)
- quantum control
- first principle-based drug and materials design
- protein folding
- turbulence, weather forecasting
- transitional flows in gas dynamics
- polymeric fluids
- plasticity
- control problems in high dimensions

Common feature of these problems is dependence on many variables.

When ML may be revolutionary

ML is the solution of choice when dealing with tasks that are:

- too complex to be formulated out of first principles
- even once successfully formulated out of first principles do not admit analytical solution or require infeasible numerical effort (nonlinear high-dimensional PDEs)
- of high uncertainty and dimensions
- multiscale
- too complex to program or learning by memorization is not possible

Some recent work

- inverse problems in imaging
- ML + first principles opens new paradigm for scientific research (physics-informed (PI) neural networks, Gaussian processes, hybrid approaches, many others)
- nonlinear PDEs (scientific machine learning (SciML); for Hamilton-Jacobi-Bellman (HJB), non-linear Black-Scholes, Allen-Cahn-type, nonlinear heat, and sine-Gordon-type, see i.e. work)
- using ML-approximated solutions of parabolic PDEs (Feynman-Kac formula) to solve of SDEs
- predicting policies (see review The Impact of Machine Learning on Economics by S. Athey)
- forecasting time series (weather)

Outline for section 2

Setup(s)

	Static		Dynamic (discrete time)	
	Deterministic	**Stochastic**	**Deterministic**	**Stochastic**
Ingredients	$\mathbf{z} \in \mathbb{R}^d$ $\mathbf{y} \in \mathbb{R}^m$	$\mathbf{z} \in L^p(\Omega, \mathbb{R}^d)$ $\mathbf{y} \in L^p(\Omega, \mathbb{R}^m)$	$\mathbf{z} \in (\mathbb{R}^d)^{\mathbb{Z}_-}$ $\mathbf{y} \in (\mathbb{R}^m)^{\mathbb{Z}_-}$	$\mathbf{z} \in L^p\left(\Omega, (\mathbb{R}^d)^{\mathbb{Z}_-}\right)$ $\mathbf{y} \in L^p\left(\Omega, (\mathbb{R}^m)^{\mathbb{Z}_-}\right)$
Problem examples	$\mathbf{y} = f(\mathbf{z})$ f measurable	$\mathrm{E}[\mathbf{y} \mid \mathbf{z}]$	$\mathbf{y}(\cdot) = F(\mathbf{z}(\cdot))$	$\mathrm{E}[\mathbf{y}(\cdot) \mid \mathbf{z}(\cdot)]$
Examples and Applications	• observables or diagnostics variables in complex physical or noiseless engineering systems • calibration of financial models • translators • transcription	• image classification • speech recognition • factor analysis • anomaly detection	• integration or path continuation of (chaotic) differential equations • molecular dynamics • structural mechanics • vibration analysis • space mission design • autopilot systems • robotics • memory tasks • games	• physiological time series classification • financial bubble detection • time series forecasting • volatility filtering • system identification (blackboxing) • filters (transducers) and equalizers • imputation of missing values • source separators

Data spaces

- $\mathcal{Z}$ - input space (measurement space, feature space, signal domain)
- $\mathcal{Y}$ - output space (output space, label space, response space, signal range)

Particular examples of input space:

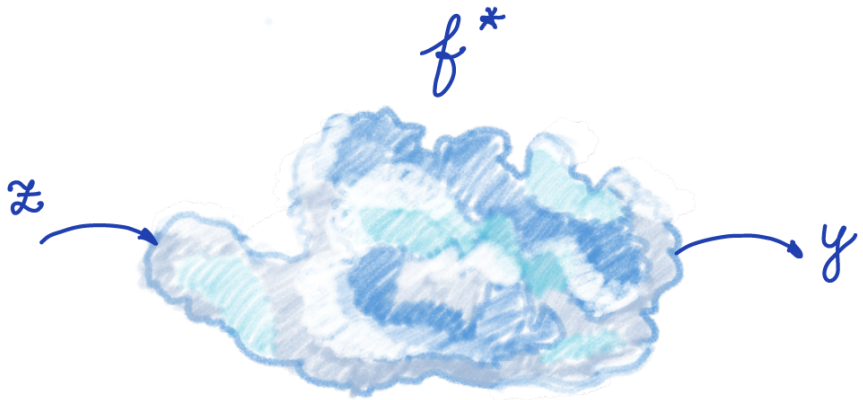
- $\mathcal{Z} \subset \mathbb{R}^d$, $d \in \mathbb{N}$: $\mathbf{Z} \in \mathcal{Z}$ we call predictors (statistics), *features* (pattern recognition), input variables (signals), measurements, covariates (in regression setting)

Particular examples of output space:

- $\mathcal{Y} \subset \mathbb{R}^m$, $m \in \mathbb{N}$: is response or signal, explained variable. For example, in the case of generalized m-dimensional regression $\mathbb{E}[\mathbf{Y}|\mathbf{Z}]$ with $\mathbf{Z} \in \mathcal{Z}$ the task is to find the output $\mathbf{Y} \in \mathcal{Y} \subset \mathbb{R}^m$ given the input $\mathbf{Z}$.
- $\mathcal{Y}$ as a discrete (in particular, binary) set of *labels*. In k-classes classification case with $\mathcal{Y} = \{c_1, c_2, \ldots, c_k\}$ the objective is, given the input of d-dimensional features $\mathbf{Z} = \mathbf{z}$, to estimate $\mathbb{P}(Y = c_i|\mathbf{Z} = \mathbf{z})$, $i \in \{1, \ldots, k\}$. Example: simple binary classification task $\mathcal{Y} = \{-1, 1\}$ and $\mathbb{P}(Y = \pm 1|\mathbf{Z} = \mathbf{z})$.

Candidate (hypothesis) functions

The goal is to find the true relation $f^* : \mathcal{Z} \to \mathcal{Y}$



One may propose a candidate function $f : \mathcal{Z} \to \mathcal{Y}$

Loss functions

How well does this candidate function $f : \mathcal{Z} \to \mathcal{Y}$ capture the true relationship? We measure that by some loss function

Definition

A *loss function* is a mapping $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_0^+$.

Example: for $\hat{Y} := f(Z) \in \mathcal{Y}$, the loss $\ell(\hat{Y}, Y) = \ell(f(Z), Y) \in \mathbb{R}_0^+$ measures the discrepancy between the result of the model for the given Z and the true target Y. Let another Z' be given. We would measure the associated loss $\ell(f(Z'), Y) \in \mathbb{R}_0^+$ again.

Question: What is the problem with this approach? How would we measure the performance of the candidate function on average with respect to different drawn Z and different corresponding Y?

Examples of loss functions in classification

Consider $\mathcal{Y} = \{0, 1\}$

- "0-1" loss $L(\hat{Y}, Y) = \mathbb{1}_{\hat{Y} \neq Y}$. In classification tasks it counts misclassifications
- asymmetric loss functions (confusion matrix and its derivatives based), for example in the case when the cost of Type I error (FP) is higher than of Type II error (FN). In the spam filter application it is preferable to rather misclassify spam emails "1" than legitimate emails "0":

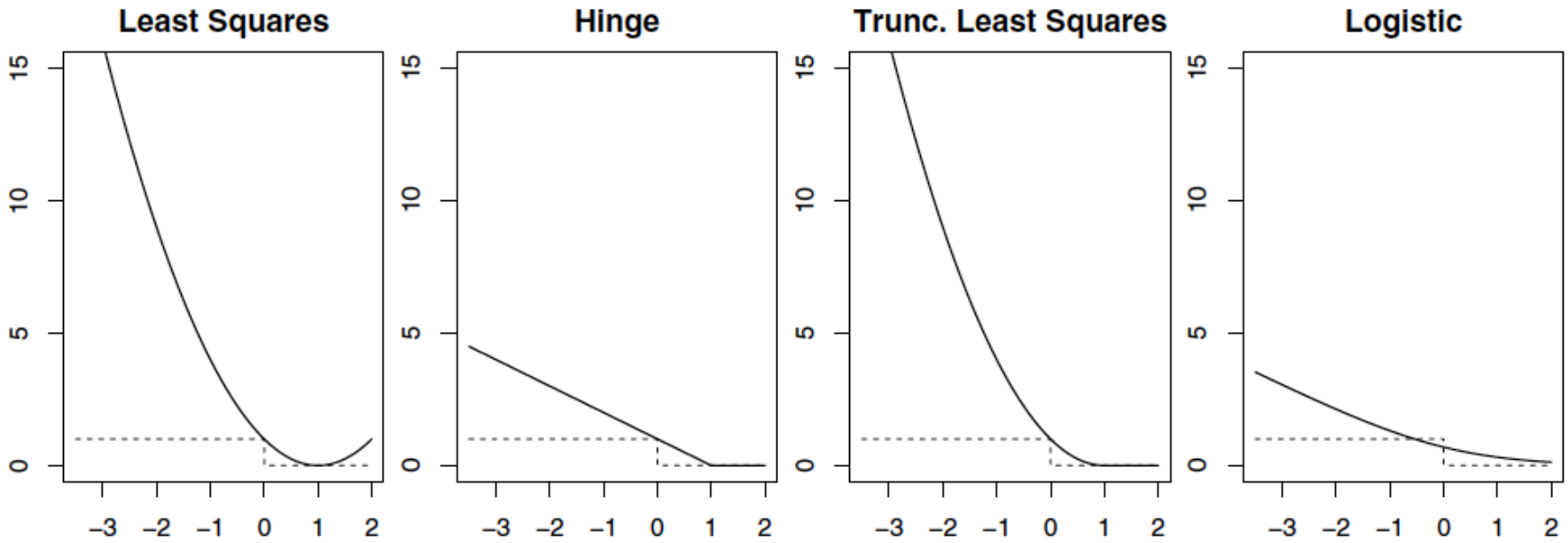
$$\ell(\hat{Y}, Y) = \begin{cases} 10, & \hat{Y} = 1, Y = 0 \\ 2, & \hat{Y} = 0, Y = 1 \\ 0, & \hat{Y} = Y \end{cases}$$

Examples of loss functions in classification

Consider $\mathcal{Y} = \{-1, 1\}$

- *squared* $\ell(\hat{Y}, Y) = (Y - \hat{Y})^2 = (1 - Y\hat{Y})^2$
  convex, locally Lipschitz
- *hinge* $\ell(\hat{Y}, Y) = \max\{0, 1 - Y\hat{Y}\}$
  convex, Lipschitz $L = 1$
- *truncated squared* (squared hinge loss) $\ell(\hat{Y}, Y) = (\max\{0, 1 - Y\hat{Y}\})^2$
  convex, locally Lipschitz
- *logistic* $\ell(\hat{Y}, Y) = \ln(1 + \exp(-Y\hat{Y}))$
  convex, Lipschitz $L = 1$, $\mathcal{C}^\infty$

Illustration of popular classification losses



Taken from [CSC08]

Examples of loss functions in regression/approximation

Consider $\mathcal{Y} = \mathbb{R}$.

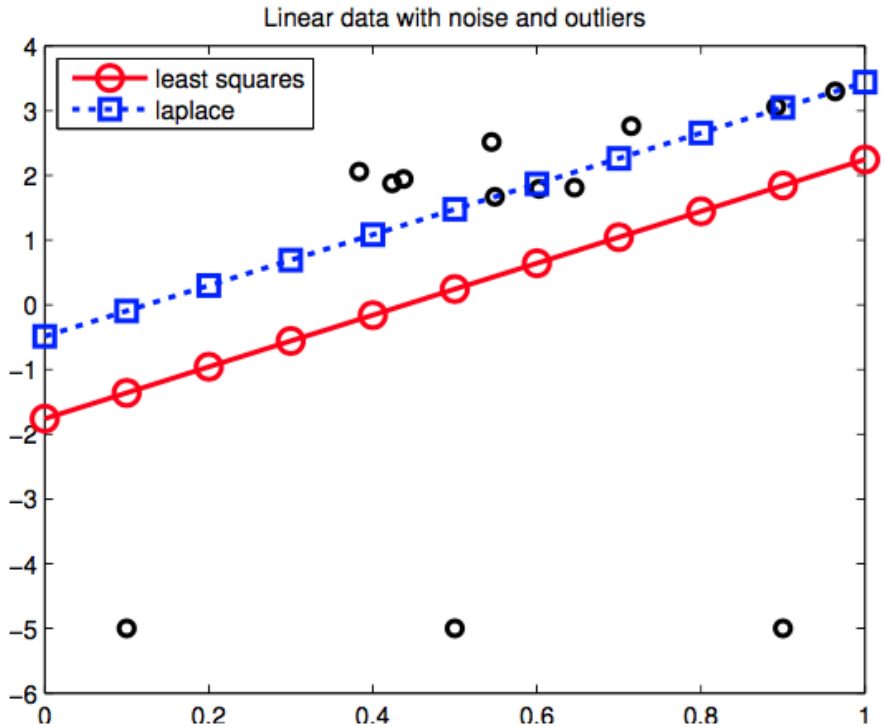
- *ℓ_p-losses* $\quad \ell(\hat{Y}, Y) = |Y - \hat{Y}|^p,\ p \geq 1$
  - ℓ_2 *squared* - the most common case
    not robust but stable
  - ℓ_1 *absolute (Laplace)*
    not everywhere differentiable
- *σ-insensitive* $\quad \ell(\hat{Y}, Y) = (|Y - \hat{Y}| - \sigma)\mathbb{1}_{|Y-\hat{Y}|>\sigma}$
  Lipschitz, convex
- *Huber's robust* (1964) with $\delta > 0$, robust, Lipschitz $L = \delta$, and differentiable

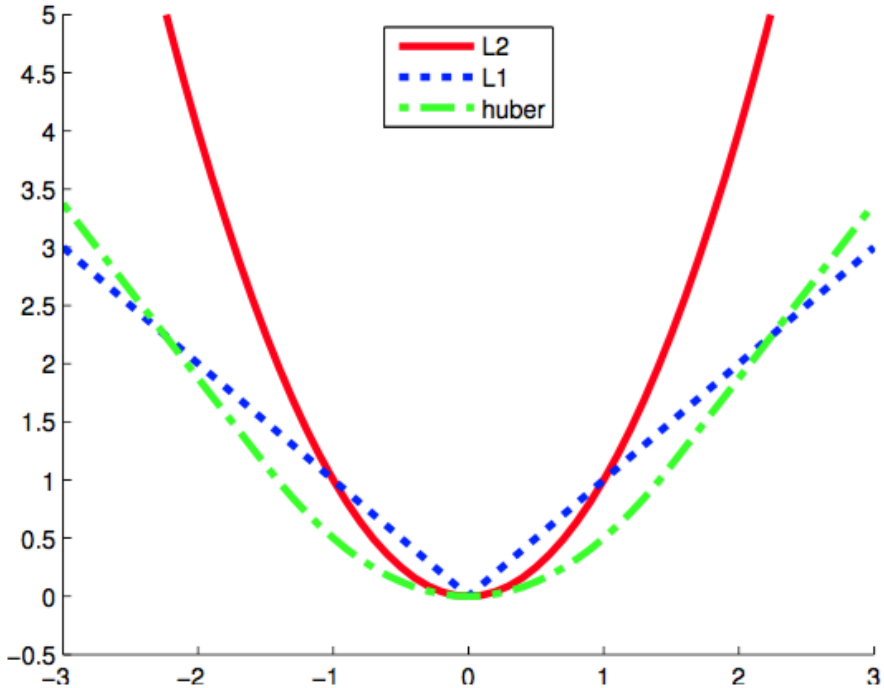
$$\ell(\hat{Y}, Y) = \begin{cases} \frac{1}{2}(Y - \hat{Y})^2, & \text{if } |Y - \hat{Y}| \leq \delta \\ \delta|Y - \hat{Y}| - \frac{\delta^2}{2}, & \text{if } |Y - \hat{Y}| > \delta \end{cases}$$

- *log-barrier*

$$\ell(\hat{Y}, Y) = \begin{cases} -a^2 \log\left(1 - (|Y - \hat{Y}|/a)^2\right), & \text{if } |Y - \hat{Y}| \leq a \\ \infty, & \text{if } |Y - \hat{Y}| > a \end{cases}$$

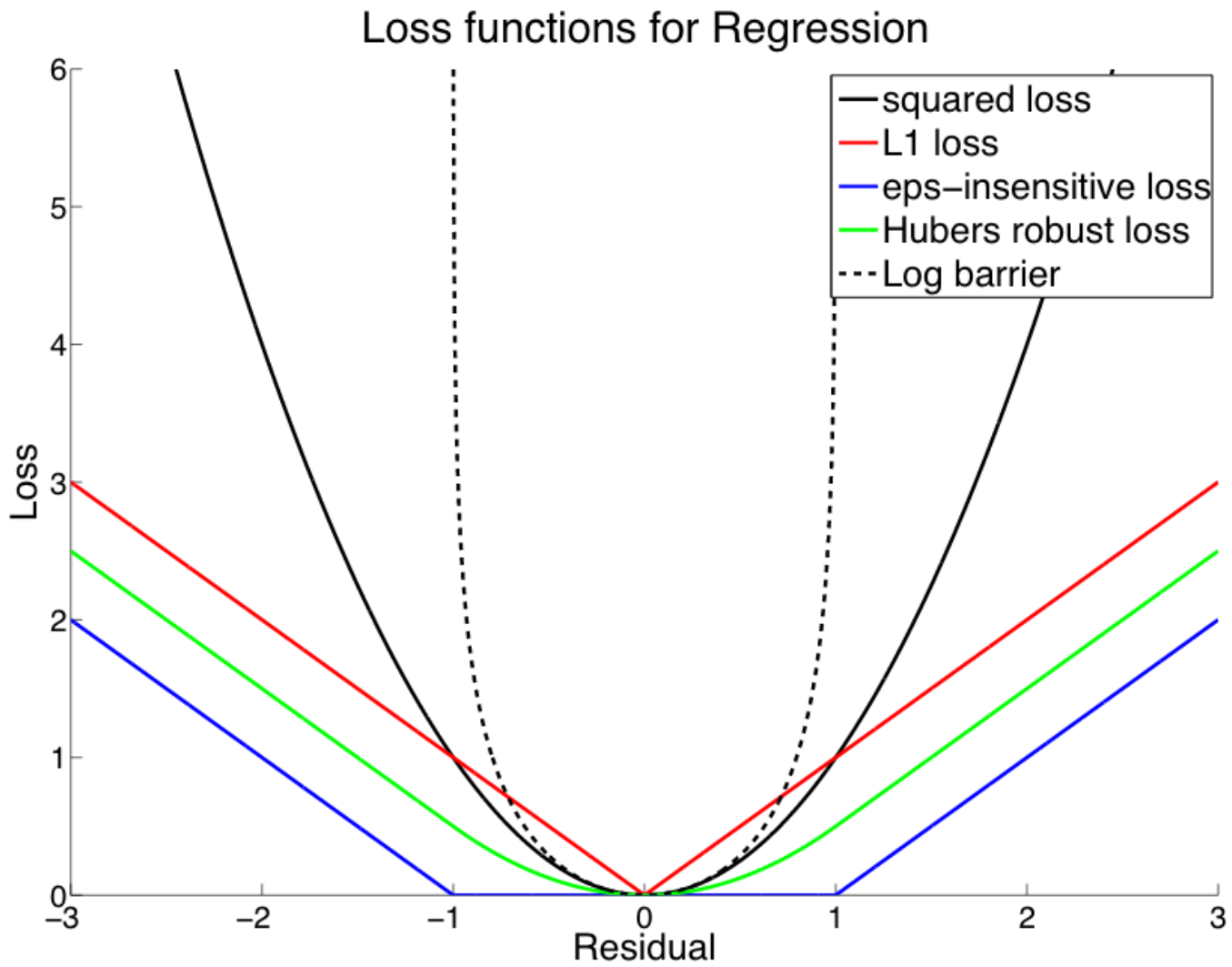
Illustration of popular regression losses





Taken from [Ros15]

Illustration of popular regression losses



Taken from Matthias Hein

Figure: $\sigma = 1$, $a = 1$

Statistical risk

Definition

The performance of a map $f : \mathcal{Z} \to \mathcal{Y}$ with respect to a loss $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_0^+$ is measured by its expected loss or statistical risk or generalization error given by

$$R(f) = \mathbb{E}_{ZY}\left[\ell(f(Z), Y)\right]. \tag{1}$$

Always exists (nonnegative loss), not necessarily finite.

Idea: One wants to have this risk as small as possible.

Using the law of iterated expectations

$$R(f) = \mathbb{E}_Z\left[\mathbb{E}_{Y|Z=z}[\ell(f(Z), Y) \mid Z = z]\right]$$

Relation to properties of loss functions

Some additional properties of loss functions are related to analogous properties of the associated risks (see careful introduction here)

1. Convex losses $\Longrightarrow$ convex risks
2. Continuous, P-integrable Nemitski losses $\Longrightarrow$ continuous risks (continuity of losses $\not\Longrightarrow$ continuity of risks in general)
3. (Locally) Lipschitz continuous losses $\Longrightarrow$ (locally) Lipschitz continuous risks
4. Differentiable losses (both L and L' P-integrable Nemitski) $\Longrightarrow$ differentiable risks (differentiability of losses $\not\Longrightarrow$ differentiability of risks in general)

Bayes risk and Bayes function

Definition

The *minimum risk value (Bayes risk)* is defined as:

$$R^*_{\mathcal{F}} := R(f^*_{\mathcal{F}}) = \inf_{f \in \mathcal{F}} R(f), \tag{2}$$

where $\mathcal{F}$ is a class of all measurable functions. We call $f^*_{\mathcal{F}}$ the *Bayes function*.

Remark

In the deterministic case, one has $Y = f^*_{\mathcal{F}}(Z)$ *a.s. and* $R^*_{\mathcal{F}} = 0$.

Regression with quadratic loss

Whenever the quadratic loss is chosen, the statistical risk of f is given by

$$R(f) = \mathbb{E}\left[\ell(f(Z), Y)\right] = \mathbb{E}\left[(f(Z) - Y)^2\right].$$

and, moreover, one can explicitly determine the Bayes function $f^*_{\mathcal{F}}$.

Theorem

The minimum statistical risk (Bayes risk) is achieved by the regression function $f^*_{\mathcal{F}}(z) = \mathbb{E}\left[Y|Z = z\right]$, namely $R^*_{\mathcal{F}} = R(f^*_{\mathcal{F}})$, that is the conditional expectation is the best MSE predictor (Bayes function).

The excess risk of any f is given by:

$$R(f) - R^*_{\mathcal{F}} = \mathbb{E}_X\left[(f(X) - \mathbb{E}\left[Y|X\right])^2\right]. \tag{3}$$

Remark

Whenever a different loss function is chosen, the regression function is not a conditional mean anymore. For example, for ℓ_1-norm based loss (or L_1 loss) one has conditional median as a regression function (least absolute deviations regression; solution is given by linear programming).

Problem

Ultimate goal: find $f^*_{\mathcal{F}}$

1. Distribution is known (partially): semi-agnostic, (semi-)informed setup
   - statistical decision theory
   - If the true distribution is given: one can succeed in compute the statistical risk explicitly (see for example [Wu16])
2. Distribution is not known: agnostic setup, uninformed setup

- In the agnostic setup: finding $f^*_{\mathcal{F}}$ is infeasible
- In the informed setup: even if the chosen loss functions have properties which allow for good optimization practices, the class $\mathcal{F}$ presents a problem.

Instead of working with $\mathcal{F}$ one restricts the task to the so-called hypothesis class (in general much smaller) $\mathcal{H} \subset \mathcal{F}$.

Complexity of this class determines the quality of learning.

Examples: finite, class of affine functions, class of polynomial functions, class of (deep) neural networks, *class of reservoir computing systems*

Bayes **in-class** risk and Bayes **in-class** function

Definition

The minimum **in-class** risk value (Bayes **in-class** risk) is defined as:

$$R^*_{\mathcal{H}} := R(f^*_{\mathcal{H}}) = \inf_{f \in \mathcal{H}} R(f), \tag{4}$$

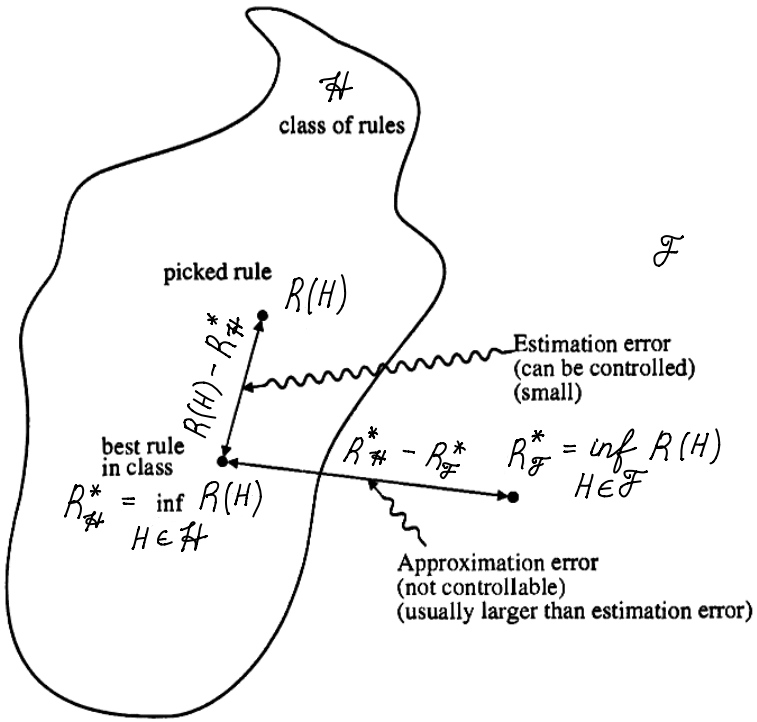
where $\mathcal{H}$ is a hypothesis class chosen by the learner. $R^*_{\mathcal{H}}$ is assumed achievable and we call $f^*_{\mathcal{H}}$ the Bayes **in-class** function.

Ultimate goal: find $f^*_{\mathcal{H}}$ which in general is again infeasible since the law is not known and/or again even the smaller hypothesis class $\mathcal{H}$ is too large.

Statistical learning problem

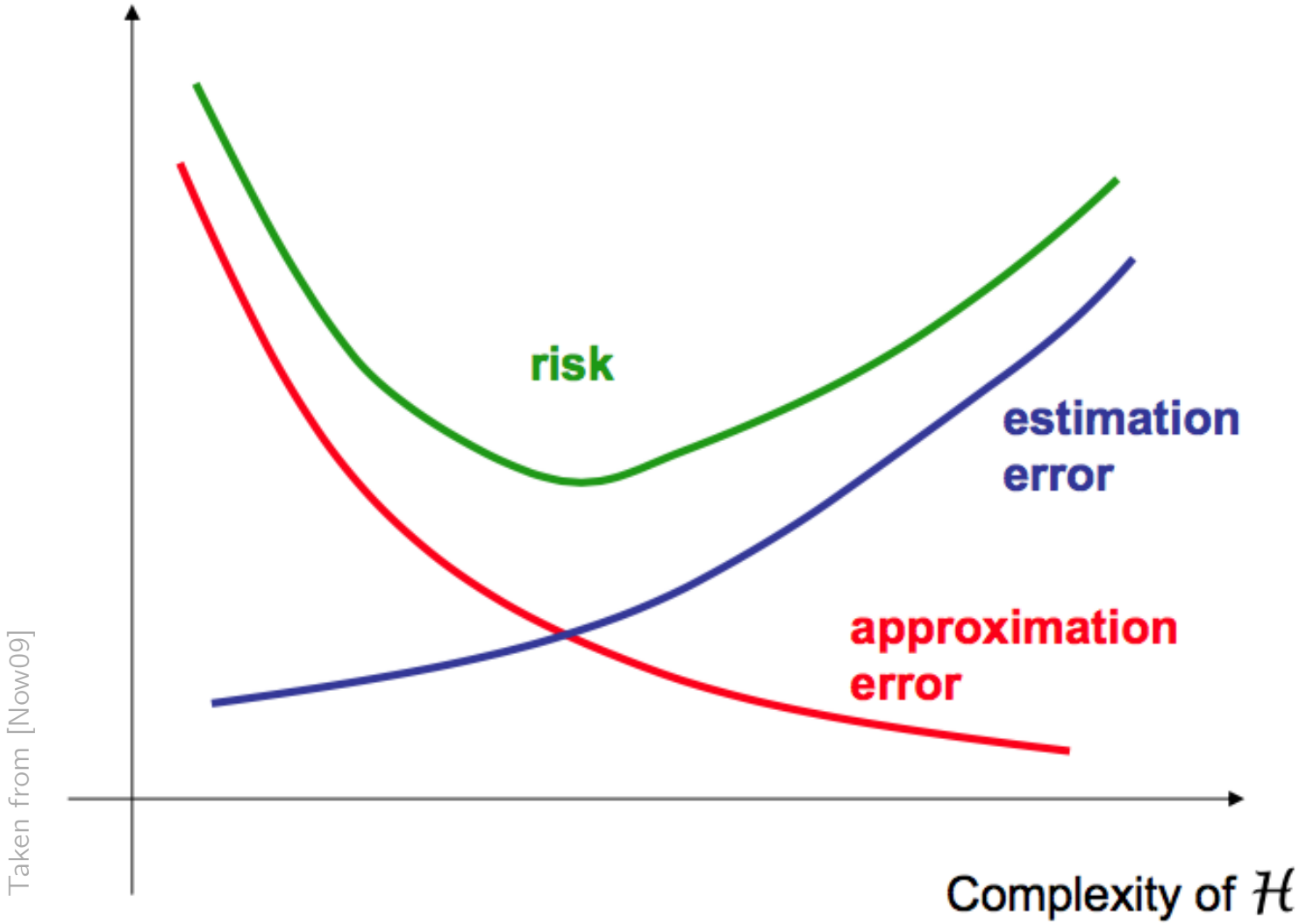
Statistical learning task consists in designing a *hypothesis class* and a *learning rule* such that for any chosen by this rule $f \in \mathcal{H}$ the associated excess error $(R(f) - R^*_{\mathcal{F}})$ is as small as possible. The error decomposition is given by

$$R(f) - R^*_{\mathcal{F}} = \underbrace{(R(f) - R^*_{\mathcal{H}})}_{\text{estimation error}} + \underbrace{(R^*_{\mathcal{H}} - R^*_{\mathcal{F}})}_{\text{approximation error}}$$



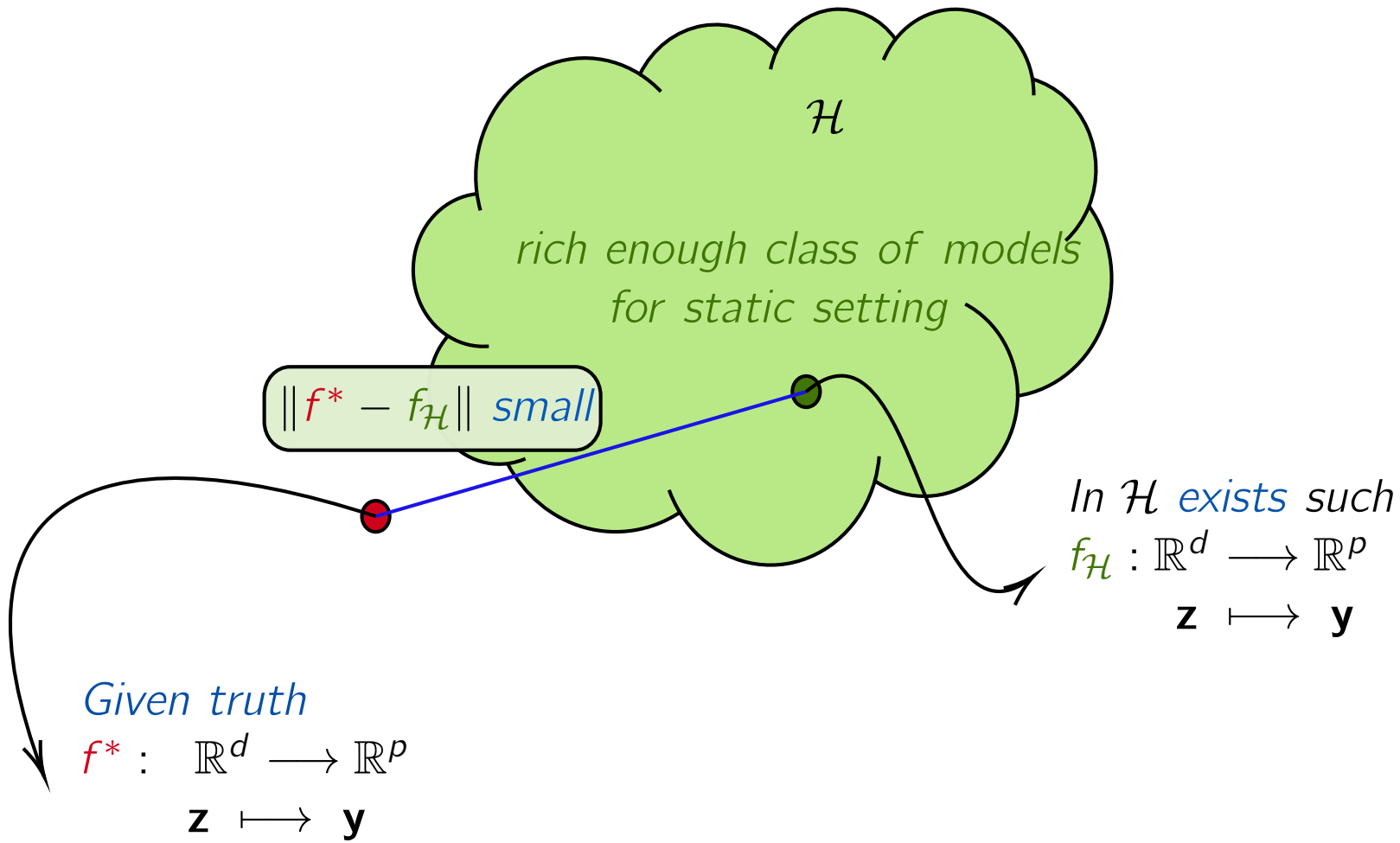
Estimation and approximation errors: Illustration

Tradeoff between estimation and approximation errors as a function of the complexity of $\mathcal{H}$.



Approximation error

For hypothesis classes with the universal approximation properties the approximation error may be made as small as needed.


Universal family of models
$\mathcal{H}$
rich enough class of models
for static setting
$\|f^* - f_{\mathcal{H}}\|$ small
In $\mathcal{H}$ exists such
$f_{\mathcal{H}} : \mathbb{R}^d \longrightarrow \mathbb{R}^p$
$\mathbf{z} \longmapsto \mathbf{y}$
Given truth
$f^* : \mathbb{R}^d \longrightarrow \mathbb{R}^p$
$\mathbf{z} \longmapsto \mathbf{y}$

Estimation error

What about estimation error?

Can we say something about $R(f) = \mathbb{E}_{ZY}[\ell(f(Z), Y)]$?

What are the difficulties in evaluating it?

Learning algorithm (learning procedure)

Since the law is not available, one can use empirical counterpart of statistical risk (1).

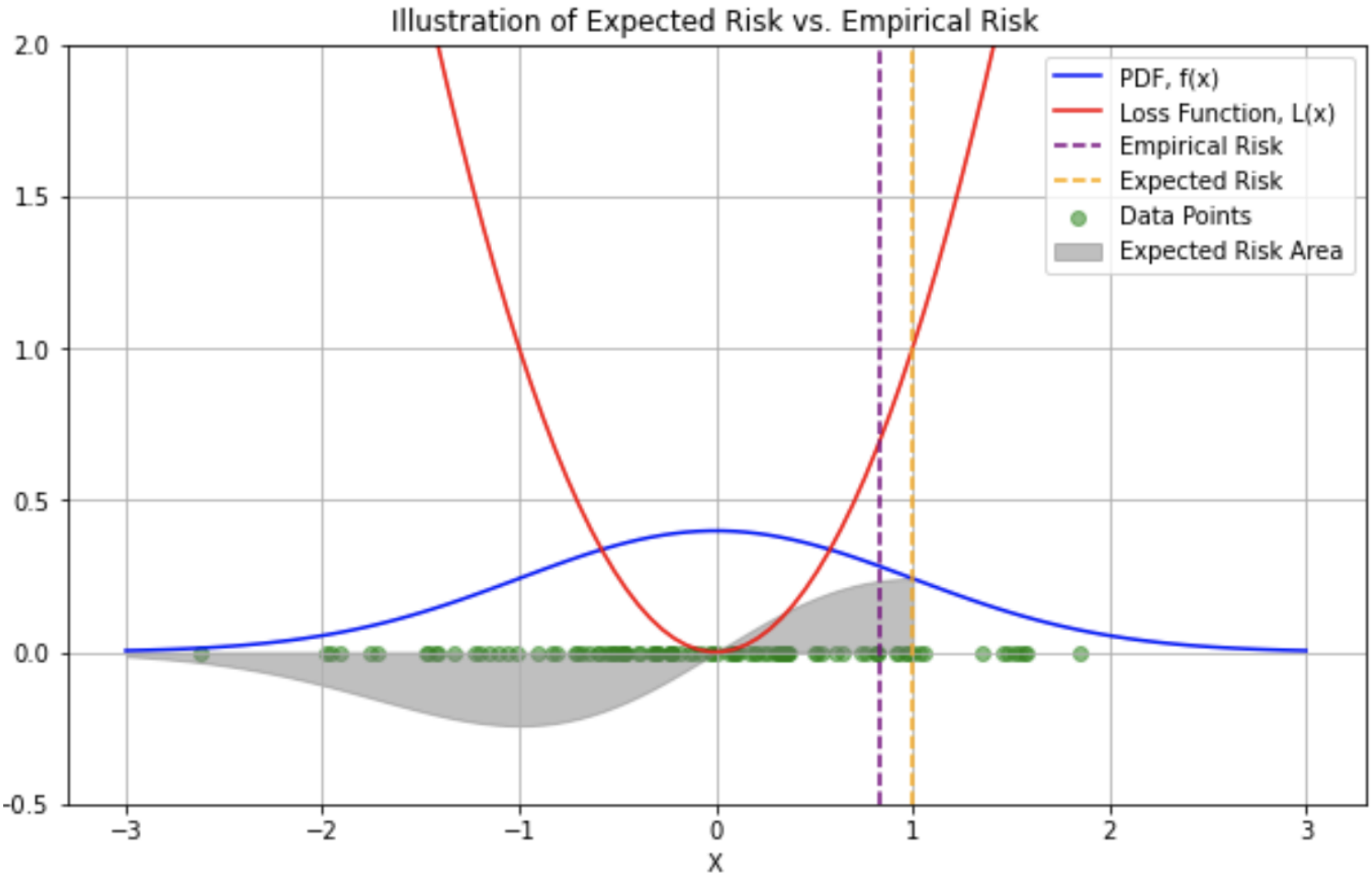
Definition

Let $D_n := \{(Z_i, Y_i)\}_{i\in\{1,\dots,n\}}$, $Z_i \in \mathcal{Z}$, $Y_i \in \mathcal{Y}$, with $(Z_i, Y_i) \overset{IID}{\sim} \mathbb{P}_{ZY}$ be a training sample (training set). Define the empirical risk of any $f \in \mathcal{H}$ associated to $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_0^+$ as

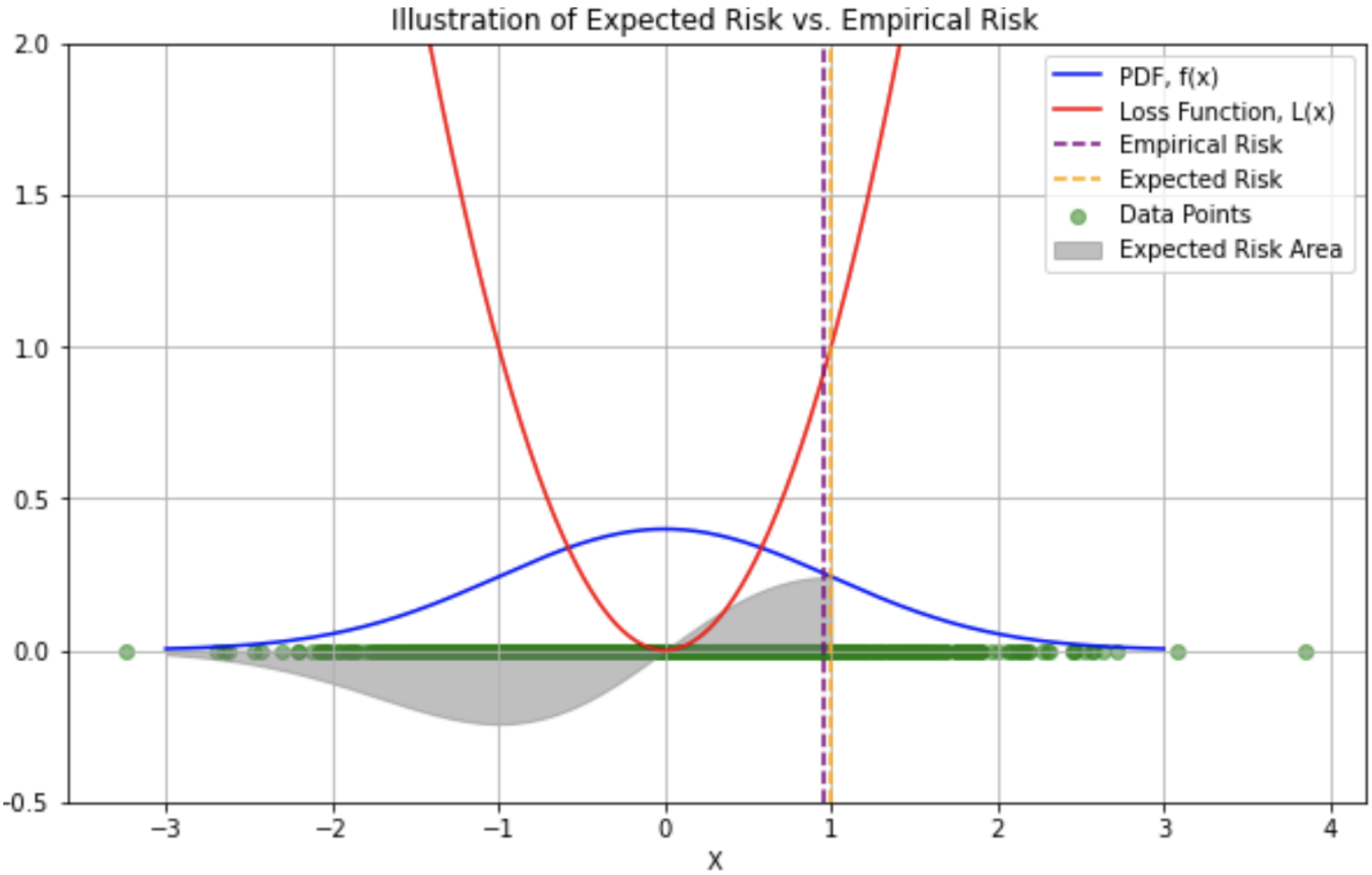
$$\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(Z_i), Y_i). \tag{5}$$

Natural strategy: to construct the learning procedure based on empirical risk.

Empirical Risk



Empirical Risk



Empirical Risk Minimization (ERM)

Definition

Empirical Risk Minimization (ERM) is a learning procedure which consists in selecting a hypothesis from a given hypothesis class which is an empirical risk minimizer within $\mathcal{H}$, namely

$$\hat{f}_n = \arg\min_{f \in \mathcal{H}} \hat{R}_n(f),$$

where $\hat{R}_n(f)$ is defined in (5), and $\hat{f}_n$ is well defined provided that such a minimizer exists and is unique (otherwise one may define $\hat{f}_n$ to be an ϵ-minimizer of the empirical risk (see Alon et al. (1997) for details). We call $\hat{R}_n(\hat{f}_n)$ a training error of $\hat{f}_n$ on the training set D_n.

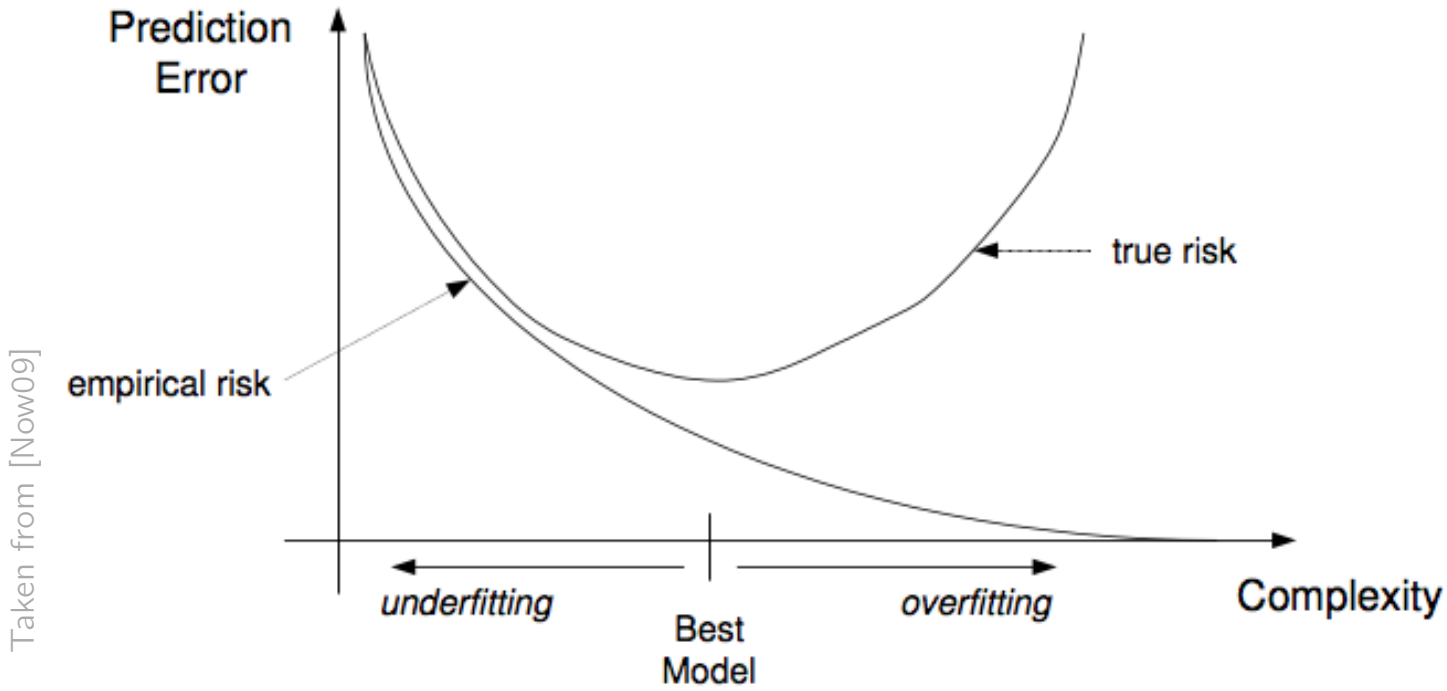
Remark

Notice that $\hat{R}_n(\hat{f}_n)$ is not an unbiased estimate of risk $R(\hat{f}_n)$ since

$$\mathrm{E}_{D_n}\left[R(\hat{f}_n) - \hat{R}_n(\hat{f}_n)\right] \geq 0.$$

Empirical and true risk

Overfitting refers to the situation where we have a small empirical risk but still a relatively large true risk. It could happen especially when sample size n is small or/and hypothesis space $\mathcal{H}$ is large.



Definition

The ERM-picked functional $\hat{f}_n$ is (universal, strong) risk-consistent within $\mathcal{H}$ if for some (all) distribution (X, Y)

$$R(\hat{f}_n) \xrightarrow[n\to\infty]{\mathbb{P}(a.s.)} R^*_{\mathcal{H}} \quad \text{and} \tag{6}$$

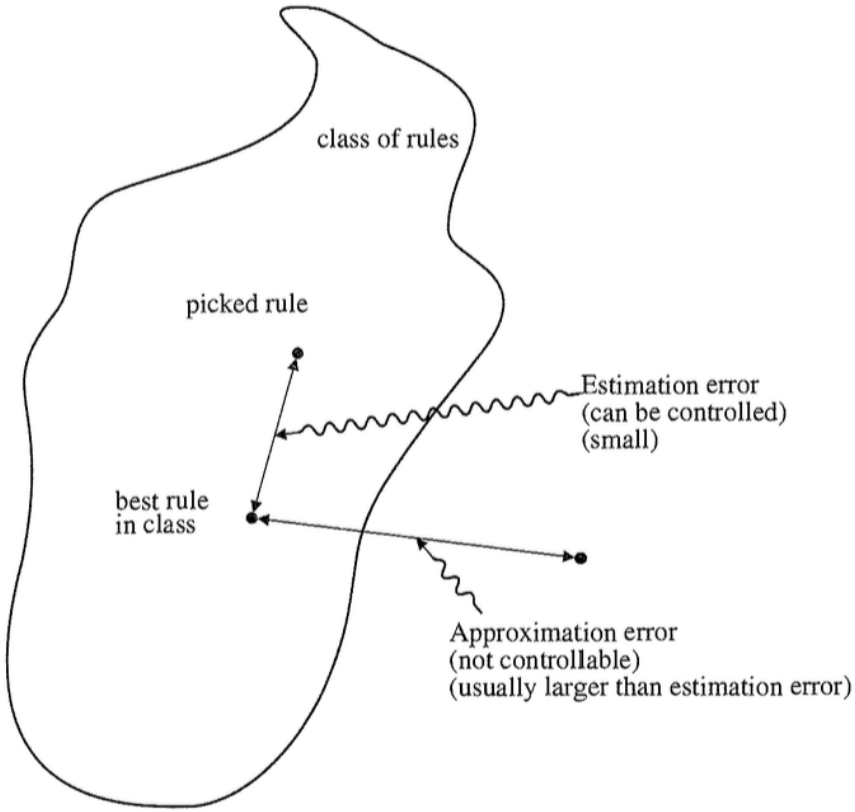
$$\hat{R}_n(\hat{f}_n) \xrightarrow[n\to\infty]{\mathbb{P}(a.s.)} R^*_{\mathcal{H}}. \tag{7}$$

Properties of ERM

The error decomposition is given by

$$R(\hat{f}_n) - R^*_{\mathcal{F}} = \underbrace{(R(\hat{f}_n) - R^*_{\mathcal{H}})}_{\textit{estimation error}} + \underbrace{(R^*_{\mathcal{H}} - R^*_{\mathcal{F}})}_{\textit{approximation error}}$$

and it turns out that handling estimation error is much easier.



For hypothesis classes with the strongly consistent ERM rules asymptotic

Vapnik-Chervonenkis result

Use a simple uniform convergence argument for the whole class $\mathcal{H}$. A standard approach to proving the strong risk-consistency of the ERM procedure for the hypothesis class of functionals $\mathcal{H}$ consists in finding a sequence $(\eta_n)_{n\in\mathbb{N}}$ converging to zero for which the inequality

$$\overline{\Delta}_n := \sup_{f\in\mathcal{H}} |\widehat{R}_n(f) - R(f)| \leq \eta_n,$$

holds $\mathbb{P}$-a.s. To see that this implies (6) and (7) one notes the following inequalities:

$$\begin{aligned} R(\widehat{f}_n) - R^*_{\mathcal{H}} &= \left(R(\widehat{f}_n) - \widehat{R}_n(\widehat{f}_n)\right) + \left(\widehat{R}_n(\widehat{f}_n) - \widehat{R}_n(f^*)\right) + \left(\widehat{R}_n(f^*) - R^*_{\mathcal{H}}\right) \\ &\leq 2\eta_n + \left(\widehat{R}_n(\widehat{f}_n) - \widehat{R}_n(f^*)\right) \leq 2\eta_n, \end{aligned} \tag{8}$$

where the last inequality follows from the fact that, by definition of ERM rule, $\widehat{f}_n$ is a minimizer of the empirical risk $\widehat{R}_n$.
[Vap98]: uniform two-sided (one-sided) convergence over $\mathcal{H}$ of the empirical risk to the risk (generalization error) is a sufficient condition (+necessary) for consistency of the ERM principle applied to $\mathcal{H}$

Generalization properties

Using empirical counterparts $\Longrightarrow$ interest in the so-called generalization of a given $f \in \mathcal{H}$: $\left\{R(f) - \hat{R}_n(f)\right\}$ or the generalization properties of the class $\mathcal{H}$

$\Delta_n := \sup_{f \in \mathcal{H}} \left\{R(f) - \hat{R}_n(f)\right\}$.

The generalization bounds (upper bounds for Δ_n) can be used:

- in the case of arbitrary learning rules to construct upper bounds for the estimation error $(R(f) - R^*_{\mathcal{H}})$
- particular relevance for the learning rules which use the empirical risk $\hat{R}_n(f)$ as a criterion to pick the $f \in \mathcal{H}$; the finite-sample bounds provide information about the complexity of the sample $n(\varepsilon, \delta) \in \mathbb{N}^+$ needed to achieve a required accuracy $\varepsilon > 0$ with a given probability $1 - \delta$, $\delta \in (0, 1)$
- in the special case of the Empirical Risk Minimization (ERM) procedure asymptotic behavior of these bounds allow to establish important properties: (universal, weak or strong) consistency of the ERM-picked functional from the class $\mathcal{H}$.

Using PAC-bounds provides generalization error estimates and allows to prove consistency of the ERM rules.

Statistical learning problem

- $\mathcal{F} := \{f : \mathbb{R}^d \to \mathbb{R} \ : \ f \text{ measurable}\}$
- *Hypothesis class* $\mathcal{H}$ of admissible functions $\mathcal{H} \subset \mathcal{F}$
- *Loss* $L : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^+$
- *Statistical risk* or *generalization error* associated with $f \in \mathcal{F}$

$$R(H) := \mathbb{E}[L(f(\mathbf{x}), y)]. \tag{9}$$

- Ultimate goal of the learning - find the *Bayes function* $H^*_{\mathcal{F}} \in \mathcal{F}$ with the minimal associated statistical risk (*Bayes risk*) in $\mathcal{F}$

$$R^*_{\mathcal{F}} := R(f^*_{\mathcal{F}}) = \inf_{f \in \mathcal{F}} R(H). \tag{10}$$

- Feasible goal - find the *best-in-class* function $f^*_{\{} \in \{$ with the minimal associated in-class statistical risk (*Bayes in-class risk*)

$$R^*_{\mathcal{H}} := R(f^*_{\mathcal{H}}) = \inf_{f \in \mathcal{H}} R(f). \tag{11}$$

- *Statistical learning task* consists in designing a *learning rule* which provides $f \in \mathcal{H}$ with the smallest possible associated excess error

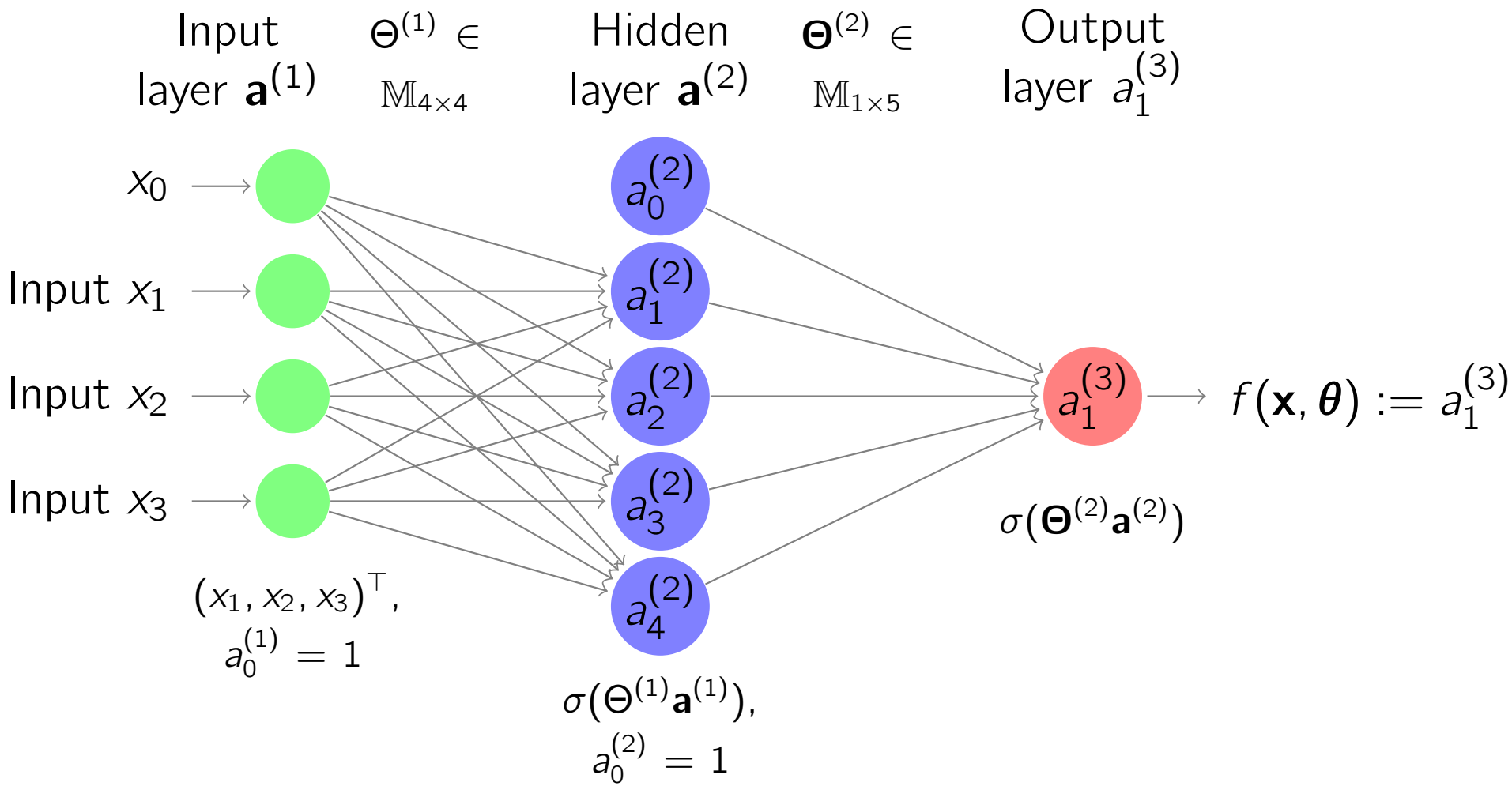
$$R(f) - R^*_{\mathcal{F}} = \underbrace{(R(f) - R^*_{\mathcal{H}})}_{\text{estimation error}} + \underbrace{(R^*_{\mathcal{H}} - R^*_{\mathcal{F}})}_{\text{approximation error}}$$

Desired properties of machine learning models

1. Universal approximation properties of the hypothesis class
2. (Universal) strong consistency of ERM rules within hypothesis class
3. Generalization guarantees with small sample complexity
4. Efficiency in computation

So far we consider only static situations.

Feedforward neural networks (FNN): single hidden layer



A feedforward neural network is a multivariate regression model $\mathbf{y} = f(\mathbf{x}, \boldsymbol{\theta})$ where the output is obtained out of the input via the composition of nonlinear functions across the layers and linear functions within each layer.

Outline for section 3

Approximation of continuous functions

Theorem (Cybenko [Cyb89])

Let σ be a continuous squashing (sigmoid) function. Then the functions $f_{\sigma,NN} : I^d \to \mathbb{R}$ of the form

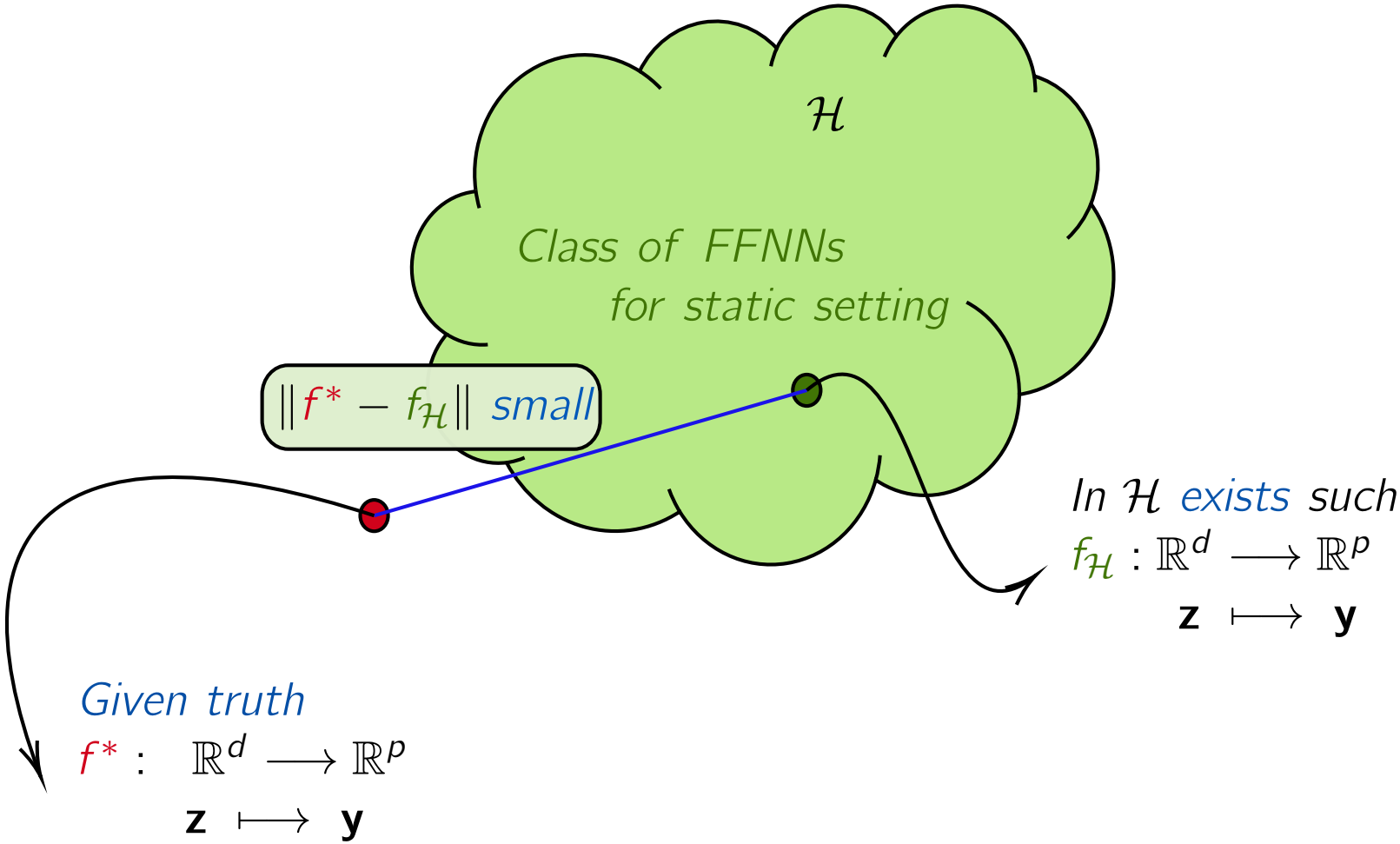
$$f_{\sigma,NN}(\mathbf{z};\boldsymbol{\theta}) = \Theta^{(2)}\sigma\left(\Theta^{(1)}\mathbf{z} + \mathbf{b}\right),$$

where $\Theta^{(1)}, \mathbf{z} \in \mathbb{R}^d, \boldsymbol{\Theta}^{(2)}, \mathbf{b} \in \mathbb{R}^N$, are dense in $C(I^d)$, that is, given any function $f \in C(I^d)$ and $\epsilon > 0$, there exists $f_{\sigma,NN}$ for which

$$|f_{\sigma,NN}(\mathbf{z};\boldsymbol{\theta}) - f(\mathbf{z})| < \epsilon, \quad \text{for all} \quad \mathbf{x} \in I^d.$$

- This result proves that any continuous function can be **approximated** using a feedforward neural network with a single hidden layer if we use a given continuous activation function.

Universal family of models



Results on approximation error bounds

Theorem (Upper bound result by Mhaskar (1996) [Mha96])

Assume $\sigma : \mathbb{R} \to \mathbb{R}$ is such that σ has arbitrary order derivatives in an open interval I, and that σ is not a polynomial on I. Then, for any $p \in [1, \infty)$, $d \geq 2$ and $m \geq 1$, there exists $C_{d,m,p} > 0$ such that

$$\sup_{f \in \mathcal{F}} \inf_{f_{\sigma,NN} \in \mathcal{H}_N} \|f - f_{\sigma,NN}\|_p \leq C_{d,m,p} N^{-m/d}.$$

Theorem (Lower bound result by Maiorov, Meir (1996) [MM99])

Let $p, m \in \mathbb{N}^+, \geq 2$. If the activation function is the standard sigmoid function $\sigma(t) = 1/(1 + \exp(-t))$, then there exists $C'_{d,m,p} > 0$ such that

$$\sup_{f \in \mathcal{F}} \inf_{f_{\sigma,NN} \in \mathcal{H}_N} \|f - f_{\sigma,NN}\|_p \geq C'_{d,m,p} (N \log(N))^{-m/d},$$

where $\log(N)$ can be removed under additional continuity assumptions.

Q: Can this curse of dimensionality be fought? A.: yes, in different spaces

Barron's remedy against curse of dimensionality

Let the function $f : \mathbb{R}^d \to \mathbb{R}$ has a Fourier representation

$$f(\mathbf{z}) = \int_{\mathbb{R}^d} \exp(i\langle \boldsymbol{\omega}, \mathbf{z}\rangle)\tilde{f}(\boldsymbol{\omega})d\boldsymbol{\omega}, \tag{12}$$

where $\tilde{f}(\boldsymbol{\omega}) \in \mathbb{C}$. Assume that $f(\mathbf{0}) = 0$ and that

$$C_f = \int_{\mathbb{R}^d} \|\boldsymbol{\omega}\|_2 |\tilde{f}(\boldsymbol{\omega})| d\boldsymbol{\omega} < \infty,$$

then the Barron's result holds

Theorem

Fix a $C > 0$ and an arbitrary probability measure μ on the unit ball $\mathbb{B}^d \subset \mathbb{R}^d$. For every function f with $C_f < C$ and every $N \in \mathbb{N}^+$, there exists some $f_{\sigma,NN} \in \mathcal{H}_N$ such that

$$\left[\int_{\mathbb{B}^d} (f(\mathbf{z}) - f_{\sigma,NN}(\mathbf{z}))^2 \mu(d\mathbf{z})\right]^{1/2} \leq \frac{2C}{\sqrt{N}}.$$

Moreover, the weights of $f_{\sigma,NN}$ may be restricted to satisfy that their 1-norm is less than $2C$.

Barron's remedy against curse of dimensionality

Intuition based on Monte Carlo simulations:
Monte Carlo approximation errors are independent of dimensionality in the evaluation of high-dimensional integrals. Let us generate $\{\boldsymbol{\omega}_j\}_{1\le j\le N}$ randomly from a given density $p(\cdot)$ in $\mathbb{R}^d$. Consider the approximation to (12) by

$$f_{\sigma,NN}(\mathbf{z}) = \frac{1}{N}\sum_{j=1}^{N}\theta_j \exp(i\langle\boldsymbol{\omega}_j,\mathbf{z}\rangle), \quad \theta_j = \frac{\tilde{f}(\boldsymbol{\omega}_j)}{p(\boldsymbol{\omega}_j)}.$$

Then $f_{\sigma,NN}(\mathbf{z})$ is a one-hidden-layer neural network with N units and the sinusoid activation function. Note that $\mathbb{E}[f_{\sigma,NN}(\mathbf{z})] = f(\mathbf{z})$ with respect to the law of $\{\boldsymbol{\omega}_j\}$. By independence,

$$\mathbb{E}[(f_{\sigma,NN}(\mathbf{z}) - f(\mathbf{z}))^2] = \frac{1}{N}\mathrm{Var}(\theta_j \exp(i\langle\boldsymbol{\omega}_j,\mathbf{z}\rangle)) \le \frac{1}{N}\mathbb{E}[\theta_j^2]$$

if $\mathbb{E}[\theta_j^2] < \infty$. The rate is independent of the dimensionality d, though the constant can be.

Outline for section 4

Regression with artificial neural networks

- Let $f_{\boldsymbol{\Theta}}(Z) := f(Z, \boldsymbol{\theta})$.
- L is the **number of layers** of the networks, s_l is the **number of neurons** in layer l without bias.
- K, dimension of the target, corresponds to the number of neurons in the last (output) layer.
- $D_n := \{(Z_i, Y_i)\}_{i \in \{1,\dots,n\}}$, $Z_i \in \mathcal{Z}$, $Y_i \in \mathcal{Y}$, with $(Z_i, Y_i) \overset{IID}{\sim} \mathbb{P}_{ZY}$ is the training sample of length n.

Regression with artificial neural networks

In the regression setup it is common to use the (regularized) mean square error as empirical error:

$$\hat{R}_n(f) = \frac{1}{2n} \sum_{i=1}^{n} (Y_i - f_{\boldsymbol{\Theta}}(Z_i)))^2$$

$$\hat{\boldsymbol{\Theta}}_{n,\lambda} = \arg\min_{\boldsymbol{\Theta}} \{\hat{R}_n(f) + \underbrace{\frac{\lambda}{2} \sum_{l=1}^{L-1} \underbrace{\sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (\Theta_{j,i}^{(l)})^2}_{\|\Theta^{(l)}\|_F^2}\}}_{\ell_p \text{ regularization: here } p=2}$$

Neural networks with random weights

FFNN architecture with random inner weights (Extreme Learning Machines):

$$\mathbf{y} = f_{\sigma,NN}(\mathbf{z}) = \Theta^{(2)}\sigma(\Theta^{(1)}\mathbf{z} + \mathbf{b}).$$

where both $\Theta^{(1)}$ and $\mathbf{b}$ are randomly sampled.
In the notebook:

$$h(\mathbf{z}; \mathbf{W}) = \sum_{j=1}^{K} W_j \sigma\left(\mathbf{A}_j \odot \mathbf{z} + \zeta_j\right)$$

$$\widehat{\mathbf{W}} := \arg\min_{\mathbf{W}\in\mathbb{R}^K} \sum_{i=1}^{n} \left(Y_i - h(Z_i; \mathbf{W})\right)^2 + \lambda\|\mathbf{W}\|_2^2$$

Let $\mathbf{X}_i := \sigma(\mathbf{A}_j \odot \mathbf{Z}_i + \boldsymbol{\zeta})$ and $Z := (\mathbf{X}_1, \ldots, \mathbf{X}_n) \in \mathbb{R}^{k\times n}$, then

$$\widehat{\mathbf{W}} = (Z'Z + \lambda I)^{-1} Z'\mathbf{Y}.$$

The RWNN estimate is given by:

$$\widehat{h}(Z) := \sum_{j=1}^{K} \widehat{W}_j \sigma\left(\mathbf{A}_j \odot \mathbf{x} + \zeta_j\right)$$

Approximation guarantees

For any $\rho > 1, R > 0$ consider the following random sampling scheme:

- let $\mathbf{A}_1, \mathbf{A}_2, \ldots$ be i.i.d. drawn from the unif. distr. on the ball $B_R \subset \mathbb{R}^q$
- let $\zeta_1, \zeta_2, \ldots$ be i.i.d. unif. distr. $[-\rho, \rho]$, independent of $\{\mathbf{A}_i\}_{i \in \mathbb{N}^+}$

Corollary

Let μ be a probability measure on $\mathbb{R}^q$, $f \in L^2(\mathbb{R}^q, \mu)$ and let $\sigma : \mathbb{R} \to \mathbb{R}$ be given as $\sigma(x) = \max(x, 0)$. Then for any $\varepsilon > 0, \delta \in (0, 1)$ there exist $N \in \mathbb{N}^+, R > 0$, $\rho > 1$ and real-valued random variables $W_1, \ldots, W_N$ s.t. the random feedforward neural network (with "inner weights" $(\mathbf{A}_1, \zeta_1), (\mathbf{A}_2, \zeta_2), \ldots$) specified by

$$h(\mathbf{z}; \mathbf{W}) = \sum_{j=1}^{K} W_j \sigma(\mathbf{A}_j \odot \mathbf{z} + \zeta_j), \quad \mathbf{z} \in \mathbb{R}^q$$

approximates f in $L^2(\mathbb{R}^q, \mu)$ up to precision ε with probability $1 - \delta$, that is,

$$\int_{\mathbb{R}^q} |f(\mathbf{z}) - h(\mathbf{z}; \mathbf{W})|^2 \mu(\mathrm{d}\mathbf{z}) < \varepsilon^2$$

Outline for section 5

Learning in static scenarios

- **We learn**

  Functions

- **Out of**

  data observations

Our goal is to model **function relations** between the input **z** and the output **y**

$$\mathbf{y} = f^*(\mathbf{z})$$

Learning of dynamic processes

- **We learn**

$$\text{Dynamic Processes} \left\{ \begin{array}{l} \text{Input/Output systems} \\ \text{Dynamical systems} \end{array} \right\} \text{In Discrete Time}$$

- **Out of**

Low-dimensional discrete-time data observations

Our goal is to model **causal functional relations** between the input $\mathbf{z}$ and the output $\mathbf{y}$ sequences, that is,

$$\mathbf{y}_t = H(\ldots, \mathbf{z}_{t-2}, \mathbf{z}_{t-1}, \mathbf{z}_t), \quad \text{for all} \quad t,$$

or **causal filters**

$$\mathbf{y} = U(\mathbf{z})$$

What families of I/O systems can we provably proxy?

- Fading memory functionals/filters

$$\ldots, z_{-100}, z_{-99}, z_{-98}, \ldots, z_{-50}, z_{-49}, z_{-48}, \ldots, z_{-2}, z_{-1}, z_0 \overset{H}{\mapsto} y_0$$

- L^p functionals/filters in generative setting

$$\ldots, \varepsilon_{-2}, \varepsilon_{-1}, \varepsilon_0 \overset{H}{\mapsto} y_0$$

Direct p-step forecasting of deterministic series

Definition

For any $\tau \in \mathbb{Z}_-$ $T_{-\tau} : (\mathbb{R}^d)^{\mathbb{Z}_-} \longrightarrow (\mathbb{R}^d)^{\mathbb{Z}_-}$ is the *time delay* operator defined by $T_{-\tau}(\mathbf{Z})_t := \mathbf{Z}_{t+\tau}$ for any $t \in \mathbb{Z}_-$.

Input $\mathbf{Z}_t$

Target $\mathbf{Y}_t = \mathbf{Z}_{t+h} = (T_{-h}(\mathbf{Z}))_t$

Goal: learn functional $H\colon (D_d)^{\mathbb{Z}_-} \longrightarrow (D_d)^{\mathbb{Z}_-}$, $D_d \subset \mathbb{R}^d$ which is given by $H(\mathbf{Z}) = p_t(T_{-p}(\mathbf{Z})) = p_{-h}(T_t(\mathbf{Z})) = (T_{t-h}(\mathbf{Z}))_0$ where for any $\tau \in \mathbb{Z}_-$ $p_\tau : (D_d)^{\mathbb{Z}_-} \longrightarrow D_d$ is the projection given by $p_\tau(\mathbf{Z}) = \mathbf{Z}_\tau$

Direct p-step forecasting of stochastic series

Assumption (GDP)

Let $\left\{\left(Y_t, \mathbf{Z}_t^\top\right)^\top\right\}_{t=1}^{\infty}$ *be a covariance-stationary stochastic process taking values on* $\mathbb{R}^{d+1}$.

For (usually predetermined) integers $h \geq 1$ and $r \geq 0$ define the M-dimensional vector of predictors $\mathbf{X}_t := \left(Y_{t-1}, \ldots, Y_{t-h}, \mathbf{Z}_t^\top, \ldots, \mathbf{Z}_{t-r}^\top\right)^\top$ where $M = p + d(r+1)$ and consider the following direct forecasting model:

$$Y_{t+h} = f_h\left(\mathbf{X}_t\right) + U_{t+h}, \quad h = 1, \ldots, H, \quad t = 1, \ldots, T$$

where $f_h : \mathbb{R}^M \to \mathbb{R}$ is an unknown (measurable) function and $U_{t+h} := Y_{t+h} - f_h\left(\mathbf{X}_t\right)$ is assumed to be zero mean and finite variance.

Direct p-step forecasting of stochastic series

This approach corresponds to learning a function. Can you think of turning it into a dynamic problem of learning a functional?

Depending on which approach we take, we either use static or recurrent networks for handling the task.

Outline for section 6

References I

Andreas Christmann, Ingo Steinwart, and Andreas Christmann.

Support Vector Machines.
Springer, 2008.

G. Cybenko.

Approximation by superpositions of a sigmoidal function.
Mathematics of Control, Signals, and Systems, 2(4):303–314, dec 1989.

N. Hrushikesh Mhaskar.

Neural networks for optimal approximation of smooth and analytic functions.
Neural computation, 8(1):164–177, 1996.

T. M. Mitchell.

Machine Learning.
McGraw-Hill, 1997.

V. Maiorov and Ron Meir.

On the near optimality of the stochastic approximation of smooth functions by neural networks.
Advances in Computational Mathematics, 13(1):79–103, 1999.

Robert Nowak.

Statistical Learning Theory: Lecture Notes, 2009.

David Rosenberg.

Loss Functions for Regression and Classification: Lecture Notes, 2015.

Vladimir Vapnik.

Statistical Learning Theory.
Wiley, 1998.

References II

Yihong Wu.
Information-theoretic Methods in High-dimensional Statistics: Lecture Notes, 2016.