Kernel Methods for Statistical Learning

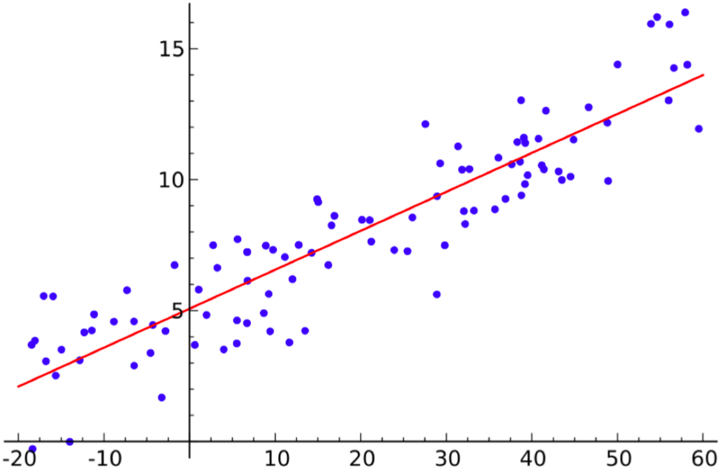
Jakob Verbeek
jakob.verbeek@inria.fr

October 7, 2014

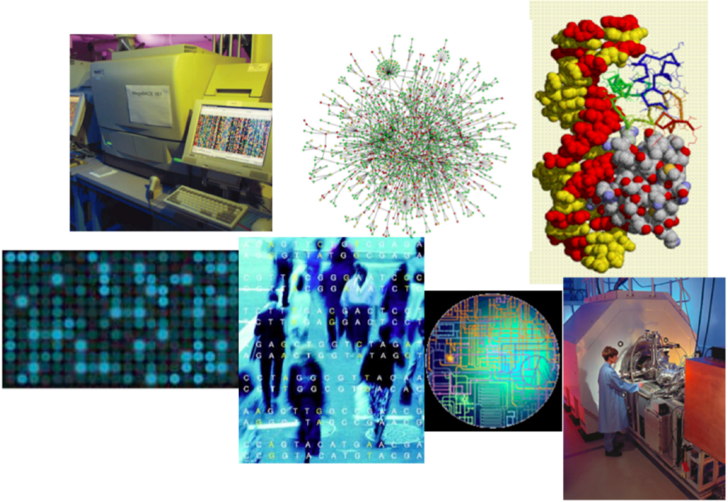
Practical aspects

- Six classes of three hours each.
- Assessment: 1/2 project, 1/2 homeworks.
- Projects: study article, either methods (implementation), or theoretical. You are free to suggest articles, or pick one from the website (more papers coming).
- End of November: preliminary report (25% of the grade). January: final (short) report.
- Three homeworks (after lectures 2, 4, and 6), due within three weeks by email.
- Website: http://lear.inrialpes.fr/people/mairal/teaching/2014-2015/MSIAM

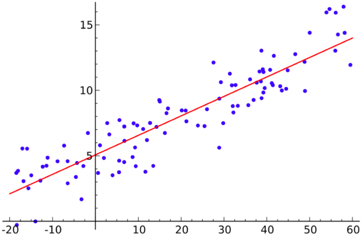
From classic linear learning problems ...

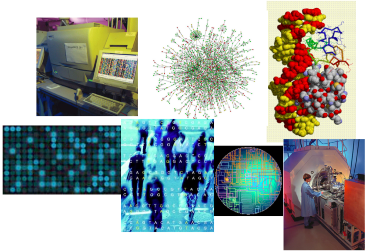


... to current practical learning problems



Main goals of this course





Extend well understood linear statistical learning techniques to real-world complicated, structured and high-dimensional data (images, text, time series, graphs, distributions, permutations, ...)

- Present basic theory of kernels and statistical learning
- Develop working knowledge for practical kernel design

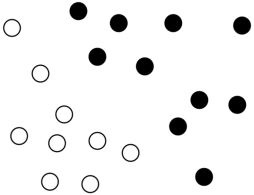
Outline of this class

1. A few examples.
2. Bias/variance trade-off and how to deal with it.
3. Statistical learning theory.

* thanks to Laurent Jacob for his slides

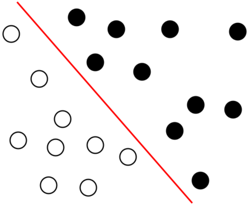
Statistical learning and applications

- This class is concerned with learning from data. Essentially:



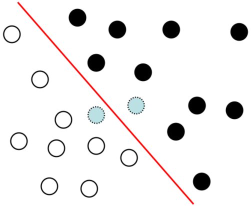
Statistical learning and applications

- This class is concerned with learning from data. Essentially:



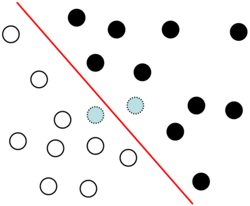
Statistical learning and applications

- This class is concerned with learning from data. Essentially:



Statistical learning and applications

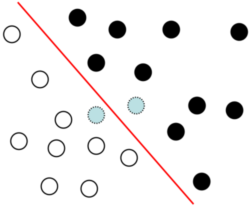
- This class is concerned with learning from data. Essentially:



- Also: multi-class, regression, unsupervised...

Statistical learning and applications

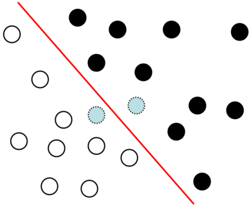
- This class is concerned with learning from data. Essentially:



- Also: multi-class, regression, unsupervised...
- We start with a few examples to make things concrete.

Statistical learning and applications

- This class is concerned with learning from data. Essentially:

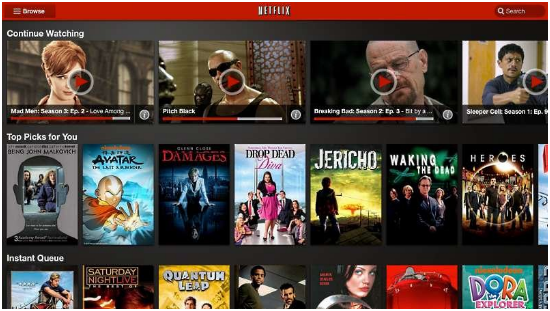


- Also: multi-class, regression, unsupervised...
- We start with a few examples to make things concrete.
- These examples highlight a general problem which we will discuss right after.

Part I

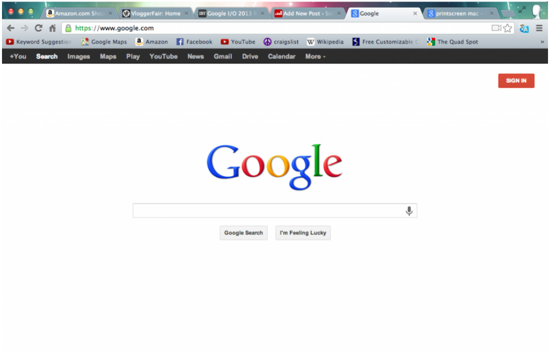
A few examples

Recommender systems



Given a user and the movies he liked, what should he watch next?

Web



Given a query what are the most relevant webpages?

Natural language processing

- Given a text, predict its topic.
- Given an email, predict whether it is a spam.
- Given a text, predict its translation in another language.

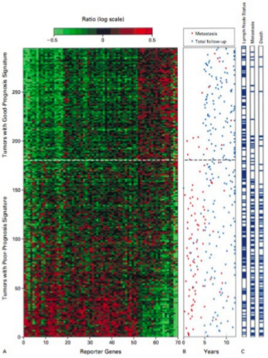
Biological data in high dimension

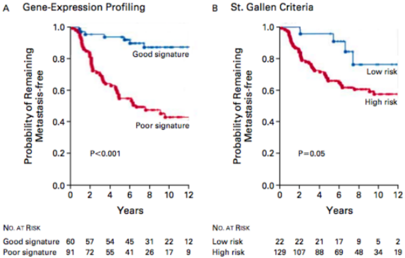
Modern technologies in molecular biology provide descriptions of individuals through thousands/millions of descriptors:

- Gene expression (arrays, sequencing),
- SNPs,
- Methylations,
- ...

Potential to allow better understanding/prediction of complex phenomena.

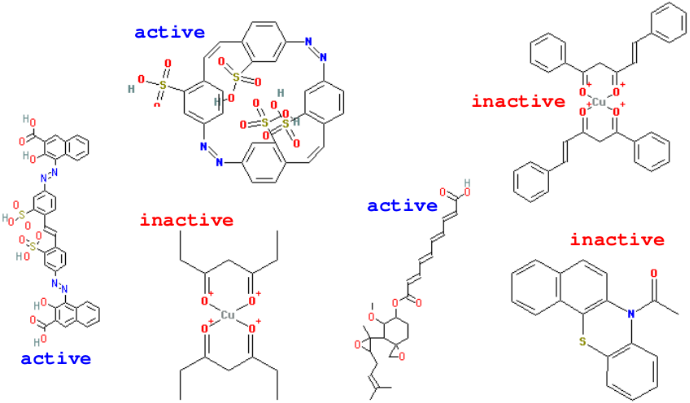
Tumor classification for prognosis





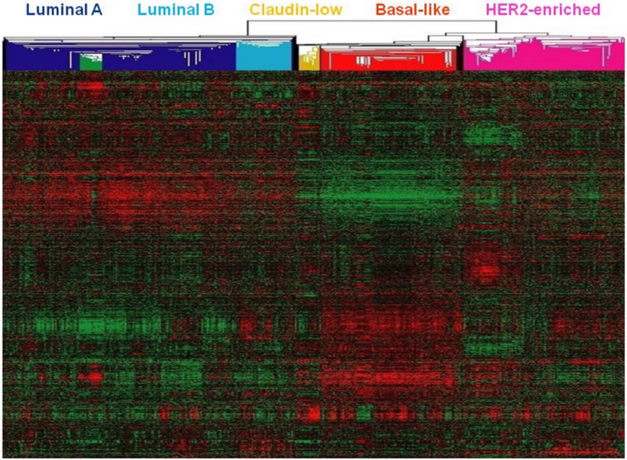
- Given the expression of the genes in a new tumor, predict the occurrence of a metastasis in the next 5 years.
- Similarly: diagnosis.

Molecule classification for drug design



Given a candidate molecule, is it active against a therapeutical target?

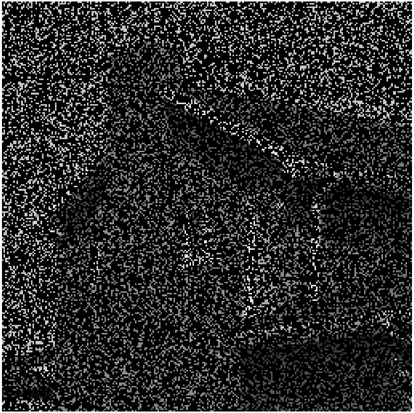
Gene expression clustering

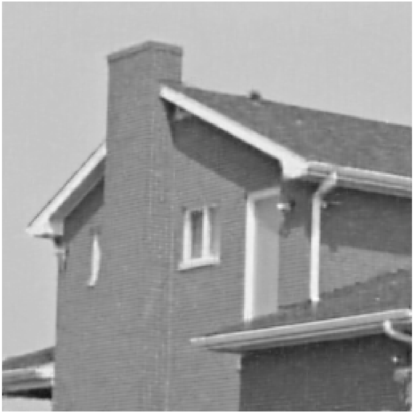


(from C. Perou's website)

Are there groups of breast tumors with similar gene expression profile?

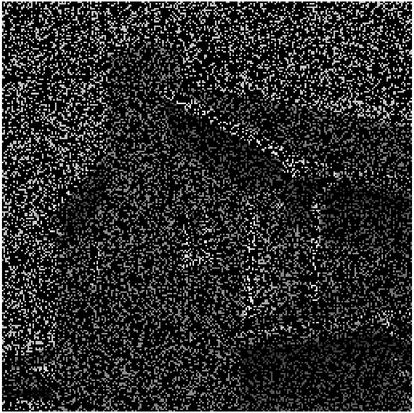
Image inpainting

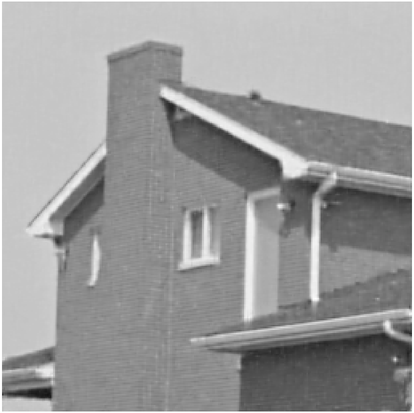




Complete an image with missing parts.

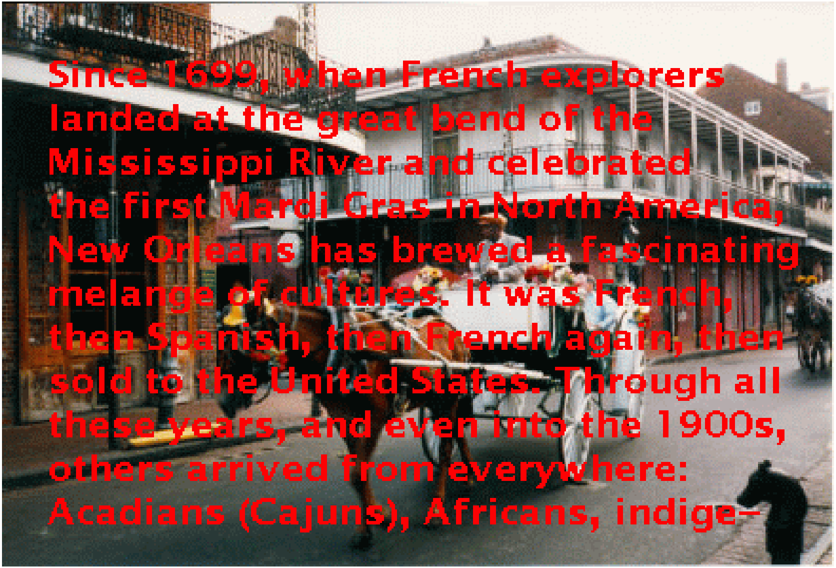
Image inpainting





Estimation problem: predict each image patch, as a linear combination of dictionary elements.

Image inpainting



Since 1699, when French explorers landed at the great bend of the Mississippi River and celebrated the first Mardi Gras in North America, New Orleans has brewed a fascinating melange of cultures. It was French, then Spanish, then French again, then sold to the United States. Through all these years, and even into the 1900s, others arrived from everywhere: Acadians (Cajuns), Africans, indige–

Image inpainting

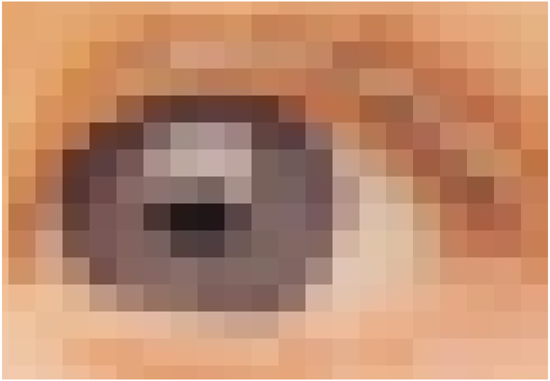
Image up-scaling

Improve the quality of an image.

Image up-scaling

Improve the quality of an image.

Image up-scaling



Improve the quality of an image.

Image up-scaling

Improve the quality of an image.

Image understanding

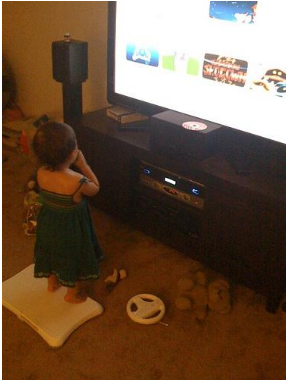
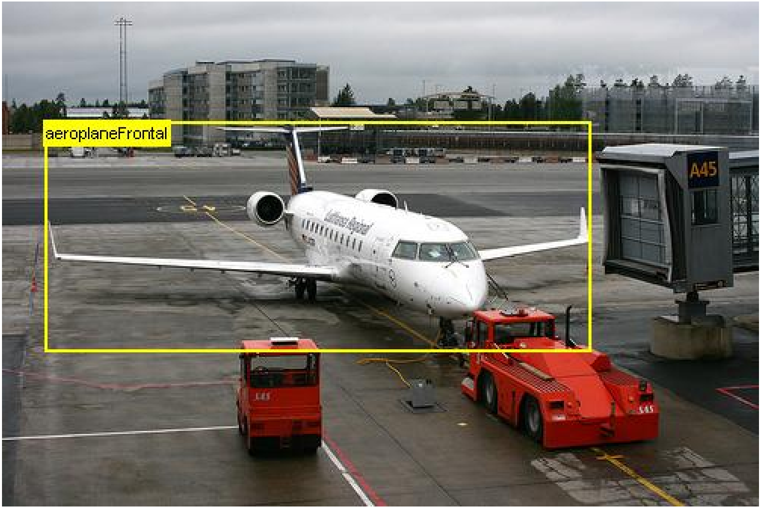


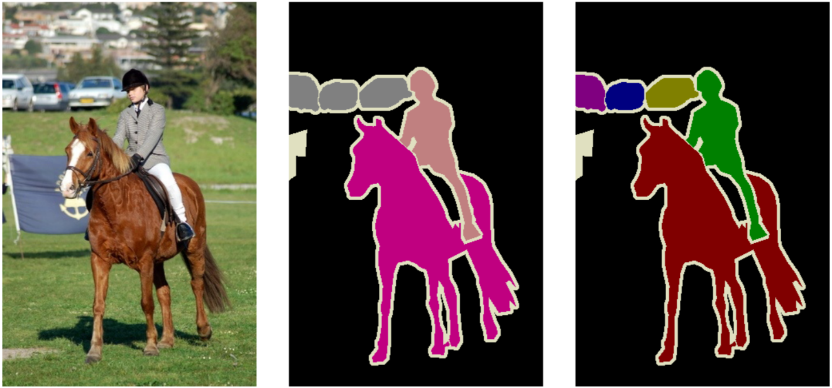
Image classification: Person=yes, TV=yes, car=no, ...

Image understanding



Object category localization: bounding box prediction.

Image understanding

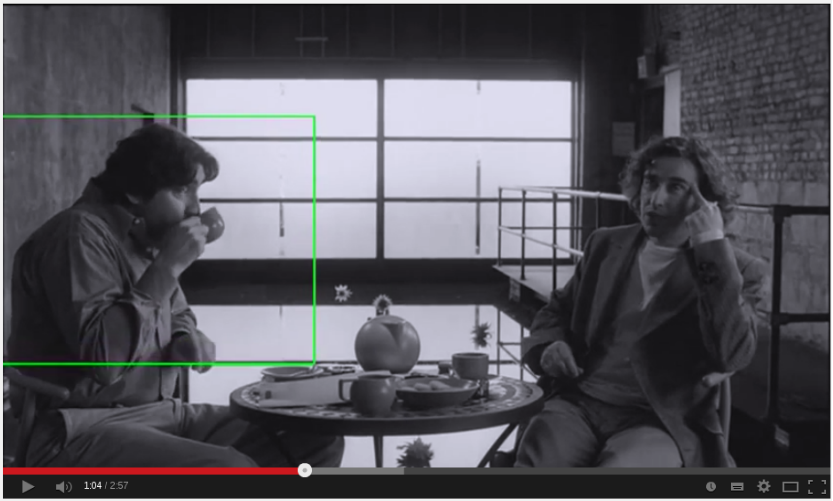


Semantic image segmentation: label pixels with object classes.

Video understanding

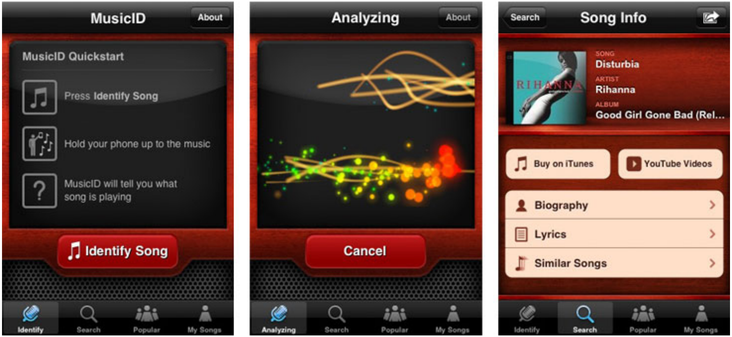
Event recognition: classify video as being, e.g., a birthday party video.

Video understanding



Action recognition: locate actions of interest in video.

Music recognition



MusicID Quickstart

Press Identify Song

Hold your phone up to the music

MusicID will tell you what song is playing

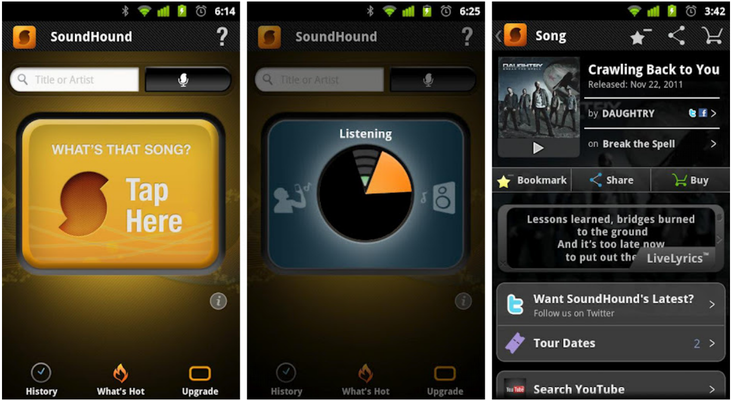
SONG Disturbia

ARTIST Rihanna

ALBUM Good Girl Gone Bad (Rel...

Guess which tune is being played.

Music recognition



Lessons learned, bridges burned
to the ground
And it's too late now
to put out the

Guess which tune is being tapped/hummed.

Learning with high dimensional data

- Each of these examples involves **complex objects/large numbers of features** for a **restricted number of samples**.
- Intuitively, observing all these characteristics should allow us to predict or understand complex mechanisms.
- We now discuss why this wealth of features can cause trouble in statistical learning.
- Understanding this problem should give more perspective to the tools we will present later.
- **Disclaimer:** no kernels today, they come later once we have established the setting. Intuitively: similarity functions to compare objects that do not live in vector spaces.

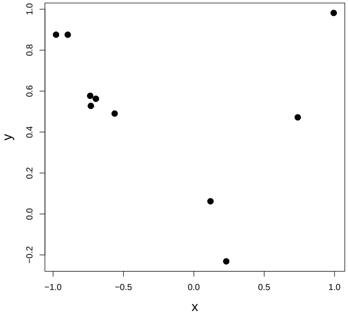
Part II

Overfitting, bias-variance tradeoff: what is the problem?

Short term

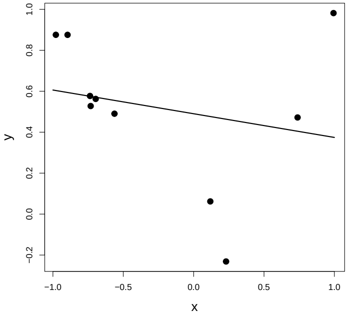
- We start with an informal example.
- We will formalize what we observe later.

Bias-variance tradeoff: intuition



- We observe 10 couples (x_i, y_i).
- We want to estimate y from x.
- **Our first strategy:** find f such that $f(x_i)$ is close to y_i.

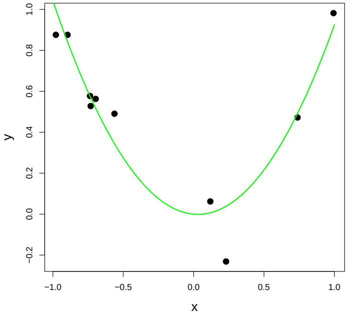
Bias-variance tradeoff: intuition



Find f as a line

$$\min_{f(x)=ax+b} \|Y - f(X)\|^2$$

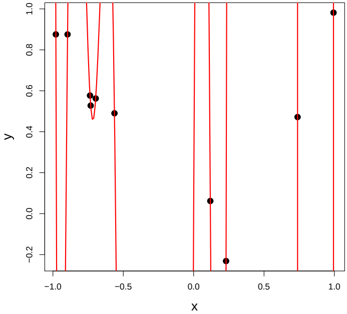
Bias-variance tradeoff: intuition



Find f as a quadratic function

$$\min_{f(x)=ax^2+bx+c} \|Y - f(X)\|^2$$

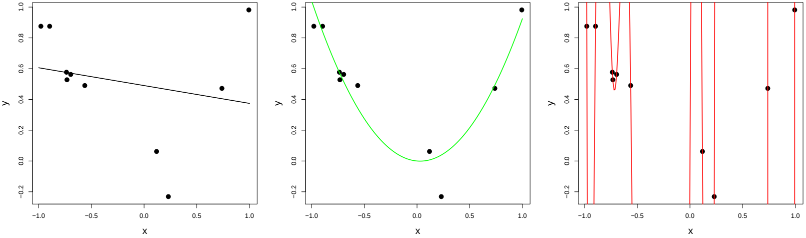
Bias-variance tradeoff: intuition



Find f as a polynomial of degree 10

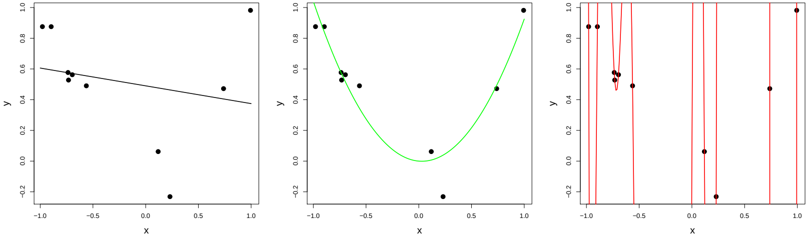
$$\min_{f(x)=\sum_{j=0}^{10} a_j x^j} \|Y - f(X)\|^2$$

Bias-variance tradeoff: intuition



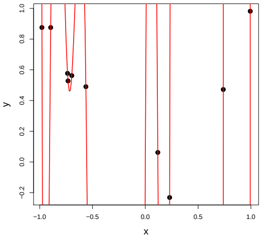
Which function would you trust to predict y corresponding to $x = 0.5$?

Bias-variance tradeoff: intuition



- Reminder: we aim at “finding f such that $f(x_i)$ is close to y_i”.
- With the polynomial of degree 10, $f(x_i) - y_i = 0$ for all 10 points.
- There is something wrong with our objective.

Bias-variance tradeoff: intuition

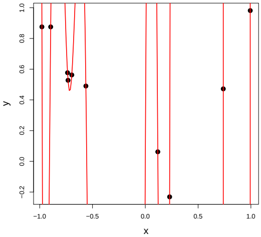


More precisely:

- If we allow **any** function f, we can find **a lot** of perfect solutions for the training data.
- Our actual goal is to estimate y for **new points** x from the same population :

$$\min_f \mathbb{E}_{(X,Y)} \| Y - f(X) \|^2$$

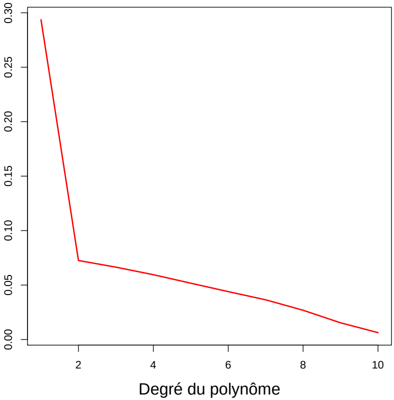
Biais-variance tradeoff: intuition



Even more precisely :

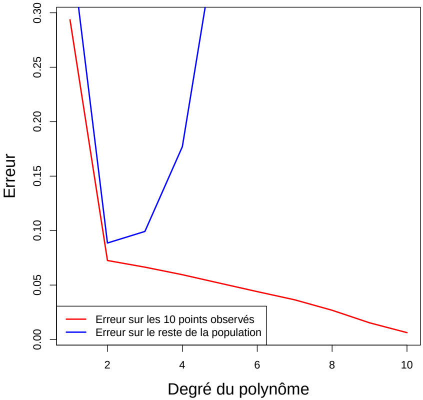
- We did not take into account the fact that our 10 points are a subsample from the population.
- If we sample 10 new points from the same population, the complex functions are likely to change more than the simple ones.
- Consequence: these fonctions will probably generalize less well to the rest of the population.

Overfitting



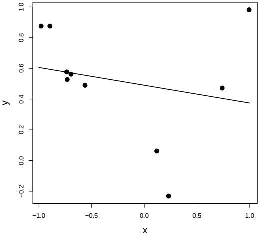
- When the degree increases, the error $\|y - f(x)\|^2$ over the 10 observations always decreases.
- Over the rest of the population, the error decreases, **then increases**.

Overfitting



- When the degree increases, the error $\|y - f(x)\|^2$ over the 10 observations always decreases.
- Over the rest of the population, the error decreases, **then increases**.

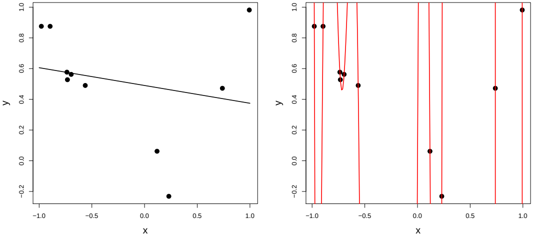
Overfitting



This suggests the existence of a **tradeoff** between two types of errors:

- Sets of functions which are too simple cannot contain functions which explain the data well enough.
- Sets of functions which are too rich may contain functions which are too specific to the observed sample.

Overfitting



This suggests the existence of a **tradeoff** between two types of errors:

- Sets of functions which are too simple cannot contain functions which explain the data well enough.
- Sets of functions which are too rich may contain functions which are too specific to the observed sample.

Parenthesis: complexity vs dimension (1/3)

- Our introductive examples had **a large number of descriptors**.
- This case involves increasingly **complex** functions of a single variable.

Parenthesis : complexity vs dimension (2/3)

- In fact, the two notions are related: here in particular, the three functions are linear in different representations.
- Reminder (linear regression): $\arg\min_{\theta \in \mathbb{R}^p} \|Y - X\theta\|^2 = (X^\top X)^{-1} X^\top Y$ (if $X^\top X$ is invertible).
- How can we use this fact to compute $\arg\min_{f(x)=\sum_{j=1}^{p} a_i x^j} \|Y - f(X)\|^2$?

Parenthesis : complexity vs dimension (3/3)

- We could have illustrated the same principle using linear functions involving more and more variables.
- Example : predicting a phenotype using the expression of an increasing number of genes.
- We sticked to polynomials, which allow for better visual representations.
- Along this class, the notion of complexity of a set of functions will become more and more precise.
- Complexity is what causes problems for inference, not just dimension.

Second parenthesis : models

- Until now, we did not need to introduce a **model** for the data, *i.e.*, a distribution over $\mathcal{X} \times \mathcal{Y}$:
  - Data could come from any population.
  - The functions we used to predict y can be derived from particular probabilistic models, but this is not necessary (they were in fact historically introduced without a model).
- The objective is not to criticize the use of models, but to show that the tradeoff problem we introduced goes beyond probabilistic models.
- We now show how using a model can give a better insight into the problem.

A little more formally: biais-variance decomposition

- We now assume that the data follow:

$$y = f(x) + \varepsilon, \tag{1}$$

  and $\mathbf{E}[\varepsilon] = 0$.

- Without loss of generality, we consider an estimator $\hat{f}$ of f, which is a function of training data $\mathcal{D} = (x_i, y_i)_{(i=1,\dots,n)}$ sampled i.i.d. from (1)
- Note: $\hat{f}$ is a random function.
- We consider the mean **quadratic error** $\mathbf{E}[(y - \hat{f}(x))^2]$ incurred when using $\hat{f}$ to estimate for a given x the corresponding y sampled from (1) independently from $\mathcal{D}$.
- Expectation is taken over $\mathcal{D}$ used to estimate $\hat{f}$, and $\varepsilon = y - f(x)$.

A little more formally: biais-variance decomposition

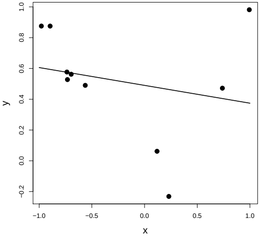
Proposition

Under the previous hypotheses,

$$\mathbf{E}[(y-\hat{f}(x))^2] = \left(\mathbf{E}[\hat{f}(x)] - f(x)\right)^2 + \mathbf{E}\left[\left(\mathbf{E}[\hat{f}(x)] - \hat{f}(x)\right)^2\right] + \mathbf{E}[(y-f(x))^2]$$

- The first term is the squared bias of $\hat{f}$: the difference between its mean (over the sample of $\mathcal{D}$) and the true f.
- The second term is the variance of $\hat{f}$: how much $\hat{f}$ varies around its average when the dataset $\mathcal{D}$ changes.
- The third term is the Bayes error, and does not depend on the estimator. The actual quantity of interest is the **excess of risk** $\mathbf{E}[(y-\hat{f}(x))^2] - \mathbf{E}[(y-f(x))^2]$.

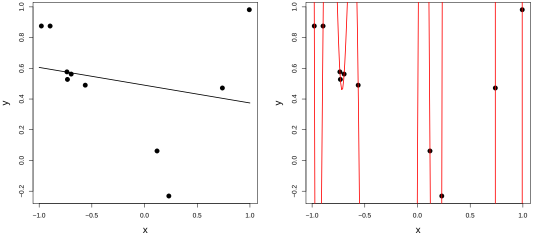
Back to our example



Tradeoff between two types of error:

- Sets of functions which are too simple cannot contain functions which explain the data well enough:
  these sets lead to estimators with a large bias.
- Sets of functions which are too rich may contain functions which are too specific to the observed sample:
  these sets lead to estimators with a large variance.

Back to our example



Tradeoff between two types of error:

- Sets of functions which are too simple cannot contain functions which explain the data well enough:
  these sets lead to estimators with a large bias.
- Sets of functions which are too rich may contain functions which are too specific to the observed sample:
  these sets lead to estimators with a large variance.

Biais-variance decomposition: proof

Reminder (König-Huygens)

For any real random variable Z, $\mathbf{E}\left[(Z-\mathbf{E}[Z])^2\right]=\mathbf{E}[Z^2]-\mathbf{E}[Z]^2$

$$\mathbf{E}[(y-\hat{f}(x))^2] = \mathbf{E}[y^2 - 2y\hat{f}(x) + \hat{f}(x)^2]$$

Biais-variance decomposition: proof

Reminder (König-Huygens)

For any real random variable Z, $\mathbf{E}\left[(Z-\mathbf{E}[Z])^2\right] = \mathbf{E}[Z^2] - \mathbf{E}[Z]^2$

$$\begin{aligned}\mathbf{E}[(y-\hat{f}(x))^2] =& \mathbf{E}[y^2 - 2y\hat{f}(x) + \hat{f}(x)^2] \\ =& \mathbf{E}[y^2] - \mathbf{E}[2y\hat{f}(x)] + \mathbf{E}[\hat{f}(x)^2]\end{aligned}$$

Biais-variance decomposition: proof

Reminder (König-Huygens)

For any real random variable Z, $\mathbf{E}\left[(Z-\mathbf{E}[Z])^2\right]=\mathbf{E}[Z^2]-\mathbf{E}[Z]^2$

$$\begin{aligned}
\mathbf{E}[(y-\hat{f}(x))^2] =& \mathbf{E}[y^2-2y\hat{f}(x)+\hat{f}(x)^2] \\
=& \mathbf{E}[y^2]-\mathbf{E}[2y\hat{f}(x)]+\mathbf{E}[\hat{f}(x)^2] \\
=& \mathbf{E}[y]^2+\mathbf{E}[(y-\mathbf{E}[y])^2] \\
& -2\mathbf{E}[y]\mathbf{E}[\hat{f}(x)] \\
& +\mathbf{E}[\hat{f}(x)]^2+\mathbf{E}[(\hat{f}(x)-\mathbf{E}[\hat{f}(x)])^2]
\end{aligned}$$

Biais-variance decomposition: proof

Reminder (König-Huygens)

For any real random variable Z, $\mathbf{E}\left[(Z-\mathbf{E}[Z])^2\right] = \mathbf{E}[Z^2] - \mathbf{E}[Z]^2$

$$\begin{aligned}
\mathbf{E}[(y-\hat{f}(x))^2] =& \mathbf{E}[y^2 - 2y\hat{f}(x) + \hat{f}(x)^2] \\
=& \mathbf{E}[y^2] - \mathbf{E}[2y\hat{f}(x)] + \mathbf{E}[\hat{f}(x)^2] \\
=& f(x)^2 + \mathbf{E}[(y-f(x))^2] \\
& - 2f(x)\mathbf{E}[\hat{f}(x)] \\
& + \mathbf{E}[\hat{f}(x)]^2 + \mathbf{E}[(\hat{f}(x) - \mathbf{E}[\hat{f}(x)])^2]
\end{aligned}$$

Biais-variance decomposition: proof

Reminder (König-Huygens)

For any real random variable Z, $\mathbf{E}\left[(Z - \mathbf{E}[Z])^2\right] = \mathbf{E}[Z^2] - \mathbf{E}[Z]^2$

$$
\begin{aligned}
\mathbf{E}[(y - \hat{f}(x))^2] =& \mathbf{E}[y^2 - 2y\hat{f}(x) + \hat{f}(x)^2] \\
=& \mathbf{E}[y^2] - \mathbf{E}[2y\hat{f}(x)] + \mathbf{E}[\hat{f}(x)^2] \\
=& f(x)^2 + \mathbf{E}[(y - f(x))^2] \\
& - 2f(x)\mathbf{E}[\hat{f}(x)] \\
& + \mathbf{E}[\hat{f}(x)]^2 + \mathbf{E}[(\hat{f}(x) - \mathbf{E}[\hat{f}(x)])^2] \\
=& \mathbf{E}[(y - f(x))^2] + \mathbf{E}[(\hat{f}(x) - \mathbf{E}[\hat{f}(x)])^2] \\
& + \Big(\mathbf{E}[\hat{f}(x)] - f(x)\Big)^2
\end{aligned}
$$

Biais-variance decomposition : perspective

$$\mathbf{E}[(y-\hat{f}(x))^2] = \left(\mathbf{E}[\hat{f}(x)] - f(x)\right)^2 + \mathbf{E}\left[\left(\mathbf{E}[\hat{f}(x)] - \hat{f}(x)\right)^2\right] + \mathbf{E}[(y-f(x))^2]$$

- Using a (rather general) model, we managed to start formalizing the tradeoff introduced with our example.
- Decomposition valid for any x, thus also in expectation over independent x.
- We now generalize this formalization.

A little more generally : structural risk minimization

- We now suppose more generally that the observations are sampled from a joint distribution $\mathbb{P}(x, y)$.
- This does not necessarily mean that we assume a particular probabilistic model: given a deterministic set of couples (x, y), $\mathbb{P}$ can be their empirical distribution.
- We also consider a **loss function**

  $$L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$$

  $L(y, y')$ quantifies the cost of the error made by predicting y' when the true value is y.
- Special case (our example): $L(y, y') = (y - y')^2$.

A little more generally : structural risk minimization

We look for an estimator $f : \mathcal{X} \to \mathcal{Y}$ minimizing

$$R(f) = \int_{\mathcal{X}\times\mathcal{Y}} L(y, f(x)) d\mathbb{P} = \mathbf{E}[L(y, f(x))]. \tag{2}$$

R is the **risk** of f : the average cost of using f to predict y from x over the joint distribution.

A little more generally : structural risk minimization

$$R(f) = \int_{\mathcal{X}\times\mathcal{Y}} L(y, f(x)) d\mathbb{P} = \mathbf{E}[L(y, f(x))]. \tag{3}$$

- The risk is minimized by the Bayes estimator $f(x) = \arg\min_{\hat{y}} \int_{\mathcal{Y}} L(y, \hat{y}) d\mathbb{P}(y|x)$.
- Generally the associated Bayes risk R^* is non-zero.
- The Bayes estimator is accessible only if $\mathbb{P}$ is known.

A little more generally : structural risk minimization

- In practice, we cannot compute $R(f)$ because the distribution $\mathbb{P}$ is unknown (otherwise we would simply use $\mathbb{P}(y|x)$ for prediction)
- We therefore use a training set ($\mathcal{D}$ in the previous example) to estimate R, for example through the **empirical risk**:

$$\hat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)). \tag{4}$$

- **Empirical risk minimization** : choose f minimizing $\hat{R}$.
- We saw in our example that minimizing the empirical risk was not enough to obtain a low risk R (overfitting)

A little more generally : structural risk minimization

- More generally, we can minimize the risk over **a function space** $\mathcal{H}$ (polynomials of a certain degree in our example).
- If R^* is the Bayes risk, we can decompose the **Bayes regret** :

$$R(f) - R^* = \left(R(f) - \inf_{g \in \mathcal{H}} R(g) \right) + \left(\inf_{g \in \mathcal{H}} R(g) - R^* \right). \qquad (5)$$

- The second term is the approximation error: the smallest excess of risk we can reach using a function of $\mathcal{H}$.
- This is a bias term, which does not depend on the data but only on the size of $\mathcal{H}$.
- The first term is the excess of risk of f with respect to the best function in $\mathcal{H}$.

A little more generally : structural risk minimization

- We consider $\hat{f}$ obtained by minimization of the empirical risk over $\mathcal{H}$:

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

- We want to bound the excess of risk $R(\hat{f}) - \inf_{g \in \mathcal{H}} R(g) \geq 0$
- This term (estimation error) can be decomposed:

$$\begin{aligned} R(\hat{f}) - \inf_{g \in \mathcal{H}} R(g) \overset{\Delta}{=} & R(\hat{f}) - R(f^*_{\mathcal{H}}) \\ = & R(\hat{f}) - \hat{R}(\hat{f}) \\ & + \hat{R}(\hat{f}) - \hat{R}(f^*_{\mathcal{H}}) \\ & + \hat{R}(f^*_{\mathcal{H}}) - R(f^*_{\mathcal{H}}). \end{aligned}$$

A little more generally : structural risk minimization

$$\begin{aligned} R(\hat{f}) - \inf_{g \in \mathcal{H}} R(g) =& R(\hat{f}) - R(f^*_{\mathcal{H}}) \\ =& R(\hat{f}) - \hat{R}(\hat{f}) \\ &+ \hat{R}(\hat{f}) - \hat{R}(f^*_{\mathcal{H}}) \\ &+ \hat{R}(f^*_{\mathcal{H}}) - R(f^*_{\mathcal{H}}). \end{aligned}$$

- Reminder :
  - $f^*_{\mathcal{H}}$ minimizes R, the **expected** risk w.r.t. $\mathbb{P}$, over $\mathcal{H}$.
  - The estimator $\hat{f}$ minimizs the **empirical** risk $\hat{R}$ over $\mathcal{H}$.
  - We therefore estimate at two levels: the function f and the risk R.

A little more generally : structural risk minimization

$$
\begin{aligned}
R(\hat{f}) - \inf_{g \in \mathcal{H}} R(g) = {} & R(\hat{f}) - \hat{R}(\hat{f}) \\
& + \hat{R}(\hat{f}) - \hat{R}(f^*_{\mathcal{H}}) \\
& + \hat{R}(f^*_{\mathcal{H}}) - R(f^*_{\mathcal{H}}).
\end{aligned}
$$

- The first term is the difference between the true risk and the estimated risk, for our estimator $\hat{f}$ of f.
- This is a complex object to study. **Statistical learning theory** (Vapnik and Chervonenkis) aims at bounding this quantity as a function of n and the complexity of $\mathcal{H}$.
- The second term is nonpositive by construction.
- The third one is easier to control as it involves a deterministic function and the law of large numbers applies.

A little more generally : structural risk minimization

We can however bound the first term:

$$R(\hat{f}) - \hat{R}(\hat{f}) \leq \sup_{f \in \mathcal{H}} \left| \mathbf{E}[L(y, f(x))] - \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) \right| ,$$

and since this quantity also bounds the third term, we get

$$R(\hat{f}) - \inf_{g \in \mathcal{H}} R(g) \leq 2 \sup_{f \in \mathcal{H}} \left| \mathbf{E}[L(y, f(x))] - \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) \right| .$$

- This bound of the estimation error suggests that it corresponds to a variance term, which increases with the size of $\mathcal{H}$.
- The more complex $\mathcal{H}$ is, the more likely it is to contain a function for which the empirical risk and the population risk are very different.

A little more generally : structural risk minimization

We can make this notion of size more precise by introducing the **Rademacher complexity** of $\mathcal{H}$:

Definition

Let ϵ_i, $i = 1, \dots, n$ i.i.d such that $\mathbb{P}(\epsilon_i = 1) = \mathbb{P}(\epsilon_i = -1) = 1/2$, Z_i, $i = 1, \dots, n$ i.i.d data and $\mathcal{H}$ a space of functions defined over this data, then

$$\mathfrak{R}_n(\mathcal{H}) = \mathbf{E}_{\epsilon_1^n, Z_1^n} \left[\sup_{f \in \mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i f(Z_i) \right| \right]$$

is the Rademacher complexity of $\mathcal{H}$.

Intuition: $\mathfrak{R}_n$ measures the capacity of $\mathcal{H}$ to provide functions which align with noise.

A little more generally : structural risk minimization

We can make this notion of size more precise by introducing the **Rademacher complexity** of $\mathcal{H}$:

Definition

Let ϵ_i, $i = 1, \dots, n$ i.i.d such that $\mathbb{P}(\epsilon_i = 1) = \mathbb{P}(\epsilon_i = -1) = 1/2$, Z_i, $i = 1, \dots, n$ i.i.d data and $\mathcal{H}$ a space of functions defined over this data, then

$$\mathfrak{R}_n(\mathcal{H}) = \mathbf{E}_{\epsilon_1^n, Z_1^n}\left[\sup_{f \in \mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i f(Z_i) \right| \right]$$

is the Rademacher complexity of $\mathcal{H}$.

This complexity increases with the size of $\mathcal{H}$ and decreases with the size n of the sample.

A little more generally : structural risk minimization

We can bound the mean estimation error in terms of the Rademacher complexity of $\mathcal{H}$.

Proposition

$$\mathbf{E}_{(x,y)_1^n} \sup_{f \in \mathcal{H}} \left| \mathbf{E}_{(x,y)}[L(y, f(x))] - \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) \right| \leq 2\mathfrak{R}_n(\mathcal{H}).$$

Therefore,

$$\mathbf{E}_{(x,y)_1^n} \left[R(\hat{f}) - R^* \right] \leq \left(\min_{g \in \mathcal{H}} R(g) - R^* \right) + 4\mathfrak{R}_n(\mathcal{H}).$$

A little more generally : structural risk minimization

Therefore

$$\mathbf{E}_{(x,y)_1^n}\left[R(\hat{f}) - R^*\right] \leq \left(\min_{g\in\mathcal{H}} R(g) - R^*\right) + 4\mathfrak{R}_n(\mathcal{H}),$$

- This result illustrates a little more generally the bias variance tradeoff for risk minimization.
- It makes explicit the link between complexity and sample size: lots of points are needed to estimate in large $\mathcal{H}$ (otherwise $\mathfrak{R}_n(\mathcal{H})$ is large).

ERM consistency and SRM

Therefore

$$\mathbf{E}_{(x,y)_1^n}\left[R(\hat{f}) - R^*\right] \leq \left(\min_{g\in\mathcal{H}} R(g) - R^*\right) + 4\mathfrak{R}_n(\mathcal{H}),$$

Concretely, this analysis is at the core of two major elements of statistical learning (Vapnik and Chervonenkis, late 60's):

- It is used in learning theory to establish consistency of empirical risk minimization: only families with bounded complexity allow to learn by ERM (are consistent).
- **It also suggests a strategy to design estimators**: build small classes $\mathcal{H}$ which we think contain good approximations.

A little more generally : structural risk minimization

$$\mathbf{E}_{(x,y)_1^n}\left[R(\hat{f}) - R^*\right] \leq \left(\min_{g\in\mathcal{H}} R(g) - R^*\right) + 4\mathfrak{R}_n(\mathcal{H}),$$

Practical procedure proposed by Vapnik and Chervonenkis: **structural risk minimization**:

1. Define nested function sets of increasing complexity.
2. Minimize the empirical risk over each family.
3. Choose the solution giving the best generalization guarantees.

A little more generally : structural risk minimization

Structural risk minimization:

1. Define nested function sets of increasing complexity.
2. Minimize the empirical risk over each family.
3. Choose the solution giving the best generalization guarantees.

We will study practical instances of this strategy later in this class.

A little more generally : structural risk minimization

Proof of the previous bound (inspired from Peter Bartlett's slides)

$$\mathbf{E}_{(x,y)_1^n} \sup_{f \in \mathcal{H}} \left| \mathbf{E}_{(x,y)}[L(y, f(x))] - \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) \right|$$

A little more generally : structural risk minimization

Proof of the previous bound (inspired from Peter Bartlett's slides)

$$
\begin{aligned}
&\mathbf{E}_{(x,y)_1^n} \sup_{f\in\mathcal{H}} \left| \mathbf{E}_{(x,y)}[L(y,f(x))] - \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) \right| \\
&= \mathbf{E}_{(x,y)_1^n} \sup_{f\in\mathcal{H}} \left| \mathbf{E}_{(x',y')_1^n} \left[\frac{1}{n}\sum_{i=1}^{n} L(y_i', f(x_i')) \right] - \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) \right|
\end{aligned}
$$

A little more generally : structural risk minimization

Proof of the previous bound (inspired from Peter Bartlett's slides)

$$
\begin{aligned}
&\mathbf{E}_{(x,y)_1^n} \sup_{f\in\mathcal{H}} \left| \mathbf{E}_{(x,y)}[L(y,f(x))] - \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) \right| \\
&= \mathbf{E}_{(x,y)_1^n} \sup_{f\in\mathcal{H}} \left| \mathbf{E}_{(x',y')_1^n} \left[\frac{1}{n}\sum_{i=1}^{n} L(y'_i, f(x'_i)) \right] - \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) \right| \\
&= \mathbf{E}_{(x,y)_1^n} \sup_{f\in\mathcal{H}} \left| \mathbf{E}_{(x',y')_1^n} \left[\frac{1}{n}\sum_{i=1}^{n} L(y'_i, f(x'_i)) - \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) \right] \right|
\end{aligned}
$$

A little more generally : structural risk minimization

Proof of the previous bound (inspired from Peter Bartlett's slides)

$$\mathbf{E}_{(x,y)_1^n} \sup_{f\in\mathcal{H}} \left| \mathbf{E}_{(x,y)}[L(y,f(x))] - \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) \right|$$

$$= \mathbf{E}_{(x,y)_1^n} \sup_{f\in\mathcal{H}} \left| \mathbf{E}_{(x',y')_1^n} \left[\frac{1}{n}\sum_{i=1}^{n} L(y'_i, f(x'_i)) \right] - \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) \right|$$

$$= \mathbf{E}_{(x,y)_1^n} \sup_{f\in\mathcal{H}} \left| \mathbf{E}_{(x',y')_1^n} \left[\frac{1}{n}\sum_{i=1}^{n} L(y'_i, f(x'_i)) - \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) \right] \right|$$

$$= \mathbf{E}_{(x,y)_1^n} \sup_{f\in\mathcal{H}} \left| \mathbf{E}_{(x',y')_1^n} \left[\frac{1}{n}\sum_{i=1}^{n} L(y'_i, f(x'_i)) - L(y_i, f(x_i)) \right] \right|$$

A little more generally : structural risk minimization

Proof of the previous bound (inspired from Peter Bartlett's slides)

$$
\begin{aligned}
&\mathbf{E}_{(x,y)_1^n} \sup_{f\in\mathcal{H}} \left| \mathbf{E}_{(x,y)}[L(y, f(x))] - \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) \right| \\
&= \mathbf{E}_{(x,y)_1^n} \sup_{f\in\mathcal{H}} \left| \mathbf{E}_{(x',y')_1^n} \left[\frac{1}{n}\sum_{i=1}^{n} L(y_i', f(x_i')) \right] - \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) \right| \\
&= \mathbf{E}_{(x,y)_1^n} \sup_{f\in\mathcal{H}} \left| \mathbf{E}_{(x',y')_1^n} \left[\frac{1}{n}\sum_{i=1}^{n} L(y_i', f(x_i')) - \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) \right] \right| \\
&= \mathbf{E}_{(x,y)_1^n} \sup_{f\in\mathcal{H}} \left| \mathbf{E}_{(x',y')_1^n} \left[\frac{1}{n}\sum_{i=1}^{n} L(y_i', f(x_i')) - L(y_i, f(x_i)) \right] \right| \\
&\leq \mathbf{E}_{(x,y)_1^n} \mathbf{E}_{(x',y')_1^n} \left[\sup_{f\in\mathcal{H}} \left| \frac{1}{n}\sum_{i=1}^{n} L(y_i', f(x_i')) - L(y_i, f(x_i)) \right| \right]
\end{aligned}
$$

A little more generally : structural risk minimization

We now introduce ϵ_i, $i = 1, \ldots, n \in \{-1, 1\}$. Notice that

$$\begin{aligned}
&\mathbf{E} \sup_{f \in \mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^{n} L(y_i', f(x_i')) - L(y_i, f(x_i)) \right| \\
&= \mathbf{E} \sup_{f \in \mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \left(L(y_i', f(x_i')) - L(y_i, f(x_i)) \right) \right|,
\end{aligned}$$

since the data is i.i.d, switching the two terms does not affect the distribution of the sup.
The equality holds for any choice of ϵ_i, so we can take the expectation over a uniform i.i.d choice.

A little more generally : structural risk minimization

Finally,

$$\begin{aligned}
&\mathbf{E} \sup_{f\in\mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i \left(L(y_i', f(x_i')) - L(y_i, f(x_i)) \right) \right| \\
&\leq \mathbf{E} \sup_{f\in\mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i L(y_i', f(x_i')) \right| + \mathbf{E} \sup_{f\in\mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i L(y_i, f(x_i)) \right| \\
&= 2\mathbf{E} \sup_{f\in\mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i L(y_i, f(x_i)) \right| = 2\mathfrak{R}_n(\mathcal{H}).
\end{aligned}$$

This proof technique is called **symmetrization**.

More intuition about the complexity of a set of functions: VC dimension

- In practice, we sometimes use VC dimension $\nu(\mathcal{H})$ of a set of functions to bound the Rademacher complexity: $\mathfrak{R}_n(\mathcal{H}) \leq C\sqrt{\nu(\mathcal{H})/n}$
- We restrict ourselves to the sets $\mathcal{H}$ of binary valued functions (useful for classification).
- We say a set $Z = (Z_1, \ldots, Z_n)$ is **shattered** by $\mathcal{H}$ if $\mathrm{Card}\,\{f(Z_1), \ldots, f(Z_n) | f \in \mathcal{H}\} = 2^n$.
- Interpretation: we can find an $f \in \mathcal{H}$ assigning 0 to any subset of Z and 1 to its complement.
- The VC dimension $\nu(\mathcal{H})$ of $\mathcal{H}$ is the largest integer n such that there exists a set $(Z_1, \ldots, Z_n)$ shattered by $\mathcal{H}$.

More intuition about the complexity of a set of functions: VC dimension

- We extend the VC dimension to real valued functions by thresholding functions at 0.
- Linear functions in p dimensions: $\mathcal{H}_L = \{f_\theta(x) = sign(\theta^\top x), \theta \in \mathbb{R}^p\}$.
- Includes linear functions and polynomials in our introduction.
- We can show that $\nu(\mathcal{H}_L) = p$.

More intuition about the complexity of a set of functions: VC dimension

- Proof of $\nu(\mathcal{H}_L) \geq p$: we build a set of p points in p dimensions shattered by a function of $\mathcal{H}_L$.
- Let $\mathcal{E}_p$ be the canonical basis of $\mathbb{R}^p$. For any set $y \in \{-1,+1\}^p$ and any $i = 1,\ldots,n$, $f_\theta(e_i) = y_i$ by choosing $\theta_i = y_i$.
- Proof of $\nu(\mathcal{H}_L) < p+1$: no set of $p+1$ points in p dimensions can be shattered by a linear function.

More intuition about the complexity of a set of functions: VC dimension

- Proof of $\nu(\mathcal{H}_L) \geq p$: we build a set of p points in p dimensions shattered by a function of $\mathcal{H}_L$.
- Let $\mathcal{E}_p$ be the canonical basis of $\mathbb{R}^p$. For any set $y \in \{-1, +1\}^p$ and any $i = 1, \ldots, n$, $f_\theta(e_i) = y_i$ by choosing $\theta_i = y_i$.
- Proof of $\nu(\mathcal{H}_L) < p + 1$: no set of $p + 1$ points in p dimensions can be shattered by a linear function.

More intuition about the complexity of a set of functions: VC dimension

- Proof of $\nu(\mathcal{H}_L) \geq p$: we build a set of p points in p dimensions shattered by a function of $\mathcal{H}_L$.
- Let $\mathcal{E}_p$ be the canonical basis of $\mathbb{R}^p$. For any set $y \in \{-1, +1\}^p$ and any $i = 1, \ldots, n$, $f_\theta(e_i) = y_i$ by choosing $\theta_i = y_i$.
- Proof of $\nu(\mathcal{H}_L) < p + 1$: no set of $p + 1$ points in p dimensions can be shattered by a linear function.

More intuition about the complexity of a set of functions: VC dimension

- Let $x_1, \ldots, x_{p+1} \in \mathbb{R}^p$. One of the points can necessarily be written as a linear combination of the p others.

More intuition about the complexity of a set of functions: VC dimension

- Let $x_1, \ldots, x_{p+1} \in \mathbb{R}^p$. One of the points can necessarily be written as a linear combination of the p others.
- Without loss of generality, let us write $x_{p+1} = \sum_{i=1}^{p} \alpha_i x_i$ and $f_\theta(x_{p+1}) = \sum_{i=1}^{p} \alpha_i \theta^\top x_i$.

More intuition about the complexity of a set of functions: VC dimension

- Let $x_1, \ldots, x_{p+1} \in \mathbb{R}^p$. One of the points can necessarily be written as a linear combination of the p others.
- Without loss of generality, let us write $x_{p+1} = \sum_{i=1}^{p} \alpha_i x_i$ and $f_\theta(x_{p+1}) = \sum_{i=1}^{p} \alpha_i \theta^\top x_i$.
- Let $y = (sign(\alpha_1), \ldots, sign(\alpha_p), -1)$, and assume there exists $\theta \in \mathbb{R}^p$ such that $sign(\theta^\top x_i) = y_i, i = 1, \ldots, p$.

More intuition about the complexity of a set of functions: VC dimension

- Let $x_1, \ldots, x_{p+1} \in \mathbb{R}^p$. One of the points can necessarily be written as a linear combination of the p others.
- Without loss of generality, let us write $x_{p+1} = \sum_{i=1}^{p} \alpha_i x_i$ and $f_\theta(x_{p+1}) = \sum_{i=1}^{p} \alpha_i \theta^\top x_i$.
- Let $y = (sign(\alpha_1), \ldots, sign(\alpha_p), -1)$, and assume there exists $\theta \in \mathbb{R}^p$ such that $sign(\theta^\top x_i) = y_i, i = 1, \ldots, p$.
- Then necessarily $sign(\theta^\top x_{p+1}) = sign(\sum_{i=1}^{p} \alpha_i \theta^\top x_i) = 1$ since $sign(\theta^\top x_i) = sign(\alpha_i), i = 1, \ldots, p$.

More intuition about the complexity of a set of functions: VC dimension

- Let $x_1, \dots, x_{p+1} \in \mathbb{R}^p$. One of the points can necessarily be written as a linear combination of the p others.
- Without loss of generality, let us write $x_{p+1} = \sum_{i=1}^{p} \alpha_i x_i$ and $f_\theta(x_{p+1}) = \sum_{i=1}^{p} \alpha_i \theta^\top x_i$.
- Let $y = (sign(\alpha_1), \dots, sign(\alpha_p), -1)$, and assume there exists $\theta \in \mathbb{R}^p$ such that $sign(\theta^\top x_i) = y_i, i = 1, \dots, p$.
- Then necessarily $sign(\theta^\top x_{p+1}) = sign(\sum_{i=1}^{p} \alpha_i \theta^\top x_i) = 1$ since $sign(\theta^\top x_i) = sign(\alpha_i), i = 1, \dots, p$.
- y can therefore not be obtained by any function of $\mathcal{H}_L$, and no set of $p+1$ vectors in $\mathbb{R}^p$ is shattered by $\mathcal{H}_L$.

Summary

- We saw how the risk could generally be decomposed as a term of bias/approximation and a term of variance/estimation.
- This decomposition highlights the tradeoff that needs to be dealt with in inference. This tradeoff is related to the complexity of the set of functions under consideration:
  - Sets too simple lead to a large approximation error.
  - Sets too large lead to a large estimation error.
- We defined this notion of complexity more precisely (Rademacher, VC), and saw it also depended on the number of samples.
- These ideas are crucial in modern applications, where we sometimes have few samples in high dimension.

Summary

- We saw how the risk could generally be decomposed as a term of bias/approximation and a term of variance/estimation.
- This decomposition highlights the tradeoff that needs to be dealt with in inference. This tradeoff is related to the complexity of the set of functions under consideration:
  - Sets too simple lead to a large approximation error.
  - Sets too large lead to a large estimation error.
- We defined this notion of complexity more precisely (Rademacher, VC), and saw it also depended on the number of samples.
- These ideas are crucial in modern applications, where we sometimes have few samples in high dimension.

Summary

- We saw how the risk could generally be decomposed as a term of bias/approximation and a term of variance/estimation.
- This decomposition highlights the tradeoff that needs to be dealt with in inference. This tradeoff is related to the complexity of the set of functions under consideration:
  - Sets too simple lead to a large approximation error.
  - Sets too large lead to a large estimation error.
- We defined this notion of complexity more precisely (Rademacher, VC), and saw it also depended on the number of samples.
- These ideas are crucial in modern applications, where we sometimes have few samples in high dimension.

Summary

- We saw how the risk could generally be decomposed as a term of bias/approximation and a term of variance/estimation.
- This decomposition highlights the tradeoff that needs to be dealt with in inference. This tradeoff is related to the complexity of the set of functions under consideration:
  - Sets too simple lead to a large approximation error.
  - Sets too large lead to a large estimation error.
- We defined this notion of complexity more precisely (Rademacher, VC), and saw it also depended on the number of samples.
- These ideas are crucial in modern applications, where we sometimes have few samples in high dimension.