Statistical Learning Theory

Imma Valentina Curato

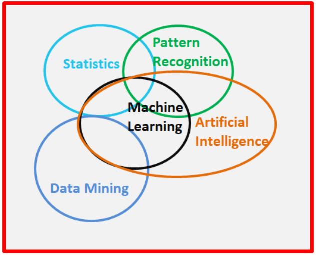
Workshop and Summer School on Applied Analysis 2023

Looking for answers

- What is the relationship between **Machine Learning** and **Statistics**?
- What does it mean **Statistical Learning Theory**?

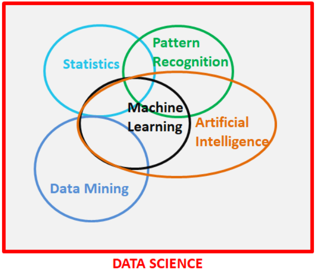
What is your take?

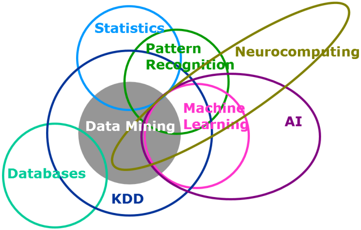
Let it be chaos!



DATA SCIENCE

Let it be chaos!





Results obtained by googling "Statistical foundation of machine learning".

What is Machine Learning?

- Machine learning is an interdisciplinary field focusing on both the mathematical foundations and practical applications of systems that learn, reason, and act. In practice, it is the process of converting experience (e.g. data) into expertise or knowledge (e.g., with a model).

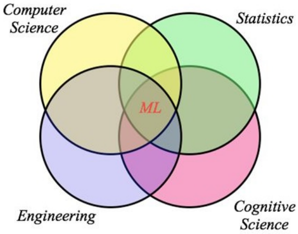


Figure: From "Machine Learning and Information Retrieval", Zoubin Ghahramani, University of Cambridge.

Machine Learning Paradigms

- **Supervised learning** is about learning to predict from examples (input-output pairs) of correct predictions.
- **Unsupervised learning** is about modeling unlabeled data. Clustering, dimensionality reduction, missing value synthesis, and anomaly detection are typical tasks for unsupervised learning.
- **Reinforcement learning** is about agents learning by themselves how to behave in their environments.
- **Phyics informed machine learning** focuses on ways to combine prior knowledge in the form of physical laws and equations with machine learning.

Learning paradigms can be used in conjunction

- **Semi-supervised learning** is about learning from supervised and unsupervised data.
- **Online learning** is about continuously learning from a stream of data.
- **Active learning** is about learning from a teacher by asking questions.
- **Transfer learning** is about transferring knowledge from one learning task to another learning task.

What is Statistics?

The objective of statistics is the understanding of **information** contained in the data.

- **Descriptive Statistics**: includes methods for organizing, and summarizing data.
- **Inductive Statistics** (or inferential statistics): deals with taking **observations** out of a **larger population** and using that data to draw conclusions, make decisions, forecasts.

Machine learning & Statistics

Different interactions exists between machine learning (in all its paradigms) and statistical tools.

Machine learning & Statistics

Different interactions exists between machine learning (in all its paradigms) and statistical tools.

- Both Statistics and Machine Learning are working under a set of assumpetions.
- The first of which is that the data follow a **Statistical Model**.

Data generating process

Let $z = (z_1, \dots, z_N)$ be an observed data set. We always *implicitly* put an assumption on their data generation, i.e. z is a realization out of a random vector (probabilistic model) $\boldsymbol{Z} = (\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_N)$. More precisely,

$$\boldsymbol{Z} : (\Omega, \mathcal{F}, P) \to (\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$$

and we call $\mathbb{P}_{\boldsymbol{Z}}$ the (unfortunately never known) distribution of the data defined as

$$\mathbb{P}_{\boldsymbol{Z}}(A) = P \circ \boldsymbol{Z}^{-1}(A)$$

for all $A \in \mathcal{B}(\mathbb{R}^n)$.

Data generating process

The vector z has different names in the literature. For example,

- in **Supervised Learning**, we call them *examples*, and typically $z_i = (x_i, y_i)$ are input-output pairs. z is also called a *training data set*;
- whereas in **Statistics** we call them *observations* or *samples*.

Independent and Dependent Set-Ups

There are different assumptions that can be put on $\boldsymbol{Z}$.

1. The $\boldsymbol{Z}_i$s are i.i.d (independent and identically distributed data)
2. the $\boldsymbol{Z}_i$s are identically distributed (stationary data)
3. non stationary data (serially correlated data)

Independent and Dependent Set-Ups

There are different assumptions that can be put on $\boldsymbol{Z}$.

1. The $\boldsymbol{Z}_i$s are i.i.d (independent and identically distributed data)
2. the $\boldsymbol{Z}_i$s are identically distributed (stationary data)
3. non stationary data (serially correlated data)

$\longrightarrow$

1. Standard machine learning (e.g. image analysis)
2. Sequential data, time series, spatio-temporal data
3. All the rest.... (e.g. data set used in weather forecast)

Independent and Identically Distributed Categorical Data

Two random element $\boldsymbol{Z}_i$ and $\boldsymbol{Z}_j$ admitting finite values in $\mathcal{Z}$, what we call a **categorical variable**:

- we call them **independent** if

  $$P(\boldsymbol{Z}_i = z_i, \boldsymbol{Z}_j = z_j) = P(\boldsymbol{Z}_i = z_i)P(\boldsymbol{Z}_j = z_j).$$

  The above is equivalent to

  $$P(A \cap B) = P(A)P(B)$$

  where
  $A = \{\omega \in \Omega : \boldsymbol{Z}_i(\omega) = z_i\}, B = \{\omega \in \Omega : \boldsymbol{Z_j}(\omega) = z_j\} \subset \mathcal{F}$.
- and **identically distributed** if all the $\boldsymbol{Z}_i \overset{d}{\sim} \mathbb{P}_{\boldsymbol{Z}_1}$ for all i. The distribution $\mathbb{P}_{\boldsymbol{Z}} = \mathbb{P}^N_{\boldsymbol{Z}_1}$.

Independent and Identically Distributed Quantitative Data

Two random element $\boldsymbol{Z}_i$ and $\boldsymbol{Z}_j$ with values in $\mathbb{R}$, what we call a **quantitative variable**:

- we call them **independent** if the $\sigma(\boldsymbol{Z}_i)$ and $\sigma(\boldsymbol{Z}_j)$ are independent, i.e., for all $A \in \sigma(\boldsymbol{Z}_i)$ and $B \in \sigma(\boldsymbol{Z}_j)$ then

$$P(A \cap B) = P(A)P(B)$$

- and **identically distributed** if all the $\boldsymbol{Z}_i \overset{d}{\sim} \mathbb{P}_{\boldsymbol{Z}_1}$ for all i. The distribution $\mathbb{P}_{\boldsymbol{Z}} = \mathbb{P}_{\boldsymbol{Z}_1}^{N}$.

Dependent random variables

- There are many ways of modeling dependence, all the models analyzed in stochastic processes and random fields courses serve to this aim. $\boldsymbol{Z}$ is a *sample* out of a stochastic process $(\boldsymbol{Z}_t)_{t\in\mathbb{R}}$. Examples: martingales, Markov chains, etc.
- The distribution $\mathbb{P}_{\boldsymbol{Z}}$ is a finite dimensional marginal of the stochastic process $(\boldsymbol{Z}_t)_{t\in\mathbb{R}}$.

Stationarity

This always refer to the stochastic process underlying the data we are modeling

- We say that a process is **strictly stationary** is for every $s \in \mathbb{N}$ and $\tau \in \mathbb{R}$

  $$P(\boldsymbol{Z}_{t_1}, \ldots, \boldsymbol{Z}_{t_s}) = P(\boldsymbol{Z}_{t_1+\tau}, \ldots, \boldsymbol{Z}_{t_s+\tau})$$

  for all $t_1, \ldots, t_s \in \mathbb{R}$
- We say that a process is **weakly stationary:** if $\mathbb{E}[\boldsymbol{Z}_t^2] < \infty$ for all t, $\mathbb{E}[\boldsymbol{Z}_t]$ is constant w.r.t. t and $Cov(\boldsymbol{Z}_t, \boldsymbol{Z}_{t+h}) = Cov(\boldsymbol{Z}_0, \boldsymbol{Z}_h)$ for all $t, h \in \mathbb{R}$ (reminder: $Cov(\boldsymbol{X}, \boldsymbol{Z}) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$).

Statistical Model

Let us consider a data generating process $\boldsymbol{Z = (Z_1, \ldots, Z_N)}$. A (parametric) statistical model is a set of probability distribution functions on $\mathcal{B}(\mathbb{R}^n)$, i.e. $\mathcal{P} := \{\mathbb{P}_{\boldsymbol{Z}}(\theta) : \text{where}\, \theta \in \Theta\}$. This means that we assume to know $\mathbb{P}_{\boldsymbol{Z}}$ apart for the value of a parameter $\theta \in \Theta \subset \mathbb{R}^k$

Statistical Model

Let us consider a data generating process $\boldsymbol{Z} = (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_N)$. A (parametric) statistical model is a set of probability distribution functions on $\mathcal{B}(\mathbb{R}^n)$, i.e. $\mathcal{P} := \{\mathbb{P}_{\boldsymbol{Z}}(\theta) : \text{where}\, \theta \in \Theta\}$. This means that we assume to know $\mathbb{P}_{\boldsymbol{Z}}$ apart for the value of a parameter $\theta \in \Theta \subset \mathbb{R}^k$

- Statistical analysis (inductive) is used to infer the parameters of the distribution underlying the data by using **statistical estimators**.
- Machine learning starts from a statistical model, defined accordingly to the paradigms. Then data are used to infer the parameters of it.

Standard Example in Statistics: Gaussian family distribution

Z is i.i.d.. In this case, it is enough to model the probability distribution of just one $\boldsymbol{Z}_i$s, we assume that its distribution belongs to the Gaussian parametric family, i.e., what we call the *density*

$$\mathcal{P} := \{f_{\mu,\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}}\mathrm{e}^{\frac{(x-\mu)^2}{2\sigma^2}}, \;\; \mu \in \mathbb{R}, \sigma^2 > 0\}$$

Standard Example in Machine Learning: Supervised Learning

- We assume throughout that $\boldsymbol{Z}$ is i.i.d. (but of course this can be generalized to dependent set-ups) and focus on **Supervised Learning**.

Standard Example in Machine Learning: Supervised Learning

- We assume throughout that $\boldsymbol{Z}$ is i.i.d. (but of course this can be generalized to dependent set-ups) and focus on **Supervised Learning**.

A supervised learning model is a statistical estimator

The umbrella data set

Let us consider data on the number of umbrellas sold by a Sachsen's shop based on the amount of rainfall in the years 2018-2019. (We could, in principle, have had different realizations of rainfall and umbrella sold but the state of the *world* happen to give us the below meteorological state). We assume $\boldsymbol{Z}$ is i.i.d.

	2018			2019	
Month	Rainfall(mm)	Umbrellas sold	Month	Rainfall(mm)	Umbrellas sold
Jan	82	15	Jan	87	14
Feb	92.5	25	Feb	97.5	27
Mar	83.2	17	Mar	88.2	14
Apr	97.7	28	Apr	102.7	30
May	131.9	41	May	123	43
Jun	141.3	47	Jun	146.3	49
Jul	165.4	50	Jul	160	49
Aug	140	46	Aug	145	44
Sep	126.7	37	Sep	131.7	39
Oct	97.8	22	Oct	118	36
Nov	86.2	20	Nov	91.2	20
Dec	99.6	30	Dec	104.6	32

Analysis of Umbrellas data set

Statistical Analysis

- Estimate the average amount of umbrellas sold and rainfall (descriptive statistics)

Analysis of Umbrellas data set

Statistical Analysis

- Estimate the average amount of umbrellas sold and rainfall (descriptive statistics)
- Estimate the variability of the umbrellas sold and rainfall across the years (descriptive statistics)
- Estimate the parameters of a statistical model (inductive statistics)

Analysis of Umbrellas data set

Statistical Analysis

- Estimate the average amount of umbrellas sold and rainfall (descriptive statistics)
- Estimate the variability of the umbrellas sold and rainfall across the years (descriptive statistics)
- Estimate the parameters of a statistical model (inductive statistics)

Supervised Learning

- Defining a model that seeks to predict how many umbrellas will be sold based on the amount of rainfall.

Linear Predictor. $y_i = \beta_0 + \beta x_i + \epsilon_i$

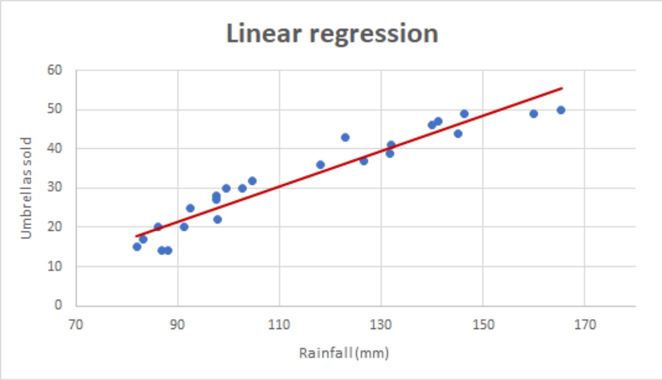


Figure: Simple linear regression: the number of umbrella sold represents realization of the output variable and the amount of rainfall a realization of the input one.

Statistical Estimators

An estimator for a parameter θ is defined as a function of the sample $\boldsymbol{Z}$:

$$\hat{\theta}_{\boldsymbol{Z}} = g(\boldsymbol{Z})$$

Statistical Estimators

An estimator for a parameter θ is defined as a function of the sample $\boldsymbol{Z}$:

$$\hat{\theta}_{\boldsymbol{Z}} = g(\boldsymbol{Z})$$

Sample mean estimator

$$\hat{\mu}_{\boldsymbol{Z}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{Z}_i$$

Statistical Estimators

An estimator for a parameter θ is defined as a function of the sample $\boldsymbol{Z}$:

$$\hat{\theta}_{\boldsymbol{Z}} = g(\boldsymbol{Z})$$

Sample mean estimator

$$\hat{\mu}_{\boldsymbol{Z}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{Z}_i$$

Sample standard deviation

$$\hat{\sigma}_{\boldsymbol{Z}} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\boldsymbol{Z}_i - \hat{\mu}_{\boldsymbol{Z}})^2}$$

Statistical Estimators

An estimator for a parameter θ is defined as a function of the sample $\boldsymbol{Z}$:

$$\hat{\theta}_{\boldsymbol{Z}} = g(\boldsymbol{Z})$$

Sample mean estimator

$$\hat{\mu}_{\boldsymbol{Z}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{Z}_i$$

Sample standard deviation

$$\hat{\sigma}_{\boldsymbol{Z}} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (\boldsymbol{Z}_i - \hat{\mu}_{\boldsymbol{Z}})^2}$$

- We can use $\hat{\mu}_z$ and $\hat{\sigma}_z$ to get the average number of umbrellas sold and the variability of their sales across the years, respectively, for a given realization z
- **Umbrellas data sets:**

$$\hat{\mu}_z = 32.29167$$

$$\hat{\sigma}_z = 12.16724$$

Estimators properties

Accuracy:

$$BIAS(\hat{\theta}_{\boldsymbol{Z}}, \theta) = \mathbb{E}_{\boldsymbol{Z}}[\hat{\theta}_{\boldsymbol{Z}}] - \theta$$

We have then that the sample mean and variance estimator are unbiased, if $BIAS(\hat{\theta}_{\boldsymbol{Z}}, \theta) = 0$.

Estimators properties

Accuracy:

$$BIAS(\hat{\theta}_{\boldsymbol{Z}}, \theta) = \mathbb{E}_{\boldsymbol{Z}}[\hat{\theta}_{\boldsymbol{Z}}] - \theta$$

We have then that the sample mean and variance estimator are unbiased, if $BIAS(\hat{\theta}_{\boldsymbol{Z}}, \theta) = 0$.

Precision:

$$VAR(\hat{\theta}_{\boldsymbol{Z}}) = \mathbb{E}_{\boldsymbol{Z}}[(\hat{\theta}_{\boldsymbol{Z}} - \mathbb{E}_{\boldsymbol{Z}}[\hat{\theta}_{\boldsymbol{Z}}])^2] = \mathbb{E}_{\boldsymbol{Z}}[\hat{\theta}_{\boldsymbol{Z}}^2] - \mathbb{E}_{\boldsymbol{Z}}[\hat{\theta}_{\boldsymbol{Z}}]^2$$

Estimators properties

Accuracy:

$$BIAS(\hat{\theta}_{\boldsymbol{Z}}, \theta) = \mathbb{E}_{\boldsymbol{Z}}[\hat{\theta}_{\boldsymbol{Z}}] - \theta$$

We have then that the sample mean and variance estimator are unbiased, if $BIAS(\hat{\theta}_{\boldsymbol{Z}}, \theta) = 0$.

Precision:

$$VAR(\hat{\theta}_{\boldsymbol{Z}}) = \mathbb{E}_{\boldsymbol{Z}}[(\hat{\theta}_{\boldsymbol{Z}} - \mathbb{E}_{\boldsymbol{Z}}[\hat{\theta}_{\boldsymbol{Z}}])^2] = \mathbb{E}_{\boldsymbol{Z}}[\hat{\theta}^2_{\boldsymbol{Z}}] - \mathbb{E}_{\boldsymbol{Z}}[\hat{\theta}_{\boldsymbol{Z}}]^2$$

Bias-Variance Trade off:

$$MSE = \mathbb{E}_{\boldsymbol{Z}}[(\hat{\theta}_{\boldsymbol{Z}} - \theta)^2] = BIAS^2(\hat{\theta}_{\boldsymbol{Z}}, \theta) + VAR(\hat{\theta}_{\boldsymbol{Z}})$$

Fundamental Notations

- A loss function $L : \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ that evaluate the distance, or error, between two elements in the output domain. For example, regression tasks have $\mathcal{Y} = \mathbb{R}, \mathbb{R}^k$ and binary classifications problems $\mathcal{Y} = \{0, 1\}$.
- We observe a training data set $z = ((x_1, y_1), \dots (x_N, y_N))$ which is a realization from $\boldsymbol{Z} := \{(\boldsymbol{X}_1, \boldsymbol{Y}_1), \dots, (\boldsymbol{X}_N, \boldsymbol{Y}_N)\}$ which is i.i.d. (generalization of course exist also for the dependent case) having an unknown distribution $\mathbb{P}_{\boldsymbol{Z}}$.
- In supervised learning we determine a model h by minimizing the training error

$$\mathcal{E}_{train}(h) = \frac{1}{N} \sum_{i=1}^{N} L(h(x_i), y_i),$$

where h belongs to a family $\mathcal{H}$ of predictors (also called learning algorithm) as, for example, linear functions, neural network, kernel methods, etc. We call such model **empirical risk minimizer**.

The ERM is a statistical estimator!

The predictor we select is a "random one", because it is going to depend on the data we observe, i.e. to different draws z will correspond a different predictor h_z:

$$h_z^{ERM} := arg \min_{h \in \mathcal{H}} \mathcal{E}_{train}(h).$$

is a realization of the statistical estimator

$$h_{\boldsymbol{Z}}^{ERM} := arg \min_{h \in \mathcal{H}} \mathcal{E}_{train}(h) = arg \min_{h \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} L(h(\boldsymbol{X}_i), \boldsymbol{Y}_i).$$

The ERM is a statistical estimator!

The predictor we select is a "random one", because it is going to depend on the data we observe, i.e. to different draws z will correspond a different predictor h_z:

$$h_z^{ERM} := arg \min_{h \in \mathcal{H}} \mathcal{E}_{train}(h).$$

is a realization of the statistical estimator

$$h_{\boldsymbol{Z}}^{ERM} := arg \min_{h \in \mathcal{H}} \mathcal{E}_{train}(h) = arg \min_{h \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} L(h(\boldsymbol{X}_i), \boldsymbol{Y}_i).$$

The ERM estimates the parameter of a statistical model? If yes, which one?

Regression Task

- We observe the training data set z which is a realization from $\boldsymbol{Z} = ((\boldsymbol{X}_1, \boldsymbol{Y}_1), \ldots, (\boldsymbol{X}_N, \boldsymbol{Y}_N))$ i.i.d. The $\boldsymbol{X}_i$s have values in $\mathbb{R}^p$ and the $\boldsymbol{Y}_i$s in $\mathbb{R}$. Moreover, we assume to work with the squared loss.
- For all i, we assume the *population model*

  $$\boldsymbol{Y}_i = h(\boldsymbol{X}_i) + \epsilon_i, \text{ where } (\epsilon_i)_{i \in \mathbb{N}} \text{ is i.i.d.}$$

  where $h(x) := \mathbb{E}[\boldsymbol{Y}_1 | \boldsymbol{X}_1 = x]$.
- Different type of functions can be used to model the conditional expectation above.

Regression Task

- In a linear regression model, we model the conditional expectation using a linear function $h_{\boldsymbol{Z}}(\cdot) := \hat{\beta}_0 + \hat{\beta}(\cdot)$, where $(\hat{\beta}_0, \hat{\beta})$ is the OLS estimator. Being $\mathbb{E}[\boldsymbol{Y}_1|\boldsymbol{X}_1]$ a random vector, what the estimator really does is determining the parameters of the statistical model $\mathcal{P} := \{\mathbb{P}_{\boldsymbol{Y}_1|\boldsymbol{X}_1}(\theta) : \theta \in \Theta\}$.
- When using a neural network $\mathbb{E}[\boldsymbol{Y}_1|\boldsymbol{X}_1]$ is modeled by using a non linear function, where the parameters are obtained by using a gradient descent method (for different realizations, we obtain different results, that means that in our framework we can also consider the output of such algorithm random!). Also in this case, we determine the parameters of the statistical model $\mathcal{P} := \{\mathbb{P}_{\boldsymbol{Y}_1|\boldsymbol{X}_1}(\theta) : \theta \in \Theta\}$.

Target of a supervised learning algorithm

We want to build a model that has good generalization performance, i.e. it is capable of making good predictions for data not belonging to the training data set.

Target of a supervised learning algorithm

We want to build a model that has good generalization performance, i.e. it is capable of making good predictions for data not belonging to the training data set.

$$\downarrow$$

We need to evaluate the generalization error of a predictor.

Generalization error

The generalization performance of a function $h \in \mathcal{H}$ is evaluated using the **test error or generalization error.** In the i.i.d. setup, this can be evaluated by focusing on

$$\mathcal{E}_{test}(h) = \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})}[L(h(\boldsymbol{X}), \boldsymbol{Y})],$$

where $(\boldsymbol{X}, \boldsymbol{Y}) \sim \mathbb{P}_{(X_1,Y_1)}$ and it is an independent new *test example*.

Generalization error

The generalization performance of a function $h \in \mathcal{H}$ is evaluated using the **test error or generalization error.** In the i.i.d. setup, this can be evaluated by focusing on

$$\mathcal{E}_{test}(h) = \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})}[L(h(\boldsymbol{X}), \boldsymbol{Y})],$$

where $(\boldsymbol{X}, \boldsymbol{Y}) \sim \mathbb{P}_{(X_1,Y_1)}$ and it is an independent new *test example*.

Ideally, we would like to choose the function h **that minimizes the test error, so we will be sure that our chosen model has the best generalization performance.**

Everything depends on the loss!

Regression Framework

- Under the squared loss, $h(x) := \mathbb{E}[\boldsymbol{Y}_1 | \boldsymbol{X}_1 = x]$
- Under the absolute loss, $h(x) := Median(\boldsymbol{Y}_1 | \boldsymbol{X}_1 = x)$

Everything depends on the loss!

Regression Framework

- Under the squared loss, $h(x) := \mathbb{E}[\boldsymbol{Y}_1|\boldsymbol{X}_1 = x]$
- Under the absolute loss, $h(x) := Median(\boldsymbol{Y}_1|\boldsymbol{X}_1 = x)$

Classification Framework

- Under the 0/1 Loss, we obtain the Bayes Classifier.

Generalization error of a learning algorithm

Evaluating the generalization performance of a predictor $h_{\boldsymbol{Z}}$, means to work with the following object

$$\mathcal{E}_{test}(h_{\boldsymbol{Z}}) = \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})}[L(h_{\boldsymbol{Z}}(\boldsymbol{X}), \boldsymbol{Y})|\boldsymbol{Z}],$$

which becomes a random element. We then work with the above formulation or, if we want to work with not-random objects,

$$\mathbb{E}_{\boldsymbol{Z}}[\mathcal{E}_{test}(h_{\boldsymbol{Z}})] := \mathbb{E}_{\boldsymbol{Z}}[\mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})}[L(h_{\boldsymbol{Z}}(\boldsymbol{X}), \boldsymbol{Y})|\boldsymbol{Z}]].$$

The latter is called **expected generalization error.**

Target of supervised learning with more mathematical details!

- Let us assume that the true model minimizing the generalization error exists and belong to the family $\mathcal{H}'$. However, we do not know this class.

Target of supervised learning with more mathematical details!

- Let us assume that the true model minimizing the generalization error exists and belong to the family $\mathcal{H}'$. However, we do not know this class.
- We then decide to determine a predictor in a class of functions known to us that we call $\mathcal{H}$. Typically $\mathcal{H} \subseteq \mathcal{H}'$.

Target of supervised learning with more mathematical details!

- Let us assume that the true model minimizing the generalization error exists and belong to the family $\mathcal{H}'$. However, we do not know this class.
- We then decide to determine a predictor in a class of functions known to us that we call $\mathcal{H}$. Typically $\mathcal{H} \subseteq \mathcal{H}'$.
- We want to find a predictor $h_{\mathbf{Z}}$ such that the excess risk is small

$$\underbrace{\mathcal{E}_{test}(h_{\mathbf{Z}}) - \inf_{h \in \mathcal{H}'} \mathcal{E}_{test}(h)}_{excess\ risk}$$

Excess risk decomposition: approximation-estimation trade off

$$\underbrace{\mathcal{E}_{test}(h_{\mathbf{z}}) - \inf_{h\in\mathcal{H}'} \mathcal{E}_{test}(h)}_{\textit{excess risk}} = \underbrace{\mathcal{E}_{test}(h_{\mathbf{z}}) - \inf_{h\in\mathcal{H}} \mathcal{E}_{test}(h)}_{\textit{estimation error}} + \underbrace{\inf_{h\in\mathcal{H}} \mathcal{E}_{test}(h) - \inf_{h\in\mathcal{H}'} \mathcal{E}_{test}(h)}_{\textit{approximation error}}$$

- **The estimation error** measures how much extra loss the data scientists suffer by choosing a predictor instead of the function h that minimizes the generalization error. In most of the statistical literature, $\inf_{h\in\mathcal{H}} \mathcal{E}_{test}(h)$ is called the *oracle*.
- **The estimation error** depends on the loss function employed in the learning problem at hand, the number of training examples, and by the notion of *complexity* of the set $\mathcal{H}$.
- **The approximation error** measures how closely the best possible function in $\mathcal{H}$ is near to the true model.
- Larger sets $\mathcal{H}$ lead to smaller approximation error but higher estimation error.

Understanding what is random!

- As the data distribution $\mathbb{P}_{\boldsymbol{Z}}$ is unknown the test error $\mathcal{E}_{test}(\cdot)$ cannot be computed, and so the estimation error and the approximation error are *not explicitly computable*.
- Nevertheless, the estimation error is used as a way to assess how well the chosen predictor performs. It is important to note that **the estimation error is random** because it is a function of $\boldsymbol{Z}$.
- On the other hand the **approximation error is deterministic**, as it simply deals with quantifying the error introduced by considering the family $\mathcal{H}$ instead of the family $\mathcal{H}'$.

Controlling the approximation error

- Controlling the approximation error and determining a suitable class $\mathcal{H}$ is purely a problem of functional and numerical analysis.
- Note that the restriction to the space $\mathcal{H}$ can be seen as embodying prior information on the learning system or as a form of **explicit regularization**. The methodology that add a penalty term to the training error are also considered *explicit regularization*. Some of them under *convexity assumptions* can be considered equivalent to restricting the minimization of $\mathcal{E}_{train}(h)$ to a set $\mathcal{H}$.
- An example of a problem where the approximation error is zero (no explicit regularization) is the linear regression with squared loss function leading to the OLS estimator. Here we can find a predictor in $\mathcal{H}'$.

Statistical Learning Theory

Statistical learning theory focuses on controlling the estimation error by establishing upper bounds of it!

Statistical Learning Theory

Statistical learning theory focuses on controlling the estimation error by establishing upper bounds of it! More precisely, we controll the estimation error by

- bounding its expectation (w.r.t. $\mathbb{P}_{\boldsymbol{Z}}$)
- or showing that the estimation error is small with high probability, i.e., with probability $1-\delta$ for a $\delta \in (0, 1)$.

Statistical Learning Theory

Statistical learning theory focuses on controlling the estimation error by establishing upper bounds of it! More precisely, we controll the estimation error by

- bounding its expectation (w.r.t. $\mathbb{P}_{\boldsymbol{Z}}$)
- or showing that the estimation error is small with high probability, i.e., with probability $1 - \delta$ for a $\delta \in (0, 1)$.

Knowing when the estimation error is minimal lead us to understanding how a given algorithm generalizes and to design new ones.

Statistical Learning Theory

Statistical learning theory focuses on controlling the estimation error by establishing upper bounds of it! More precisely, we controll the estimation error by

- bounding its expectation (w.r.t. $\mathbb{P}_{\boldsymbol{Z}}$)
- or showing that the estimation error is small with high probability, i.e., with probability $1-\delta$ for a $\delta \in (0, 1)$.

Knowing when the estimation error is minimal lead us to understanding how a given algorithm generalizes and to design new ones.

Mathematical Tools: Union bounds, non asymptotic law of large numbers, concentration inequalities, optimization theory, etc...

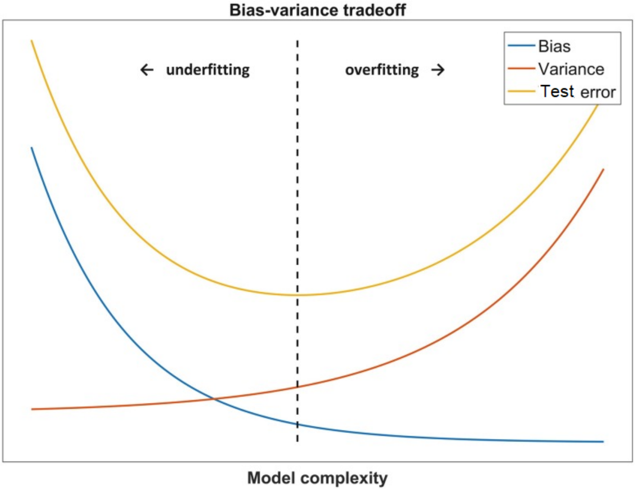
Linear regression setting with squared loss

To assess the generalization performance of the learning algorithm it is often used in machine learning the **bias-variance trade off** which is easy to compute in this context and corresponds to the expected excess risk (which is equal to the expected estimation error, i.e., $\mathcal{H} = \mathcal{H}'$).

$$\mathbb{E}_{\boldsymbol{Z}}[\mathcal{E}_{test}(h_{\boldsymbol{Z}}) - \inf_{h \in \mathcal{H}'} \mathcal{E}_{test}(h)] = \underbrace{\mathbb{E}_{\boldsymbol{Z}}[(\mathbb{E}[h_{\boldsymbol{Z}}(\boldsymbol{X})|\boldsymbol{X}] - \inf_{h \in \mathcal{H}'} \mathcal{E}_{test}(h))^2]}_{\textit{expected squared bias}} + \underbrace{\mathbb{E}_{\boldsymbol{Z}}[Var(h_{\boldsymbol{Z}}(\boldsymbol{X})|\boldsymbol{X})]}_{\textit{expected variance}}$$

In general, the larger the class $\mathcal{H}'$ is (i.e. the more flexible the methods we consider are), the higher the variance is and the smaller the bias is.

Bias-Variance Trade Off (underparameterized regime)



Statistical Learning Theory for the ERM (Statistical Aspects)

One of the firs question that statistical learning theory historically tried to find an answer for is **finding upper bounds for $h \in \mathcal{H}$ when $h_{\mathbf{z}}$ is the ERM.** We call it $h_{\mathbf{z}}^{ERM}$ in the following and $h^* := arg \inf_{h \in \mathcal{H}} \mathcal{E}_{test}(h)$.

Statistical Learning Theory for the ERM (Statistical Aspects)

One of the firs question that statistical learning theory historically tried to find an answer for is **finding upper bounds for $h \in \mathcal{H}$ when $h_{\boldsymbol{Z}}$ is the ERM.** We call it $h_{\boldsymbol{Z}}^{ERM}$ in the following and $h^* := arg \inf_{h \in \mathcal{H}} \mathcal{E}_{test}(h)$.

First of all, the estimation error can be further decomposed as:

$$\begin{aligned}
\underbrace{\mathcal{E}_{test}(h_{\boldsymbol{Z}}^{ERM}) - \mathcal{E}_{test}(h^*)}_{estimation\, error} =& \mathcal{E}_{test}(h_{\boldsymbol{Z}}^{ERM}) - \mathcal{E}_{train}(h_{\boldsymbol{Z}}^{ERM}) \\
&+ \underbrace{\mathcal{E}_{train}(h_{\boldsymbol{Z}}^{ERM}) - \mathcal{E}_{train}(h^*)}_{\leq 0} + \mathcal{E}_{train}(h^*) - \mathcal{E}_{test}(h^*) \\
&\leq \mathcal{E}_{test}(h_{\boldsymbol{Z}}^{ERM}) - \mathcal{E}_{train}(h_{\boldsymbol{Z}}^{ERM}) + \mathcal{E}_{train}(h^*) - \mathcal{E}_{test}(h^*)
\end{aligned}$$

The reality of it all!

- We assumed in the previous slide that we could compute the ERM but this rarely happens! In practice computing $h_{\mathbf{z}}^{ERM}$ in NP hard for most problem of interest, so we need a more flexible strategy to solve the problem
- Let us denote by $\hat{h}_{\mathbf{z}}^{ERM}$ an approximation to the ERM minimizer (e.g. think of $\hat{h}_{\mathbf{z}}^{ERM}$ as the output of a gradient descent algorithm run to minimize the training error over elements in $\mathcal{H}$). The approximation $\hat{h}_{\mathbf{z}}^{ERM}$ is also a random quantity, as it is a function of the data.

Statistical Learning Theory for the ERM (Optimization Aspects)

Let us consider the following new decomposition for the estimation error:

$$
\begin{aligned}
\underbrace{\mathcal{E}_{test}(\hat{h}_{\boldsymbol{z}}^{ERM}) - \mathcal{E}_{test}(h^*)}_{estimation\ error} =&\mathcal{E}_{test}(\hat{h}_{\boldsymbol{z}}^{ERM}) - \mathcal{E}_{train}(\hat{h}_{\boldsymbol{z}}^{ERM}) \\
&+ \mathcal{E}_{train}(\hat{h}_{\boldsymbol{z}}^{ERM}) - \mathcal{E}_{train}(h_{\boldsymbol{z}}^{ERM}) \\
&+ \underbrace{\mathcal{E}_{train}(h_{\boldsymbol{z}}^{ERM}) - \mathcal{E}_{train}(h^*)}_{\leq 0} + \mathcal{E}_{train}(h^*) - \mathcal{E}_{test}(h^*) \\
&\leq \underbrace{\mathcal{E}_{train}(\hat{h}_{\boldsymbol{z}}^{ERM}) - \mathcal{E}_{train}(h_{\boldsymbol{z}}^{ERM})}_{optimization} \\
&+ \underbrace{\mathcal{E}_{test}(\hat{h}_{\boldsymbol{z}}^{ERM}) - \mathcal{E}_{train}(\hat{h}_{\boldsymbol{z}}^{ERM}) + + \mathcal{E}_{train}(h^*) - \mathcal{E}_{test}(h^*)}_{statistics}
\end{aligned}
$$

About the optimization error when using stochastic gradient descent to compute interpolating solutions (overparameterized regime)

- We assume that $\mathcal{H} = \mathcal{H}'$. We use no *explicit regularization*, i.e. the estimation error coincides with the excess risk.
- Note that SGD includes sources of randomness in the problem other than $\boldsymbol{Z}$ and typically produce $\hat{h}_{\boldsymbol{Z}}$ such that $\mathcal{E}_{train}(\hat{h}_{\boldsymbol{Z}}) \simeq 0$ (**interpolation solution** as a form of **implicit regularization**).
- The very essence of the algorithm – as the learning rate, the size of the batch, the epochs you use, the flavor of SGD you decide to employ– enters in the determination of the optimization error.
- The right way to estimate the estimation error in such cases is a new and actively studied branch of statistical learning theory.

Bias-Variance trade off in the case of linear regressions

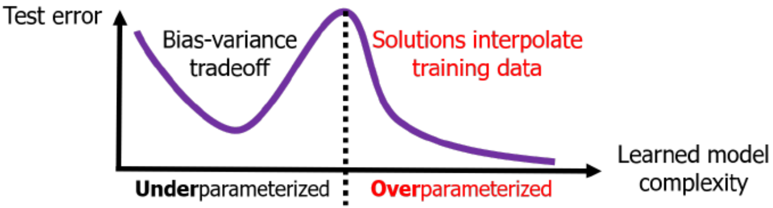


Figure: Source: "A Farewell to the Bias-Variance Tradeoff? An Overview of the Theory of Overparameterized Machine Learning", arXiv:2109.02355v1, Dar Y., Muthukumar V. and Baraniuk R.G.

This is just the beginning ...

- Algorithm Foundations of Learning, Patrick Rebeschini, University of Oxford, https://www.stats.ox.ac.uk/ rebeschi/teaching/AFoL/22/index.html
- SHALEV-SHWARTZ, S., AND BEN-DAVID, S. (2014). Understanding machine learning: from theory to algorithms. Cambridge University Press.
- VAPNIK, V. (1995). The nature of statistical learning theory. Springer Science & Business Media.
- WAINWRIGHT, M. J (2019). High-dimensional statistics: a non-asymptotic viewpoint, volume 48. Cambridge University Press.

NOW HIRING PH.D STUDENTS

Topics: Statistical Learning for Sequential and Spatio-Temporal Data

imma-valentina.curato@mathematik.tu-chemnitz.de

Thank you very much
for your attention