



Introduction to Statistical Learning

G. Perrin¹

CEA-EDF-INRIA Summer School | Juin 2021

¹COSYS, Université Gustave Eiffel, 77420 Champs-sur-Marne, France



CEA-EDF-INRIA School

Title : Multi-fidelity, multi-level, model selection/aggregation : how the presence of several versions of a code can improve the prediction of complex phenomena ?

Main objectives :

- How to exploit different versions of a code with a hierarchy of accuracies ?
- How to exploit different competitive versions of a code (no clear hierarchy of precision between the different versions) ?

⇒ Multi-level (ML) and multi-fidelity (MF) for the first question,
⇒ Bayesian model averaging (BMA) and model selection techniques for the second question.



Organization of the week

Day	Title	Speakers
Monday	Introduction	G. Perrin (Université Gustave Eiffel)
Tuesday	Multi-level approaches	F. Nobile (EPFL, Lausanne)
Wednesday	Multi-fidelity approaches	C. Cannamela and B. Kerleguer (CEA)
Thursday	Bayesian Model Averaging	P. Cinnella (Sorbonne Université)
Friday	Model Selection	S. Arlot and M. Gallopin (Université Paris Saclay)

Lecture course in the morning and practical session in the afternoon.



Pedagogical team

Organizing committee :



Teachers :





This course

- 1 What is statistical learning (SL)?
- 2 Formalization of a SL problem
- 3 Generalization and over-fitting
- 4 Model selection
- 5 The particular case of Gaussian regression
- 6 Conclusions



What is statistical learning (SL) ?

Definition (Wikipedia)

- SL theory is a framework for **machine learning** drawing from the fields of **statistics** and **functional analysis**.
- SL theory deals with the problem of finding a **predictive function** based on **data**.

⇒ It is a **hybrid** scientific field (statistics, computer sciences, signal processing, control theory,...) relying on **generalist** techniques (database management, optimization techniques, hardware...) with a **wide range** of applications (computer vision, speech recognition, detection of pathologies, risk analysis...).



Why such an interest in SL ?

- Some resounding successes :
 - 1997 : DeepBlue defeats Kasparov,
 - 2017 : AlphaGO defeats Ke Jie,
 - 2019 : AlphaStar is the new Starcraft II champion.
- **Epistemic** reason : one may not know how to model a complex system, yet have many examples representing a wide variety of situations \Rightarrow "data-driven" vs "model-based".
- **Scientific** reason : learning is an essential faculty of life.
- **Economic** reason : data collection is easier than expertise development.



Learning : a paradigm shift

- The scientific method is historically a **deductive** approach starting from hypotheses \Rightarrow the data **validates** the model.
- Data-driven approaches are **inductive** \Rightarrow the model **is** the output.

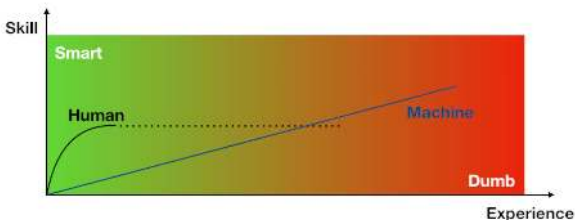
However...

...the efficiency of a data driven model is based on the (strong) assumption that the future will resemble the past, that the set of possibilities is included in the data.

\Rightarrow towards **hybrid models**, combining data and expert models (in particular for the study of physical phenomena / for configurations with limited data).

Remark 1 : intelligence \neq experience

One definition of intelligence is : $\text{intelligence} = \frac{\text{skill}}{\text{experience}}$.



AlphaZero needs 21 Million games of GO for training, but training takes only 24 hours using highly parallelized computers.

To maximize the predictive capabilities of your model (\leftrightarrow its "skill") :

- work on the data collection, from a quantitative and qualitative point of view (to increase its "experience"),
- think carefully about the model and the way it is optimized using the available data (to increase its "intelligence").

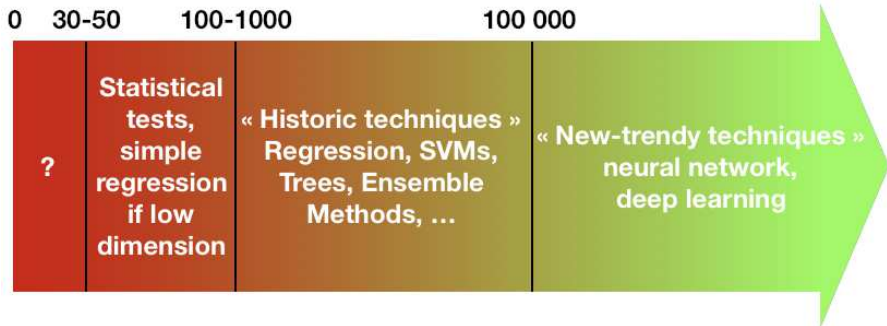
Remark 2 : the GI-GO rule

- Data-driven techniques obey the **Garbage In \Rightarrow Garbage Out** (GI-GO) rule : nonsense input data produces nonsense output.



- Good inference from data requires :
 - having (enough) data of good quality,
 - experts with excellent understanding of this data to guide approaches.

Remark 3 : what is "enough" data ?



There is no universal learning technique : the choice of the method must be adapted to the type of problem considered, as well as to the information available!



Remark 4 : a hypothesis-dependent predictive quality

No free-lunch Theorem (Wolpert, 1996)

For any two machine learning algorithms A and B, the average performance of A and B will be the same across all possible problem instances drawn from a uniform probability distribution.

↔ Every SL model is a simplification of reality, each simplification is based on assumptions (model bias), assumptions fail in certain situations :

- ⇒ no one model works best for all possible situations.
- ⇒ the performance of a SL algorithm on any given problem depends on how well the algorithm's assumptions align with the problem's reality.



Outline of the presentation

- 1 What is statistical learning (SL)?
- 2 Formalization of a SL problem**
- 3 Generalization and over-fitting
- 4 Model selection
- 5 The particular case of Gaussian regression
- 6 Conclusions



Several types of SL problems

Statistical learning falls into several categories, including :

- **supervised** learning : the training data contains the prediction objectives (annotations, labels),
- **unsupervised** learning : the training data is raw,
- **semi-supervised** learning : the training data are partially annotated,
- **transfer** learning : the training data is close to the target problem
- **reinforcement** learning : predictions are derived from a sequence of actions and are characterised by a quality measure ("reward").

In this course, we will focus on **supervised learning**.



Applications of SL for physical models

Objective 1 : accelerate parametric study ("surrogate modeling")

- sensitivity analyses,
- robust conception,
- reliability analyses,...

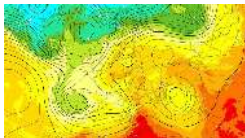




Applications of SL for physical models

Objective 2 : compensate model error ("data-assimilation")

- forecast problems,
- filtering problems,
- smoothing problems,...





Notations

Supervised learning involves learning from a training set of data, noted

$$\mathcal{D}_n = \{(\mathbf{x}_i, y_i), 1 \leq i \leq n\},$$

- \mathbf{x}_i is the **input data** to be interpreted (scalars, vectors, images, texts,...),
 - y_i is the **output data** of interest (value, decision, choice, action, answer, group,...).
- ⇒ The learning problem consists in **inferring the function** f that maps between the input and the output, such that the learned function can be used to **predict** the output y from future input \mathbf{x} :

$$\mathcal{D}_n, \mathbf{x} \rightarrow f(\mathbf{x}; \mathcal{D}_n) \approx y.$$



Formalization of a SL problem

Supervised learning

- Data : $\mathcal{D}_n = \{(\mathbf{x}_i, y_i), 1 \leq i \leq n\}$.
- Goal : learn a function f in a particular **hypothesis space** \mathcal{F} defined on the input space \mathbb{X} (close to) optimal for some **loss function** L :

$$\min_{f \in \mathcal{F}} L(f(\mathbf{x}; \mathcal{D}_n), y) \quad \text{for any new } (\mathbf{x}, y) \in \mathbb{X} \times \mathbb{Y}.$$

- When y is assumed to belong to finite set, we face **Classification**.
- When f depends on parameters to be adjusted, we face **Regression**.

Examples : estimate the price of an apartment, detection of spams...



Definition of a risk function

- To solve the former problem, SL theory takes the perspective that there is some unknown **probability distribution** over the product space $\mathbb{X} \times \mathbb{Y}$, and that the training set is made up of n samples from this probability distribution.
- In this formalism, the inference problem consists in finding the solution of the following optimization problem :

$$\min_{f \in \mathcal{F}} \mathcal{R}(f) := \mathbb{E}[L(f(\mathbf{x}; \mathcal{D}_n), y) \mid \mathcal{D}_n], \quad \mathcal{R} \leftrightarrow \text{"Risk function"}.$$

SL idea

⇒ Learn a function on the Training Data such that the Risk is « small » **on average** or **with high probability**.



Best solution

Let us rewrite the risk as Euclidean projection :

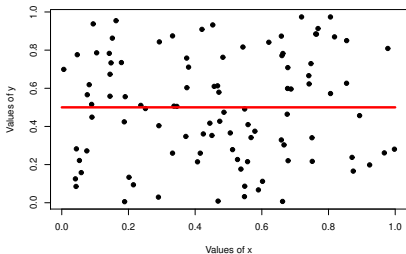
$$\begin{aligned}\mathcal{R}(f) &= \mathbb{E}[(f(\mathbf{x}; \mathcal{D}_n) - y)^2 \mid \mathcal{D}_n] \\ &= \mathbb{E}[(f(\mathbf{x}) - \mathbb{E}[y|\mathbf{x}, \mathcal{D}_n])^2 \mid \mathcal{D}_n] + \mathbb{E}[(\mathbb{E}[y|\mathbf{x}, \mathcal{D}_n] - y)^2 \mid \mathcal{D}_n].\end{aligned}$$

⇒ For Euclidean-related risks, optimal solution is related to **conditional expectation**.

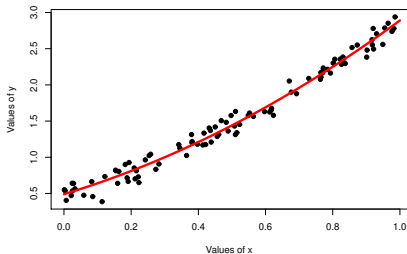
Issue : We do not have access to the law of (\mathbf{x}, y) neither to $\mathbb{E}[y|\mathbf{x}, \mathcal{D}_n]$.

Strategy : build upon \mathcal{D}_n to approximate it.

Conditional expectation in graphs



(a) $x \sim U(0, 1) \perp y \sim U(0, 1)$



(b) $y = (x + 0.7)^2 + \xi$, $x \sim U(0, 1) \perp \xi \sim \mathcal{N}(0, 0.1^2)$

FIGURE: Black dots $\leftrightarrow (x_i, y_i)$ / Red line $\leftrightarrow \mathbb{E}[y|x]$

In 1D, the conditional average can be seen as a moving average...



Minimization of the empirical risk

- As y is unknown for a non-observed \mathbf{x} , $L(f(\mathbf{x}; \mathcal{D}_n), y)$ can not be computed, and the former problem is ill-defined.
- Assuming that the $(\mathbf{x}_i, y_i)_i$ are independent and identically distributed (iid) (or exchangeable), the former problem is generally replaced by :

$$\min_{f \in \mathcal{F}} \widehat{\mathcal{R}}_n(f) := \frac{1}{n} \sum_{i=1}^n [L(f(\mathbf{x}_i; \mathcal{D}_n), y_i)], \quad \widehat{\mathcal{R}}_n \leftrightarrow \text{"Empirical Risk function"}.$$

- Depending on the choice for \mathcal{F} , explicit solutions can be derived.
- In the general case, we will try as much as possible to propose loss functions which are **derivable** and **convex** in order to more easily solve this problem.



Remarks on the minimization problem

$$\min_{f \in \mathcal{F}} \widehat{\mathcal{R}}_n(f) := \frac{1}{n} \sum_{i=1}^n [L(f(\mathbf{x}_i; \mathcal{D}_n), y_i)], \quad \widehat{\mathcal{R}}_n \leftrightarrow \text{"Empirical Risk function"}.$$

- Minimizing $\widehat{\mathcal{R}}_n$ is generally an **ill-posed problem**, in the sense that it does not admit a unique solution that depends continuously on the initial conditions \rightarrow there may exist an infinite number of solutions making \mathcal{R} be equal to 0.
- The fact that $\widehat{\mathcal{R}}_n$ tends to \mathcal{R} a.s. when n tends to infinity is **not sufficient** to make $\min_{f \in \mathcal{F}} \widehat{\mathcal{R}}_n(f)$ tend to $\min_{f \in \mathcal{F}} \mathcal{R}(f) \rightarrow$ additional conditions on \mathcal{F} are required.



Summary of the SL process

1. Choose $\mathcal{F}, L, \mathcal{R}$ (**Problem definition**)
2. Observe $\mathcal{D}_n = \{(\mathbf{x}_i, y_i), 1 \leq i \leq n\}$ (**Data**)
3. Build $\mathcal{A} : \mathcal{D}_n \mapsto$ measurable function (**Learning strategy**)
4. Consider $\hat{f} := \mathcal{A}(\mathcal{D}_n)$ (**Predictor learnt**)
5. $\mathcal{R}(\hat{f})$ is the **Score**, the smaller the better.



Outline of the presentation

- 1 What is statistical learning (SL)?
- 2 Formalization of a SL problem
- 3 Generalization and over-fitting**
- 4 Model selection
- 5 The particular case of Gaussian regression
- 6 Conclusions



Generalization

Definition


Generalization is the ability of a model to make correct predictions on new data, which have not been used to build it.

⇒ **Generalization ≠ memorization !**

For instance,

$$\widehat{f}(\mathbf{x}; \mathcal{D}_n) := \begin{cases} y_i & \text{if } \mathbf{x} = \mathbf{x}_i, \\ y_J & \text{with } J \text{ picked at random in } \{1, \dots, n\} \text{ otherwise,} \end{cases}$$

allows to cancel the empirical risk, $\widehat{\mathcal{R}}(\widehat{f}) = 0$, but it is clearly an example of a model with weak generalization capacity ($\mathcal{R}(\widehat{f})$ has no reason to be small).



Over-fitting

- As we have seen, it is generally easy to make $\widehat{\mathcal{R}}(\widehat{f})$ be equal to 0, and that small values of $\widehat{\mathcal{R}}(\widehat{f})$ do not necessarily imply low values of $\mathcal{R}(\widehat{f})$.
 - In (most of) SL problems, the data is **noisy** (experimental errors, labelling problems, parameters not taken into account...).
- ⇒ We say that a model is **over-fitting** when it also learns the noise of the data, and is not able to generalize. An over-fitting model is generally too "complex", which allows it to stick too much to the data.
- ⇒ On the contrary, we say that a model is **under-fitting** when it is too simple to have good performances even on training data.

Take away message

$\widehat{\mathcal{R}}(\widehat{f}) \approx 0$ is a **necessary** but **not sufficient** condition for \widehat{f} to be a predictive model.



Bias-variance trade-off

If $\mathcal{R}^* := \min_{f \in \{\text{measurable functions}\}} \mathcal{R}(f)$,

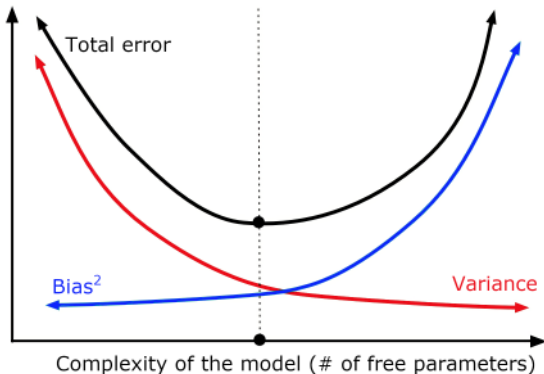
$$\mathcal{R}(\hat{f}) - \mathcal{R}^* = \underbrace{\left[\mathcal{R}(\hat{f}) - \min_{f \in \mathcal{F}} \mathcal{R}(f) \right]}_{\text{variance}} + \underbrace{\left[\min_{f \in \mathcal{F}} \mathcal{R}(f) - \mathcal{R}^* \right]}_{\text{bias}}.$$

- $e_v := \mathcal{R}(\hat{f}) - \min_{f \in \mathcal{F}} \mathcal{R}(f)$ quantifies the distance between \hat{f} and the best model in $\mathcal{F} \leftrightarrow$ **estimation error** (noise of \mathcal{D}_n , optim. issues...).
- $e_b := \min_{f \in \mathcal{F}} \mathcal{R}(f) - \mathcal{R}^*$ quantifies the quality of the optimal model in \mathcal{F} , and therefore the relevance of $\mathcal{F} \leftrightarrow$ **approximation error**.

"The price to pay for achieving low bias is high variance", Geman, 1992.

→ by increasing the complexity of \mathcal{F} , we reduce e_b , but e_v is likely to increase as the search in that space may be more difficult.

Bias-variance trade-off



⇒ a **trade-off** between bias and variance is needed to adjust the complexity of \mathcal{F} to the available data in \mathcal{D}_n .



The need for regularization

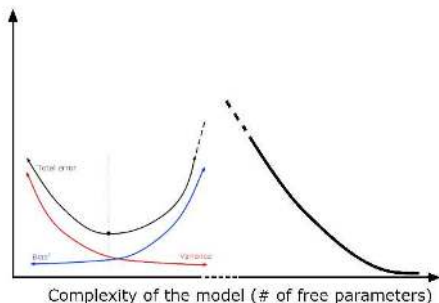
To limit the effect of overfitting, we need to control the complexity of the hypothesis space $\mathcal{F} \Rightarrow$ this is the objective of **regularization** :

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \widehat{R}(f) + \underbrace{\lambda \Omega(f)}_{\text{Regularization}} ,$$

- $\Omega(f)$ is a constraint on possible solution (it can be equal to the number of non-zero parameters on which the function f depends),
- $\lambda \geq 0$ is a (hyper-)parameter introduced to balance the importance of each term :
 - when $\lambda \rightarrow 0$, there is no regularization and the variance is likely to dominate the bias,
 - when $\lambda \rightarrow +\infty$, the regularization is dominating, and there is no more learning, the variance tends to 0 and the bias is likely to be high.



The surprising case of neural networks



- By strongly increasing the complexity of the model, the need for the bias-variance trade-off seems to disappear in some situations.
- A conjecture to meditate on : "the more we increase the number of model parameters, the more we reduce the number of local minima, and the more likely we are to converge towards the global minimum".



Outline of the presentation

- 1 What is statistical learning (SL)?
- 2 Formalization of a SL problem
- 3 Generalization and over-fitting
- 4 Model selection**
- 5 The particular case of Gaussian regression
- 6 Conclusions



Test sets to validate the model

- As no method is a priori better than another in the general case (no free-lunch Theorem), it is generally interesting to consider several models, and select the most appropriate one for the considered application.
- In this presentation, a model f is better than a model h if $\mathcal{R}(f) < \mathcal{R}(h)$, but other criteria could be considered (computational resources required, for example).
- To estimate $\mathcal{R}(f)$, it is essential to have data that **has not been used** to build model f .
- The easiest way to do this is to split the available data into **training** and **test** sets.
- The empirical risk calculated with the test set is usually a good estimator of the loss function.



Validation set

- In the case where we want to choose between M models f_1, \dots, f_M , a natural approach is to consider :

$$\hat{f} = \arg \min_{1 \leq m \leq M} \sum_{(x,y) \in \mathcal{D}_{\text{test}}} L(f_m(\mathbf{x}; \mathcal{D}_n), y).$$

- Nevertheless, the test set is now used to select the final model, and does not represent an independent set of point composed of new data.
- ⇒ To correctly estimate the generalization error of \hat{f} , a possible solution is to divide the available data in three sets :
- the **training** set \mathcal{D}_n allows the training of each model,
 - the **validation** set $\mathcal{D}_{\text{valid}}$ allows comparing the different models,
 - the **test** set $\mathcal{D}_{\text{test}}$ is used to estimate the generalization error.

Once a model selected, it can be trained again based on \mathcal{D}_n and $\mathcal{D}_{\text{valid}}$.



Cross-validation

The separation of training and test data is necessarily arbitrary and can create unrepresentative data sets.

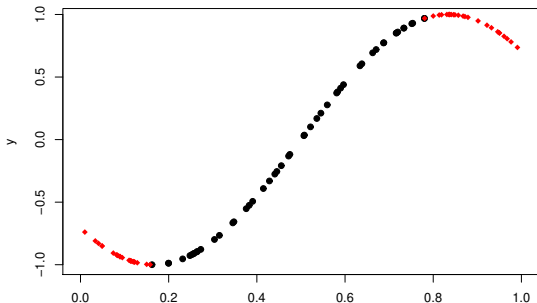


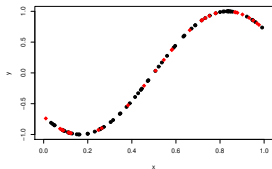
FIGURE: $\mathcal{D}_n = \{\text{black dots}\}^x$, $\mathcal{D}_{\text{test}} = \{\text{red squares}\}$



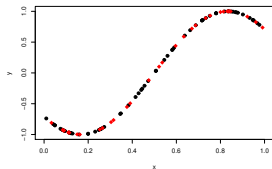
Cross-validation

To avoid pathological configurations, **cross-validation** relies on :

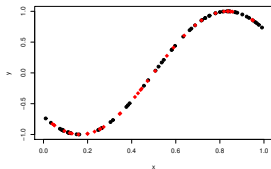
1. the partition of the data in K subsets ("folds") of similar sizes, $\mathcal{D}^{(1)}, \dots, \mathcal{D}^{(K)}$,
2. for each $1 \leq k \leq K$:
 - train a model using $\bigcup_{l \neq k} \mathcal{D}^{(l)}$,
 - evaluate the model using $\mathcal{D}^{(k)}$.



(a) $k = 1$



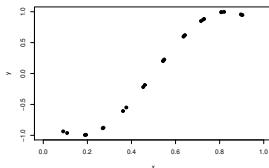
(b) $k = 2$



(c) $k = 3$

A limit case : the leave-one-out cross-validation

- The more available data, the more trained the model.
 - To get as close as possible to the case where the model is trained by n points, one is then encouraged to make K tend towards n . This is known as **Leave-one-out** (LOO) validation.
- + particularly attractive for small-dimensional datasets,
- may involve large computation times,
 - may strongly underestimate the generalization errors when the data shows clusters.





An empirical approach : the bootstrap method

- Another approach to estimating the generalization error is the **bootstrap** method.
- It consists in building $K \gg 1$ training sets $\mathcal{D}_1, \dots, \mathcal{D}_K$ gathering n elements of \mathcal{D}_n chosen at random **with replacement**.
- The complementary sets $\mathcal{D}_n \setminus \mathcal{D}_k$ then define n test sets that can be used to evaluate the generalization error.

Remark. The probability for a couple (x_i, y_i) to be in a set \mathcal{D}_k is equal to $1 - (1 - 1/n)^n$, which tends to $1 - e^{-1} \approx 0.63$ when n increases, such that each set \mathcal{D}_k gathers in average 2/3 of the elements of \mathcal{D}_n .



Comparison to naive techniques

From a slightly different perspective, the performance of an SL approach can (should) also be measured through comparison with very simple or even naive approach (which require little or no training), such as, for instance :

- **mean predictor** : return the mean value of y_i ,
- **random predictor** : return a value y_i chosen at random in \mathcal{D}_n ,
- **nearest neighbor predictor** : return y_i such that $i = \arg \min_{1 \leq j \leq n} \|\mathbf{x} - \mathbf{x}_j\|$ for a particular norm $\|\cdot\|$.

⇒ we can then measure the **learning capacity** of the model.



Choice of a model

Finally, it is important to keep in mind that other criteria than the generalization error can be considered to evaluate the performance of a SL method, such as :

- learning costs and time,
- prediction costs and time (offline-online distinction),
- the number of hyperparameters,
- the "effort" of familiarising oneself with an algorithm.



Outline of the presentation

- 1 What is statistical learning (SL)?
- 2 Formalization of a SL problem
- 3 Generalization and over-fitting
- 4 Model selection
- 5 The particular case of Gaussian regression**
- 6 Conclusions



The Gaussian case (with known mean and covariance)

- Let us assume that $y|\mathbf{x} \sim \text{PG}(\mu(\mathbf{x}), C(\mathbf{x}, \mathbf{x}))$. It comes :

$$\begin{pmatrix} y \\ \mathbf{y} \end{pmatrix} | \mathbf{x}, \mathbf{x}_1, \dots, \mathbf{x}_n \sim \mathcal{N} \left(\begin{pmatrix} \mu(\mathbf{x}) \\ \boldsymbol{\mu} \end{pmatrix}, \begin{bmatrix} C(\mathbf{x}, \mathbf{x}) & C(\mathbf{x}, \mathcal{X}_n) \\ C(\mathbf{x}, \mathcal{X}_n)^T & C(\mathcal{X}_n, \mathcal{X}_n) \end{bmatrix} \right),$$

$$\mu_i := \mu(\mathbf{x}_i), \quad (C(\mathbf{x}, \mathcal{X}_n))_i := C(\mathbf{x}, \mathbf{x}_i), \quad (C(\mathcal{X}_n, \mathcal{X}_n))_{ij} = C(\mathbf{x}_i, \mathbf{x}_j).$$

- By **Gaussian conditioning**, we deduce :

$$y | \mathbf{x}, \mathcal{D}_n \sim \mathcal{N}(\mu_c(\mathbf{x}), C_c(\mathbf{x}, \mathbf{x})),$$

$$\mathbb{E}[y | \mathbf{x}, \mathcal{D}_n] = \mu_c(\mathbf{x}) := \mu(\mathbf{x}) + C(\mathbf{x}, \mathcal{X}_n)C(\mathcal{X}_n, \mathcal{X}_n)^{-1}(\mathbf{y} - \boldsymbol{\mu}),$$

$$C_c(\mathbf{x}, \mathbf{x}) := C(\mathbf{x}, \mathbf{x}) - C(\mathbf{x}, \mathcal{X}_n)C(\mathcal{X}_n, \mathcal{X}_n)^{-1}C(\mathbf{x}, \mathcal{X}_n)^T.$$



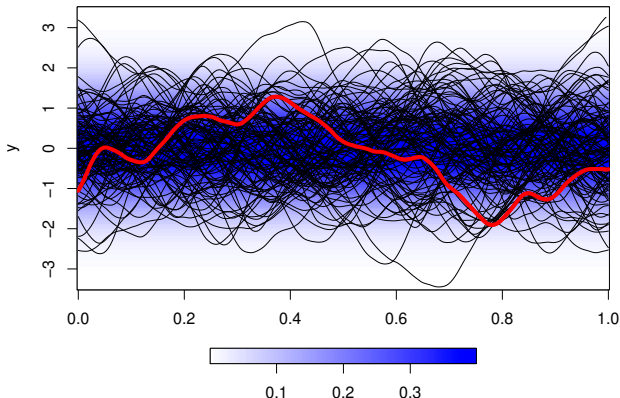
Remarks on the Gaussian case

- Assuming that $y|x$ is Gaussian leads to an **affine expression** for $\mathbb{E}[y | \mathbf{x}, \mathcal{D}_n]$ with respect to \mathbf{y} .
 - the prediction is a **weighted sum** of the observations.
- Reciprocally, if we assume that the conditional expectation of y given \mathbf{x} and \mathcal{D}_n is a simple linear function of \mathbf{y} , $\mathbb{E}[y | \mathbf{x}, \mathcal{D}_n] = \mathbf{w}(\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{y}$, we can show that the expression for optimal (from the risk function minimization point of view) $\mathbf{w}(\mathbf{x})$ and $\mathbf{A}(\mathbf{x})$ is given by :

$$\mathbf{A}(\mathbf{x}) = C(\mathbf{x}, \mathcal{X}_n)C(\mathcal{X}_n, \mathcal{X}_n)^{-1}, \quad \mathbf{w}(\mathbf{x}) = \mu(\mathbf{x}) - \mathbf{A}(\mathbf{x})\boldsymbol{\mu}.$$

- although it may be convenient to assume that $y|x$ is Gaussian, it is **not necessary** to make this assumption, so long as the assumed distribution has well defined first and second moments.
- Gaussian regression **interpolates** the data → the empirical risk is always zero.

Graphical illustration ($\mu = 0$, $C \leftrightarrow \text{Matern52}(\theta = 0.1)$)



The Gaussian prior consists in assuming that the function to predict (here in red thick solid line) is one particular realization of the considered Gaussian process (among an infinite number of possible realizations).

Graphical illustration ($\mu = 0$, $C \leftrightarrow \text{Matern52}(\theta = 0.1)$)

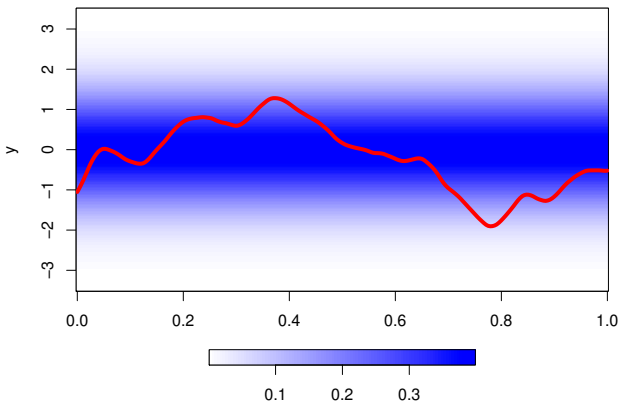


FIGURE: Red continuous line \leftrightarrow function to predict / contour plots \leftrightarrow PDF of $y|x, \mathcal{D}_n$ / Vertical lines \leftrightarrow positions of x_i / Black line $\leftrightarrow \mathbb{E}[y|x, \mathcal{D}_n]$

Graphical illustration ($\mu = 0, C \leftrightarrow \text{Matern52}(\theta = 0.1)$)

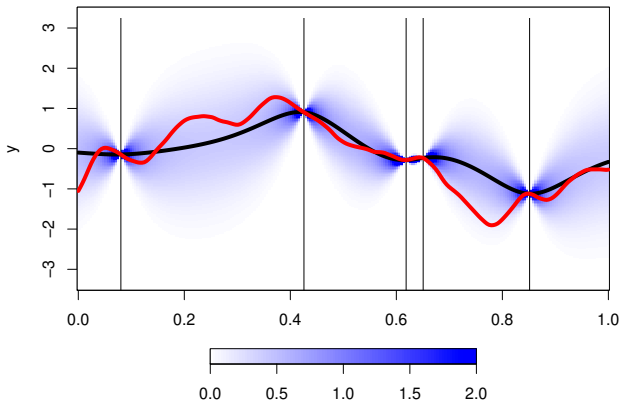


FIGURE: Red continuous line \leftrightarrow function to predict / contour plots \leftrightarrow PDF of $y|x, \mathcal{D}_n$ / Vertical lines \leftrightarrow positions of x_i / Black line $\leftrightarrow \mathbb{E}[y|x, \mathcal{D}_n]$

Graphical illustration ($\mu = 0$, $C \leftrightarrow \text{Matern52}(\theta = 0.1)$)

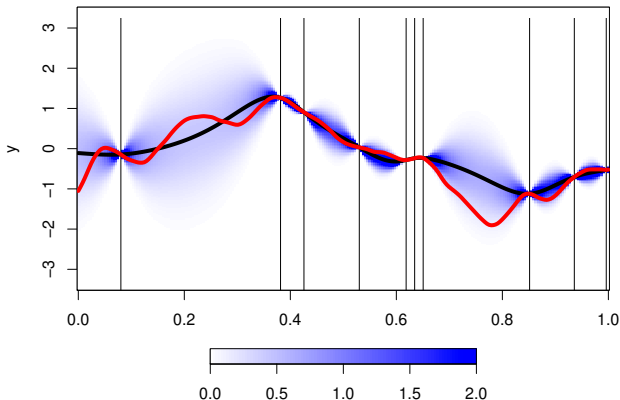


FIGURE: Red continuous line \leftrightarrow function to predict / contour plots \leftrightarrow PDF of $y|x, \mathcal{D}_n$ / Vertical lines \leftrightarrow positions of x_i / Black line $\leftrightarrow \mathbb{E}[y|x, \mathcal{D}_n]$

Graphical illustration ($\mu = 0$, $C \leftrightarrow \text{Matern52}(\theta = 0.1)$)

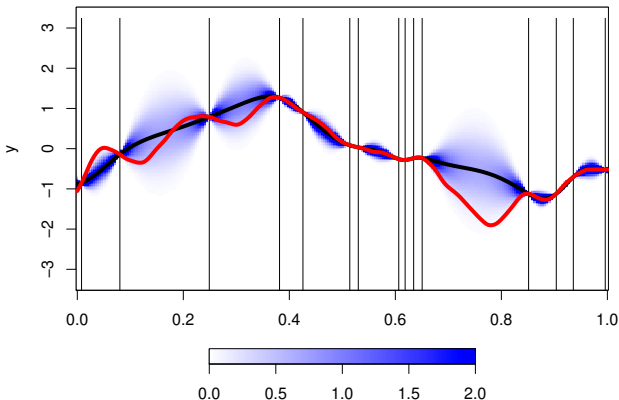


FIGURE: Red continuous line \leftrightarrow function to predict / contour plots \leftrightarrow PDF of $y|x, \mathcal{D}_n$ / Vertical lines \leftrightarrow positions of x_i / Black line $\leftrightarrow \mathbb{E}[y|x, \mathcal{D}_n]$

Graphical illustration ($\mu = 0$, $C \leftrightarrow \text{Matern52}(\theta = 0.1)$)

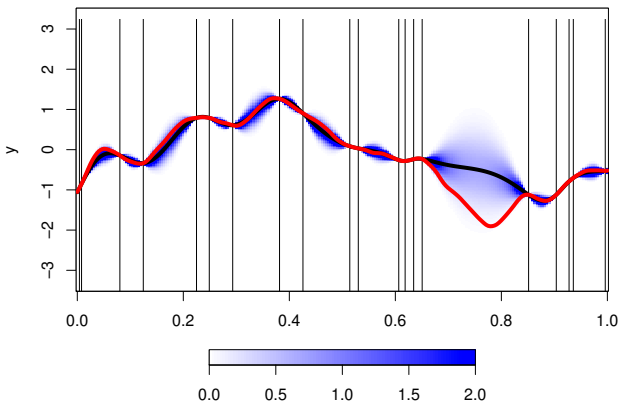


FIGURE: Red continuous line \leftrightarrow function to predict / contour plots \leftrightarrow PDF of $y|x, \mathcal{D}_n$ / Vertical lines \leftrightarrow positions of x_i / Black line $\leftrightarrow \mathbb{E}[y|x, \mathcal{D}_n]$

Graphical illustration ($\mu = 0$, $C \leftrightarrow \text{Matern52}(\theta = 0.1)$)

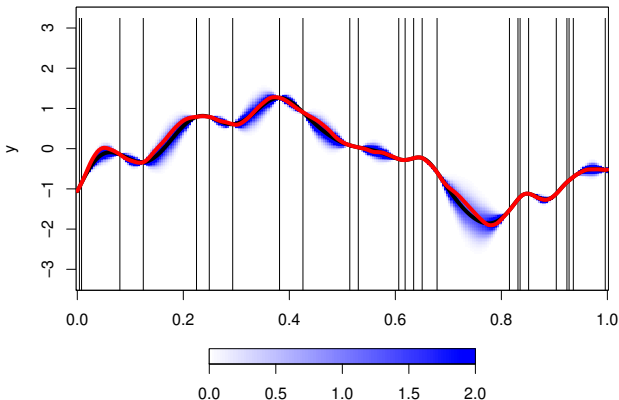


FIGURE: Red continuous line \leftrightarrow function to predict / contour plots \leftrightarrow PDF of $y|x, \mathcal{D}_n$ / Vertical lines \leftrightarrow positions of x_i / Black line $\leftrightarrow \mathbb{E}[y|x, \mathcal{D}_n]$



The Gaussian case with unknown statistical moments

Problem : the mean and covariance functions of $y|\mathbf{x}$ are generally **unknown** but need to be estimated from the data.

→ a very popular choice for μ and C is :

$$\mu(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T \boldsymbol{\beta}, \quad C(\mathbf{x}, \mathbf{x}') = \sigma^2 R(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}),$$

with \mathbf{f} a **chosen** vector-valued function, $R(\cdot, \cdot; \boldsymbol{\theta})$ a parametric covariance function, and $\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}$ **unknown** parameters modeled by random quantities.

For instance :

$$\mathbf{f}(\mathbf{x}) = (1, x_1, \dots, x_d), \quad R(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}) = \exp\left(-\sum_{k=1}^d \frac{(x_k - x'_k)^2}{\theta_k^2}\right).$$



The Gaussian case with unknown statistical moments

$$y|\mathbf{x}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta} \sim \text{PG}(\mathbf{f}(\mathbf{x})^T \boldsymbol{\beta}, \sigma^2 R(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})).$$

- If parametric expressions are proposed for μ and C , the conditional expectation can be rewritten :

$$\begin{aligned}\mathbb{E}[y | \mathbf{x}, \mathcal{D}_n] &= \mathbb{E}[\mathbb{E}[y | \mathbf{x}, \mathcal{D}_n, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}] | \mathbf{x}, \mathcal{D}_n] \\ &= \mathbb{E}_{\boldsymbol{\beta}, \boldsymbol{\theta}}[\mathbf{r}(\mathbf{x}; \boldsymbol{\theta})^T \mathbf{y} + (\mathbf{f}(\mathbf{x}) - \mathbf{F}\mathbf{r}(\mathbf{x}; \boldsymbol{\theta}))^T \boldsymbol{\beta} | \mathbf{x}, \mathcal{D}_n],\end{aligned}$$

$$\mathbf{r}(\mathbf{x}; \boldsymbol{\theta}) := R(\mathcal{X}_n, \mathcal{X}_n; \boldsymbol{\theta})^{-1} R(\mathcal{X}_n, \mathbf{x}; \boldsymbol{\theta}), \quad (\mathbf{F})_{ji} = f_j(\mathbf{x}_i).$$

- Taking into account the unknown character of the parameters results in an **average** of the former expression of the posterior mean with respect to the conditional distribution of $(\boldsymbol{\beta}, \boldsymbol{\theta})$.



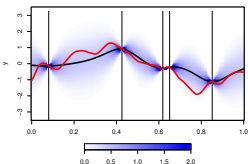
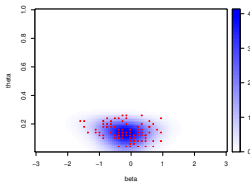
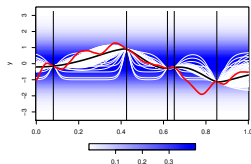
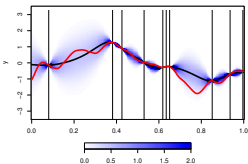
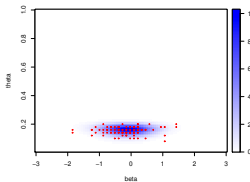
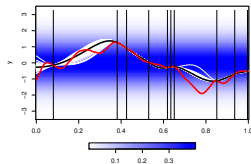
The Gaussian case with unknown statistical moments

If $f_{\beta, \sigma^2, \theta}$ is the prior distribution of $(\beta, \sigma^2, \theta)$, the distribution of $(\beta, \sigma^2, \theta) \mid \mathbf{x}, \mathcal{D}_n$ is proportional to :

$$\frac{f_{\beta, \sigma^2, \theta}(\beta, \sigma^2, \theta)}{(\det(R(\mathcal{X}_n, \mathcal{X}_n; \theta)))^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{F}^T \beta - \mathbf{y})^T R(\mathcal{X}_n, \mathcal{X}_n; \theta)^{-1}(\mathbf{F}^T \beta - \mathbf{y})\right).$$

- Sampling techniques (such as Markov Chain Monte Carlo) can be used to sample iid realizations of $(\beta, \sigma^2, \theta) \mid \mathbf{x}, \mathcal{D}_n$ to empirically estimate the conditional expectation.
- As the former expression does not depend on \mathbf{x} , the **same** realizations can be used for the prediction of the output in any \mathbf{x} .
- Simplified expressions can be obtained when considering additional assumptions on the prior distribution (ex : β Gaussian).

Graphical illustration ($\mu = 0$, $C \leftrightarrow \text{Matern52}(\theta = 0.1)$)

(a) With known μ, C (b) $f_{\beta, \theta | \mathbf{y}}$ (c) With parametric μ, C (d) With known μ, C (e) $f_{\beta, \theta | \mathbf{y}}$ (f) With parametric μ, C



Outline of the presentation

- 1 What is statistical learning (SL)?
- 2 Formalization of a SL problem
- 3 Generalization and over-fitting
- 4 Model selection
- 5 The particular case of Gaussian regression
- 6 Conclusions**



Conclusions

- Statistical learning is associated with an increasing number of scientific applications.
- In this presentation, we introduced the standard formalism of statistical learning (loss and risk functions, empirical risk, conditional expectation...)
- The links between generalization error and predictive capacity of models were highlighted (bias-variance trade-off, over-fitting...).
- When expert models exist, the role of SL is to enrich the information that these models provide rather than to replace them ("data-driven physical models" ...).
- There is no optimal SL model. Choosing a SL method requires to know the data and to know the problem !



Thank you for your attention.