Fundamentals of Machine Learning - Intro - M2 IASD

Yann Chevaleyre (based partly on D.Rosenberg Lectures)

September 28, 2023

Outline

1. Supervised Learning: a Formal Setting
2. Empirical Risk, True risks, and the Empirical Risk Minimization Principle
   - Empirical Risk and the ERM
   - Overfitting
   - The True Risk
3. Decomposing the Risk of the ERM
   - Error Decomposition
   - The Bayes Error
   - Approximation and Estimation error
   - Warmup: Bounding the estimation error in a simple case
4. No Learner is universal: No Free Lunch thm
5. The Risk of k-Nearest Neighbors

Outline

1. Supervised Learning: a Formal Setting
2. Empirical Risk, True risks, and the Empirical Risk Minimization Principle
    - Empirical Risk and the ERM
    - Overfitting
    - The True Risk
3. Decomposing the Risk of the ERM
    - Error Decomposition
    - The Bayes Error
    - Approximation and Estimation error
    - Warmup: Bounding the estimation error in a simple case
4. No Learner is universal: No Free Lunch thm
5. The Risk of k-Nearest Neighbors

Formalization: The Spaces

The Three Spaces:

- Input space: $\mathcal{X}$
- label space: $\mathcal{Y}$
- prediction space: $\hat{\mathcal{Y}}$

Concept check:

- 1. What are the spaces for linear regression?
- 2. What are the spaces for a bi-class classification problem ?
- 3. What are the spaces for a bi-class learning problem where we predict probabilities for each class ?

Some Formalization

The learner takes a data set as input:

A **data set** (a **batch**) $S=\{(x_1,y_1)\dots(x_N,y_N)\}$

The learner has a Loss Function

A **loss function** evaluates how prediction $\hat{y}$ fits the label y.

$$\begin{array}{rccc} \ell: & \hat{\mathcal{Y}}\times\mathcal{Y} & \to & \mathbb{R} \\ & (\hat{y},y) & \mapsto & \ell(\hat{y},y) \end{array}$$

The learner outputs a Prediction Function (or “classifier”)

A **prediction function** (or classifier) gets input $x\in\mathcal{X}$ and outputs $\hat{y}\in\hat{\mathcal{Y}}$:

$$\begin{array}{rccc} f: & \mathcal{X} & \to & \hat{\mathcal{Y}} \\ & x & \mapsto & f(x) \end{array}$$

(Important) examples of loss functions

- Least Square Regression:
    - Spaces: $\hat{\mathcal{Y}} = \mathcal{Y} = \mathbb{R}$
    - Square loss:

$$\ell^{sq}(\hat{y}, y) = (\hat{y} - y)^2$$

- Multiclass Classification:Spaces:
    - $\hat{\mathcal{Y}} = \mathcal{Y} = \{1, \ldots, k\}$
    - 0-1 loss:

$$\ell^{0/1}(\hat{y}, y) = 1(\hat{y} \neq y) := \begin{cases} 1 & \text{if } \hat{y} \neq y \\ 0 & \text{otherwise.} \end{cases}$$

Outline

1. Supervised Learning: a Formal Setting
2. Empirical Risk, True risks, and the Empirical Risk Minimization Principle
    - Empirical Risk and the ERM
    - Overfitting
    - The True Risk
3. Decomposing the Risk of the ERM
    - Error Decomposition
    - The Bayes Error
    - Approximation and Estimation error
    - Warmup: Bounding the estimation error in a simple case
4. No Learner is universal: No Free Lunch thm
5. The Risk of k-Nearest Neighbors

Empirical Risk, ERM algorithms

- Loss function ℓ evaluates a single prediction. How to evaluate the prediction function as a whole ? Let $S = ((x_1, y_1), \ldots, (x_N, y_N))$

Definition

The **empirical risk** of $f : \mathcal{X} \rightarrow \hat{\mathcal{Y}}$ with respect to S is $\hat{R}_N(f) = \frac{1}{N} \sum_{i=1}^{N} \ell(f(x_i), y_i)$.

Definition

A learning algorithm is an ERM (empirical risk minimizing) algorithm if it outputs a function $\hat{f}$ minimizing the empirical risk

$$\hat{f} \in \arg\min_{f \in \mathcal{F}} \hat{R}_N(f)$$

- A few algorithms are strictly ERM (e.g. ordinary linear regression)
- k-Nearest neighbors is not (because $\mathcal{F}$ does not exist)

Empirical Risk, ERM algorithms

Definition

A learning algorithm is an ERM (empirical risk minimizing) algorithm if it outputs a function $\hat{f}$ minimizing the empirical risk

$$\hat{f} \in \arg\min_{f \in \mathcal{F}} \hat{R}_N(f) \text{ where } \hat{R}_N(f) = \frac{1}{N} \sum_{i=1}^{N} \ell(f(x_i), y_i)$$

Concept check: What is the minimization problem of an ERM algorithm

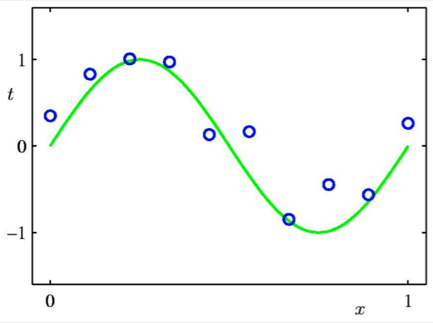
- for the linear regression with squared loss ?
- for linear classification with 0-1 loss ?

Are ERM Algorithms good ? Polynomial Curve Fitting Example

- A polynomial model $f(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M$
- $\mathcal{X} = \mathcal{Y} = \hat{\mathcal{Y}} = \mathbb{R}$
- The square loss $\ell^{sq}(\hat{y}, y) = (\hat{y} - y)^2$
- **Learning algorithms** find the best **parameters** $w_0, w_1, \ldots, w_M$.
- M is a (hyper-)parameter that is set by an expert

Example: Polynomial Curve Fitting

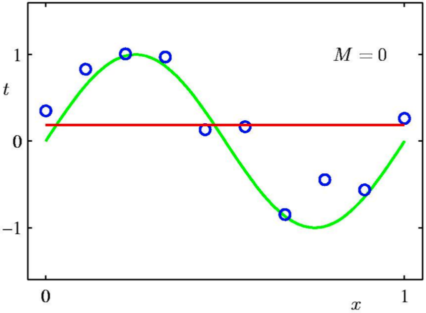
- Green curve is truth ($Y = \sin(2\pi x) + \epsilon$)



[1] From Bishop's *Pattern Recognition and Machine Learning*, Ch 1.

Example: Polynomial Curve Fitting

- Fit with $M = 0$:

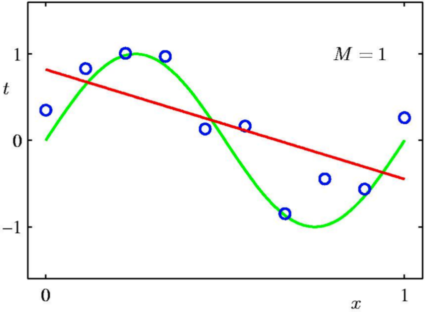


UNDERFIT (not fitting data well enough)

From Bishop's *Pattern Recognition and Machine Learning*, Ch 1.

Example: Polynomial Curve Fitting

- Fit with $M = 1$

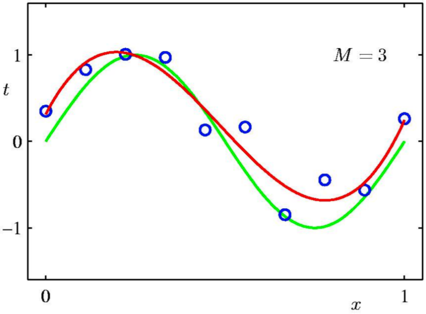


UNDERFIT (not fitting data well enough)

From Bishop's *Pattern Recognition and Machine Learning*, Ch 1.

Example: Polynomial Curve Fitting

- Fit with $M = 3$

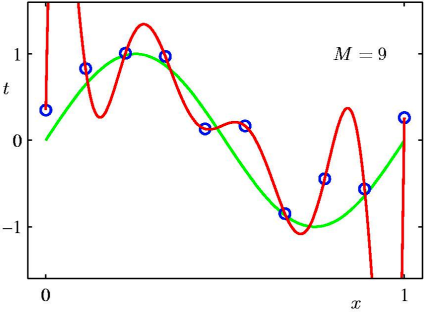


PRETTY GOOD!

From Bishop's *Pattern Recognition and Machine Learning*, Ch 1.

Example: Polynomial Curve Fitting

- Fit with $M = 9$

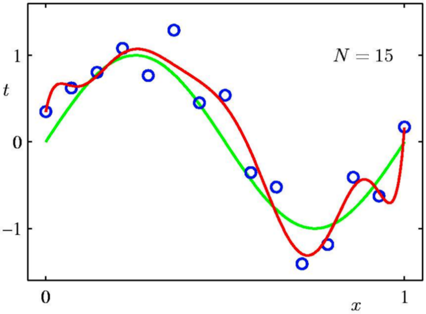


OVERFIT (fits data **too well**: $\hat{R}(f) = 0$, but f is “not good”)

From Bishop’s *Pattern Recognition and Machine Learning*, Ch 1.

Example: Polynomial Curve Fitting

- Fit with $M = 9$ (more data)

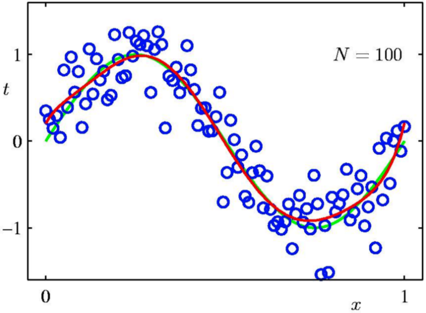


Pretty good - slightly overfit?

From Bishop's *Pattern Recognition and Machine Learning*, Ch 1.

Example: Polynomial Curve Fitting

- Fit with $M = 9$ (even more data)



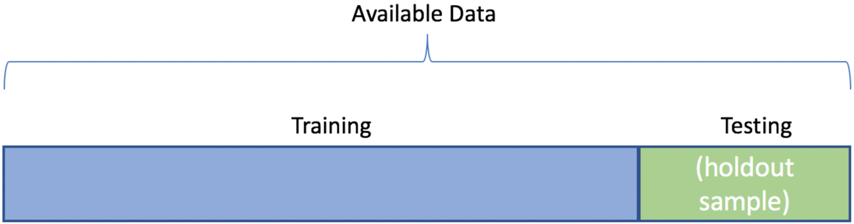
From Bishop's *Pattern Recognition and Machine Learning*, Ch 1.

Critique of ERM : Overfitting

- Loosely speaking, we say a model **overfits** when
  - training performance is good but **performance on other data from the same source is poor.**
  - Overfitting appears when
    - Model complexity is to high (here, M)
    - not enough data
- In general a low empirical risk does not garantee a good performance on future data
- **ERM alone may be a bad learning principle, because of overfitting**

Checking Overfitting

- We can estimate whether $\hat{f}$ overfits ***after*** the training (ERM,...) with the train-test split method:



but NOT during training !!!

Measuring performance of a classifier: Necessary assumptions

- We are interested in the performance of our classifier on *future data coming from the same source*. What does "same source" mean ?
- In statistics, we make very strong assumptions **(e.g. fixed design setting)**
- In machine learning, we make the following weak assumption:
  - Assume there exists an unknown **data generating distribution** P over $\mathcal{X} \times \mathcal{Y}$.
  - All input/output pairs (x, y), including pairs from the dataset and future data, are generated i.i.d. from a distribution P

Measuring performance of a classifier: the true risk

Definition

The **risk** of a prediction function $f : \mathcal{X} \to \hat{\mathcal{Y}}$ is

$$R(f) = \mathbb{E}_{X,Y}\left[\ell(f(X), Y)\right] = \int_{\mathcal{X},\mathcal{Y}} \ell(f(X), Y) dP(X, Y)$$

In words, it's the **expected loss** of f on a new example (X, Y) drawn randomly from P.

What we really want is a classifier minimizing the risk, not the empirical risk !!
But risk function cannot be computed
Thus, ERM is good if and only if the risk is close to the empirical risk

Risk vs Empirical risk

Let $S = ((x_1, y_1), \ldots, (x_N, y_N))$ be drawn i.i.d. from P.

- Let's draw some inspiration from the Strong Law of Large Numbers: If $z, z_1, \ldots, z_n$ are i.i.d. with expected value $\mathbb{E}z$, then w.p. one,

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} z_i = \mathbb{E}z,$$

- By the Strong Law of Large Numbers, if f **is independant from** S then

$$\lim_{N \to \infty} \hat{R}_N(f) = R(f),$$

- ERM seeks to find the $\hat{f}$ that **minimizes** $\hat{R}_N(f)$ - so independence **does not hold !!!** so in general, $\hat{R}_N(\hat{f}) \nrightarrow R(\hat{f})$. But after the training, with the train-test split, we can estimate the true risk of $\hat{f}$.

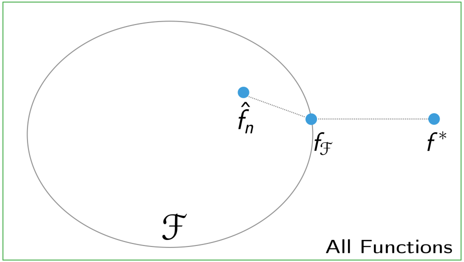
Recap on ERM

- We can estimate the empirical risk $\hat{R}(f)$ but we are interested in the true risk $R(f)$
- We saw ERM that can overfit because it optimizes $\hat{R}(f)$ which can be far from $R(f)$.
- If the number of examples N is big enough, will $R(\hat{f})$ converge to the best possible classifier in the class $\mathcal{F}$?
- What is the "best classifier" ?

Outline

1. Supervised Learning: a Formal Setting
2. Empirical Risk, True risks, and the Empirical Risk Minimization Principle
   - Empirical Risk and the ERM
   - Overfitting
   - The True Risk
3. Decomposing the Risk of the ERM
   - Error Decomposition
   - The Bayes Error
   - Approximation and Estimation error
   - Warmup: Bounding the estimation error in a simple case
4. No Learner is universal: No Free Lunch thm
5. The Risk of k-Nearest Neighbors

Error Decomposition of ERM



$$f^* = \arg\min_{f} \mathbb{E}_{X,Y} \ell(f(X), Y)$$

$$f_{\mathcal{F}} = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{X,Y} \ell(f(X), Y))$$

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

- **Approximation Error** (of $\mathcal{F}$) $=$ $R(f_{\mathcal{F}}) - R(f^*)$
- **Estimation error** (of $\hat{f}_n$ in $\mathcal{F}$) $=$ $R(\hat{f}_n) - R(f_{\mathcal{F}})$

Error Decomposition of ERM

- The risk of ERM $\hat{f}_n$ can be decomposed:

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

- *Bigger* $\mathcal{F}$ mean *smaller* approximation error but bigger estimation error

Chosing the right classifier space: Decomposition of the risk in ERM case

The Bayes Predictor

Definition

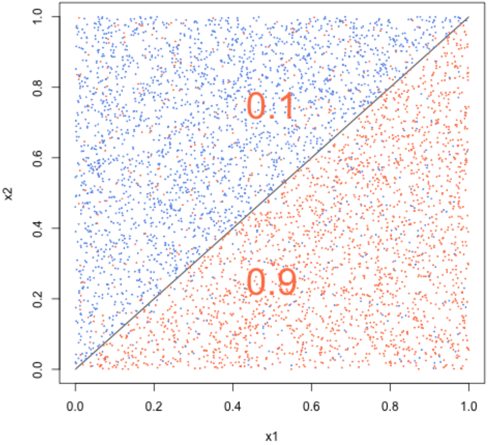
A **Bayes predictor** $f^* : \mathcal{X} \to \hat{\mathcal{Y}}$ is a function that achieves the *minimal risk* among all measurable functions:

$$f^* \in \underset{f \in \{\text{measurable functions}\}}{\arg\min} R(f),$$

where the minimum is taken over all functions from $\mathcal{X}$ to $\hat{\mathcal{Y}}$.

- The risk of a Bayes prediction function is called the **Bayes risk**.
- A Bayes prediction function is often called the "**target function**", since it's the best prediction function we can possibly produce.

An illustration of the notion of Bayes risk with $\ell^{0/1}$



$$\begin{aligned}\mathcal{Y} &= \{\text{blue}, \text{orange}\}\\ P_{\mathcal{X}} &= \text{Uniform}([0,1]^2)\end{aligned}$$

$$\begin{aligned}\mathbb{P}(\text{orange} \mid x^{(1)} > x^{(2)}) &= .9\\ \mathbb{P}(\text{orange} \mid x^{(1)} \leqslant x^{(2)}) &= .1\end{aligned}$$

$$\text{Risk of } f(x) = \begin{cases} \textit{orange} & \textit{if } x^{(1)} > x^{(2)} \\ \textit{blue} & \textit{otherwise} \end{cases} = ?$$

Bayes Risk $= ?$

A reminder on the Expectation and its properties (1/3)

- For any predicate C, we have $\mathbb{E}[1(C)] = \mathbb{P}(C)$ where $1(\cdot)$ is the indicator function
- As usual, $\mathcal{X}$ and $\mathcal{Y}$ are the domain of the random variables X and Y. Let $g(X,Y)$ be an arbitrary real-valued function.
- If $\mathcal{X}$ and $\mathcal{Y}$ are continuous spaces, assuming their distribution P admits a joint probability density function $p(X,Y)$, then:
  - The **expectation** of $g(X,Y)$ is:

$$\mathbb{E}_{X,Y}[g(X,Y)] = \int_{\mathcal{X}\times\mathcal{Y}} g(X,Y)p(X,Y)dXdY$$

  - The **conditional expectation** of $g(X,Y)$ given X is

$$\mathbb{E}_{Y}[g(X,Y) \mid X] = \int_{\mathcal{Y}} g(X,Y)p(Y \mid X)dY$$

  - Finally, the **Law of total expectation**[1] states that:

$$\mathbb{E}_{X,Y}[g(X,Y)] = \mathbb{E}_{X}[\mathbb{E}_{Y}[g(X,Y) \mid X]]$$

[1] (see e.g. https://en.wikipedia.org/wiki/Law_of_total_expectation)

A reminder on the Expectation and its properties (2/3)

- The **Law of total expectation** states that:

$$\mathbb{E}_{X,Y}[g(X,Y)] = \mathbb{E}_X[\mathbb{E}_Y[g(X,Y) \mid X]]$$

proof:

$$\begin{aligned}
\mathbb{E}_{X,Y}[g(X,Y)] &= \int_{\mathcal{X}\times\mathcal{Y}} g(X,Y)p(X,Y)dXdY \\
&= \int_{\mathcal{X}\times\mathcal{Y}} g(X,Y)p(Y \mid X)p(X)dXdY \\
&= \int_{\mathcal{X}} \left(\int_{\mathcal{Y}} g(X,Y)p(Y \mid X)dY \right) p(X)dX \\
&= \mathbb{E}_X[\mathbb{E}_Y[g(X,Y) \mid X]]
\end{aligned}$$

A reminder on the Expectation and its properties (3/3)

- If $\mathcal{X}$ is continuous and $\mathcal{Y}$ discrete, and if $p(X, Y)$ is there joint density function, then
  - The **expectation** of $g(X, Y)$ is:

$$\mathbb{E}_{X,Y}\left[g(X, Y)\right] = \sum_{Y \in \mathcal{Y}} \int_{\mathcal{X}} g(X, Y) p(X, Y) dX$$

  - The **conditional expectation** over Y of $g(X, Y)$ given X is:

$$\mathbb{E}_{Y}\left[g(X, Y) \mid X\right] = \sum_{Y \in \mathcal{Y}} g(X, Y) p(Y \mid X) dY$$

  - Finally, the **Law of total expectation** still is (same proof):

$$\mathbb{E}_{X,Y}\left[g(X, Y)\right] = \mathbb{E}_{X}\left[\mathbb{E}_{Y}\left[g(X, Y) \mid X\right]\right]$$

Bayes Predictor for binary Classification

- Spaces: $\hat{\mathcal{Y}} = \mathcal{Y} = \{0,1\}$
- 0-1 loss:

$$\ell(\hat{y}, y) = 1(\hat{y} \neq y) := \begin{cases} 1 & \text{if } \hat{y} \neq y \\ 0 & \text{otherwise.} \end{cases}$$

- Risk:

$$\begin{aligned} R(f) &= \mathbb{E}\left[1(f(X) \neq Y)\right] \quad = 0 \cdot \mathbb{P}(f(X) = Y) + 1 \cdot \mathbb{P}(f(X) \neq Y) \\ &= \mathbb{P}(f(X) \neq Y) \end{aligned}$$

which is just the misclassification error rate.

- Bayes prediction function is just the assignment to the most likely class:

$$f^*(x) \in \underset{c \in \{0,1\}}{\arg\max}\, P(Y = c \mid X = x)$$

Bayes Predictor for binary Classification (proof)

- Proof: define $\eta(X) = P(Y = 1 \mid X)$

$$\begin{aligned} R(f) &= \mathbb{E}_{X,Y}\left[1(f(X) \neq Y)\right] \\ &= \mathbb{E}_X\left[\mathbb{E}_Y\left[1(f(X) \neq Y) \mid X\right]\right] \\ &= \mathbb{E}_X\left[P\left(f(X) \neq Y \mid X\right)\right] \\ &= \mathbb{E}_X\left[P\left(Y = 0 \mid X\right).1(f(X) = 1) + P\left(Y = 1 \mid X\right).1(f(X) = 0)\right] \\ &= \mathbb{E}_X\left[(1 - \eta(X)).1(f(X) = 1) + \eta(X).1(f(X) = 0)\right] \end{aligned}$$

- Thus, the bayes predictor f^* which minimizes $R(\cdot)$ will be

$$f^*(x) = \begin{cases} 1 & \text{if } 1 - \eta(x) \leqslant \eta(x) \\ 0 & \text{otherwise.} \end{cases} = \arg\max_{c \in \{0,1\}} P(Y = c \mid X = x)$$

- And $R(f^*) = \mathbb{E}_X\left[\min\left(\eta(X), 1 - \eta(X)\right)\right]$

Bayes Predictor for Multiclass Classification

- Spaces: $\hat{\mathcal{Y}} = \mathcal{Y} = \{1, \ldots, k\}$
- 0-1 loss:

$$\ell(\hat{y}, y) = 1(\hat{y} \neq y) := \begin{cases} 1 & \text{if } \hat{y} \neq y \\ 0 & \text{otherwise.} \end{cases}$$

- Risk:

$$\begin{aligned} R(f) &= \mathbb{E}[1(f(X) \neq Y)] \quad = 0 \cdot \mathbb{P}(f(X) = Y) + 1 \cdot \mathbb{P}(f(X) \neq Y) \\ &= \mathbb{P}(f(X) \neq Y), \end{aligned}$$

  which is just the misclassification error rate.

- Bayes prediction function is just the assignment to the most likely class:

$$f^*(x) \in \underset{1 \leqslant c \leqslant k}{\arg\max}\, P(Y = c \mid X = x)$$

Bayes Predictor for Multiclass Classification (proof)

Proof of the Bayes Risk in the Multiclass case:

$$
\begin{aligned}
R(f) &= \mathbb{E}_{X,Y}\left[1(f(X) \neq Y)\right] \\
&= \mathbb{E}_{X}\left[\mathbb{E}_{Y}\left[1(f(X) \neq Y) \mid X\right]\right] \\
&= \mathbb{E}_{X}\left[P\left(f(X) \neq Y \mid X\right)\right] \\
&= \mathbb{E}_{X}\left[1 - P\left(f(X) = Y \mid X\right)\right] \\
&= 1 - \mathbb{E}_{X}\left[\sum_{c=1}^{K} P\left(Y = c \mid X\right).1(f(X) = c)\right]
\end{aligned}
$$

Thus, the bayes predictor f^* which minimizes $R(\cdot)$ will be

$$
\begin{aligned}
f^*(x) &= \underset{c \in \{1 \ldots K\}}{\arg\max} \sum_{c'=1}^{K} P\left(Y = c' \mid X\right).1(c = c) \\
&= \underset{c \in \{1 \ldots K\}}{\arg\max} P(Y = c \mid X = x)
\end{aligned}
$$

Bayes Predictor for Least Squares Regression

- Spaces: $\hat{\mathcal{Y}} = \mathcal{Y} = \mathbb{R}$
- Square loss:

$$\ell(\hat{y}, y) = (\hat{y} - y)^2$$

- Risk:

$$\begin{aligned} R(f) &= \mathbb{E}\left[(f(X) - Y)^2\right] \\ (\text{exercise} \Longrightarrow) &= \mathbb{E}\left[(f(X) - \mathbb{E}[Y|X])^2\right] + \mathbb{E}\left[(Y - \mathbb{E}[Y|X])^2\right] \end{aligned}$$

- So Bayes prediction function and Bayes risk are ?

$$f^*(x) = ? \qquad R(f^*) = ?$$

Bayes Predictor for regression (proof)

Proof of the Bayes Risk in the regression case:

$$\begin{aligned}
R(f) &= \mathbb{E}_{X,Y}\left[(f(X)-Y)^2\right] \\
&= \mathbb{E}_X\left[\mathbb{E}_Y\left[(f(X)-Y)^2 \mid X\right]\right] \\
&= \mathbb{E}_X\left[\mathbb{E}_Y\left[(f(X)-\mathbb{E}[Y \mid X]+\mathbb{E}[Y \mid X]-Y)^2 \mid X\right]\right] \\
&= \mathbb{E}_X\left[\mathbb{E}_Y\left[(f(X)-\mathbb{E}[Y \mid X])^2+(\mathbb{E}[Y \mid X]-Y)^2+2\underbrace{(f(X)-\mathbb{E}[Y \mid X])(\mathbb{E}[Y \mid X]-Y)}_{=0} \mid X\right]\right.
\end{aligned}$$

The third term is null because given X, $(f(X)-\mathbb{E}[Y \mid X])$ is constant and $\mathbb{E}_Y[\mathbb{E}[Y \mid X]-Y \mid X]$ is null.

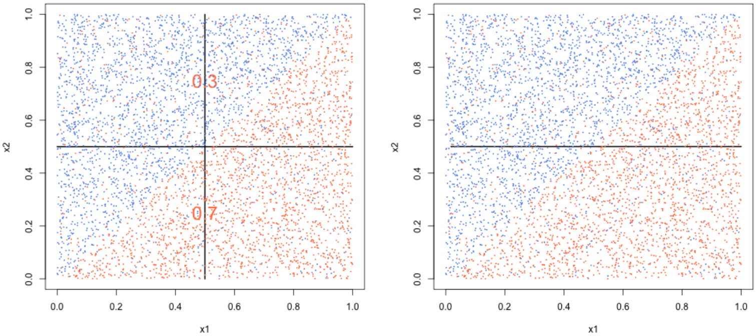
Thus, the bayes predictor f^* which minimizes $R(\cdot)$ will be $f^*(x)=\mathbb{E}[Y \mid X=x]$ and

$$R(f^*)=\mathbb{E}_X\left[\mathbb{E}_Y\left[(\mathbb{E}[Y \mid X]-Y)^2 \mid X\right]\right]=\mathbb{E}_X[var(Y \mid X)]$$

Approximation and Estimation error for Decision Trees

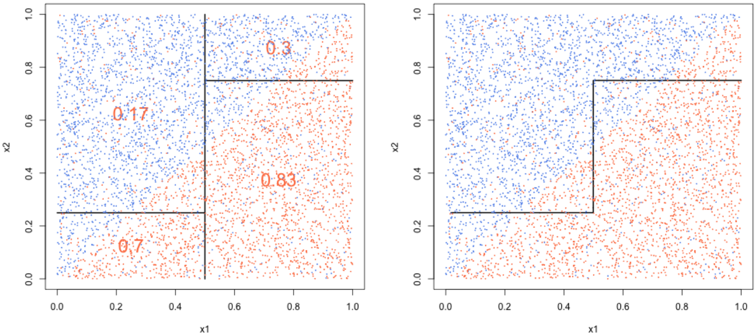
- $\mathcal{F} = \left\{ \text{all decision tree classifiers on } [0,1]^2 \right\}$
- $\mathcal{F}_d = \left\{ \text{all decision tree classifiers on } [0,1]^2 \text{ with DEPTH} \leqslant d \right\}$
- We'll consider
$$\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \mathcal{F}_4 \cdots \subset \mathcal{F}_{15}$$
- Bayes risk $= 0.1$
- In the next slides, we will have a look at the **Approximation error**, and then at the **estimation error** on a sample.

Approximation error for $\mathcal{F}_1$



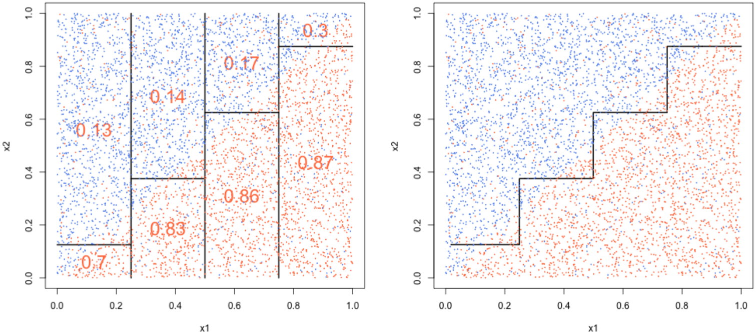
- Risk Minimizer in $\mathcal{F}_1$ has Risk $= \mathbb{P}(\text{error}) = 0.3$.
- Approximation Error $= 0.3 - 0.1 = 0.2$.

Approximation error for $\mathcal{F}_2$



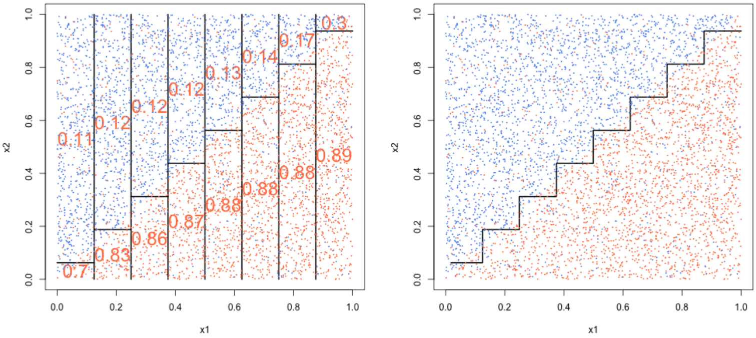
- Risk Minimizer in $\mathcal{F}_2$ has Risk $= \mathbb{P}(\text{error}) = 0.2$.
- Approximation Error = 0.2 - 0.1 = 0.1

Approximation error for $\mathcal{F}_3$



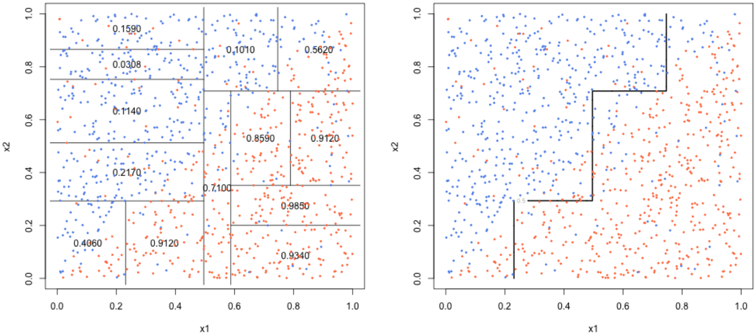
- Risk Minimizer in $\mathcal{F}_3$ has Risk $= \mathbb{P}(\text{error}) = 0.15$.
- Approximation Error = 0.15 - 0.1 = 0.05

Approximation error for $\mathcal{F}_4$



- Risk Minimizer in $\mathcal{F}_4$ has Risk $= \mathbb{P}(\text{error}) = 0.125$.
- Approximation Error = 0.125 - 0.1 = 0.025

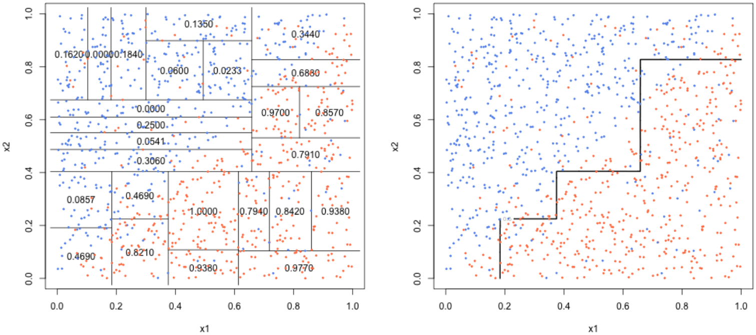
Estimation error for $\mathcal{F}_3$ Estimated From Sample ($n = 1024$)



$$R(\hat{f}) = \mathbb{P}(\text{error}) = 0.176 \pm .004$$

$$\text{Estimation Error} = \underbrace{0.176 \pm .004}_{R(\hat{f})} - \underbrace{0.150}_{\min_{f \in \mathcal{F}_3} R(f)} = .026 \pm .004$$

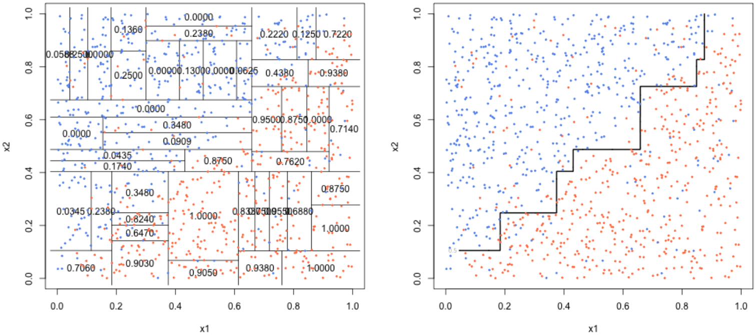
Estimation error for $\mathcal{F}_4$ Estimated From Sample ($n = 1024$)



$$R(\hat{f}) = \mathbb{P}(\text{error}) = 0.144 \pm .005$$

$$\text{Estimation Error} = \underbrace{0.144 \pm .005}_{R(\hat{f})} - \underbrace{0.125}_{\min_{f \in \mathcal{F}_4} R(f)} = .019 \pm .005$$

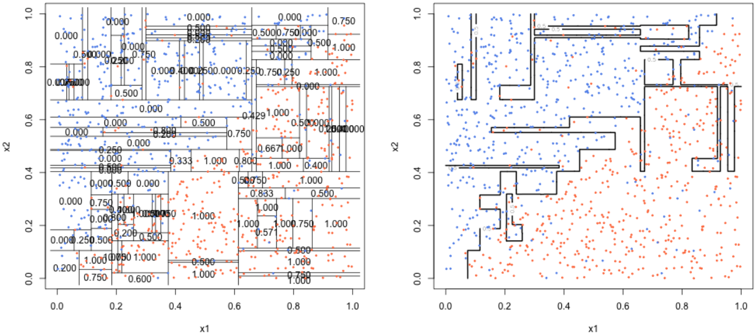
Estimation error for $\mathcal{F}_6$ Estimated From Sample ($n = 1024$)



$$R(\hat{f}) = \mathbb{P}(\text{error}) = 0.148 \pm .007$$

$$\text{Estimation Error} = \underbrace{0.148 \pm .007}_{R(\hat{f})} - \underbrace{0.106}_{\min_{f \in \mathcal{F}_6} R(f)} = .042 \pm .007$$

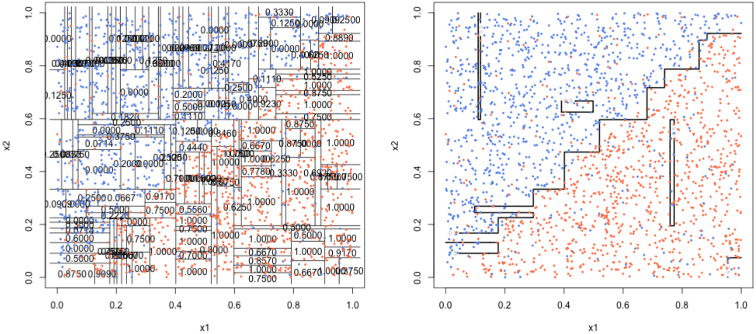
Estimation error for $\mathcal{F}_8$ Estimated From Sample ($n = 1024$)



$$R(\hat{f}) = \mathbb{P}(\text{error}) = 0.162 \pm .009$$

$$\text{Estimation Error} = \underbrace{0.162 \pm .009}_{R(\hat{f})} - \underbrace{0.102}_{\min_{f \in \mathcal{F}_8} R(f)} = .061 \pm .009$$

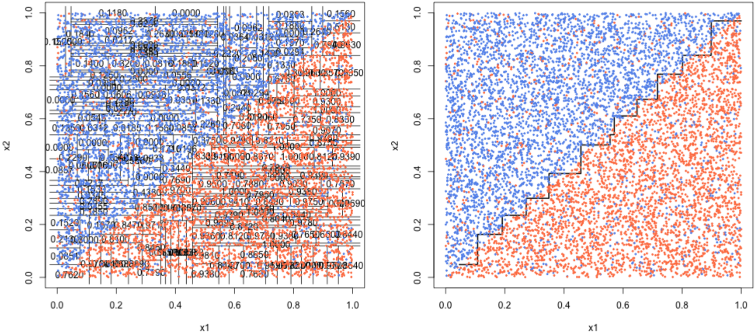
Estimation error for $\mathcal{F}_8$ Estimated From Sample ($n = 2048$)



$$R(\hat{f}) = \mathbb{P}(\text{error}) = 0.146 \pm .006$$

$$\text{Estimation Error} = \underbrace{0.146 \pm .006}_{R(\hat{f})} - \underbrace{0.102}_{\min_{f \in \mathcal{F}_3} R(f)} = .045 \pm .006$$

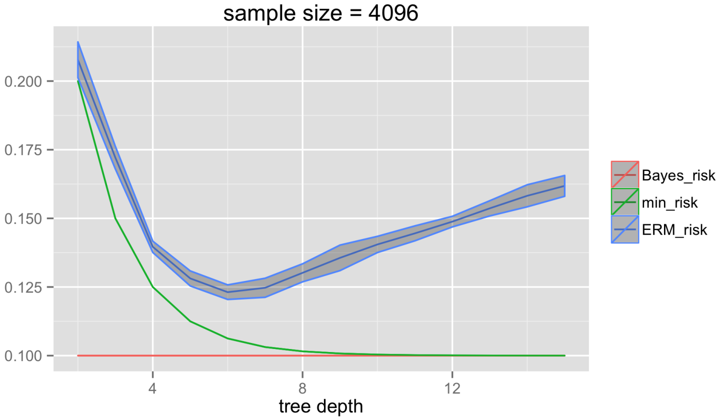
Estimation error for $\mathcal{F}_8$ Estimated From Sample ($n = 8192$)



$$R(\hat{f}) = \mathbb{P}(\text{error}) = 0.121 \pm .002$$

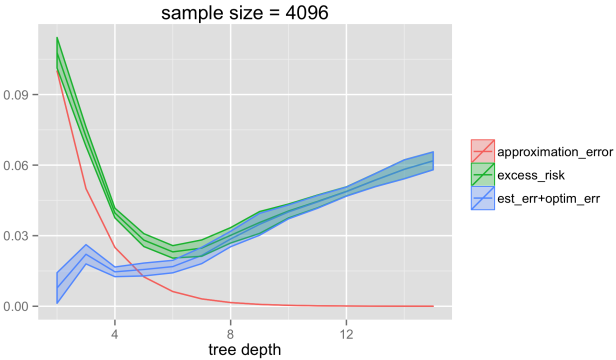
$$\text{Estimation Error} = \underbrace{0.121 \pm .002}_{R(\hat{f})} - \underbrace{0.102}_{\min_{f \in \mathcal{F}_3} R(f)} = .019 \pm .002$$

Error decomposition Summary for trees



Why do some curves have confidence bands and others not?

Error decomposition Summary for trees



Why do some curves have confidence bands and others not?

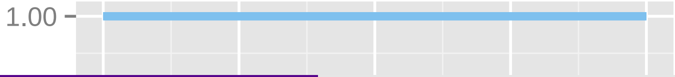
Error decomposition - Can we bound the risk ?

- With the notion of risk, with the bayes predictor, we can now make our questions on overfitting more precise for ERM. Assume a ERM learner produces f_S for a dataset S of size N drawn from P.
  - **Q1: Can the estimation error vanish as N gets large ?**
    - Can we have $R(\hat{f}_S) \xrightarrow{p} R(f_{\mathcal{F}})$ when $N \to \infty$? If so, the algorithm is said to be $\mathcal{F}$-**consistent.**
    - **How many training samples** do we need so that $R(\hat{f}_S)$ gets arbitrarily close to $R(f_{\mathcal{F}})$?
    - Is this possible for infinite classes $\mathcal{F}$?
  - **Q2: Can the true risk converge to bayes risk as N gets large ?**
    - Can we have $R(\hat{f}_S) \xrightarrow{p} R(f^*)$ when $N \to \infty$? If so, a learner is said to be **Bayes consistent.**
    - To achieve this, $\mathcal{F}$ should include all possible functions f^*. Is it possible ?
- Let us first answer Q2

Let us now show an example where a ERM is not Bayes consistent.

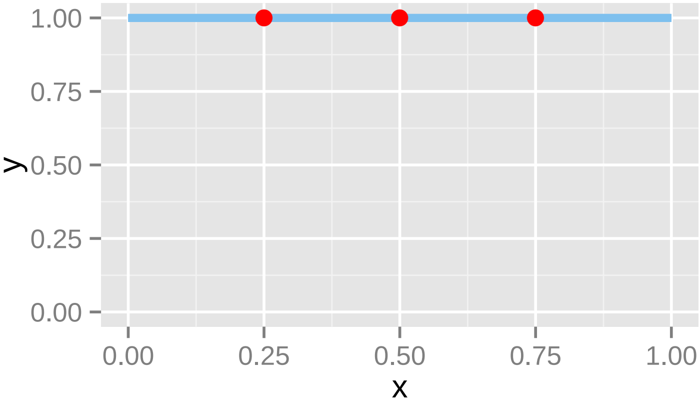
$P_{\mathcal{X}} = \text{Uniform}[0,1]$, $Y \equiv 1$ (i.e. Y is always 1). So the best prediction is to output 1 always.

$\mathcal{F} = \mathcal{X} \mapsto \{0,1\}$



Failure of ERM to converge to Bayes risk

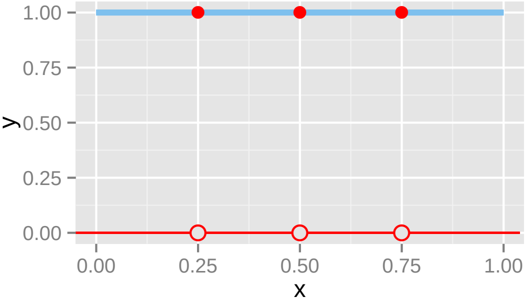
$P_{\mathcal{X}} = \text{Uniform}[0,1]$, $Y \equiv 1$ (i.e. Y is always 1).



A sample of size 3 from $P_{\mathcal{X}\times\mathcal{Y}}$.

Failure of ERM to converge to Bayes risk

$P_{\mathcal{X}} = \text{Uniform}[0,1]$, $Y \equiv 1$ (i.e. Y is always 1).

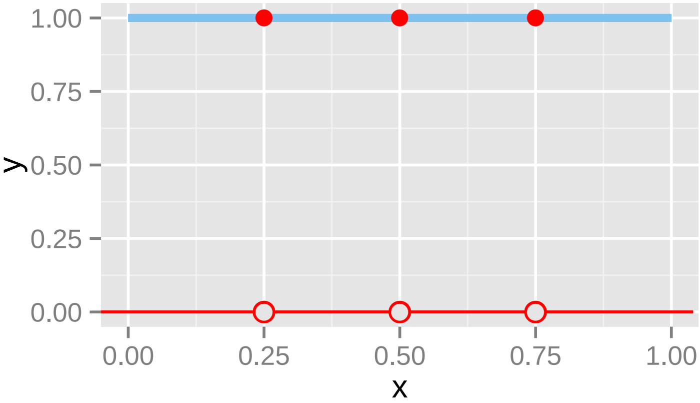


ERM principle is underdetermined (several functions might achieve equally good empirical risk). Suppose our learner picks:

$$\hat{f}(x) = 1(x \in \{0.25, 0.5, 0.75\}) = \begin{cases} 1 & \text{if } x \in \{0.25, .5, .75\} \\ 0 & \text{otherwise} \end{cases}$$

Failure of ERM to converge to Bayes risk

$P_{\mathcal{X}} = \text{Uniform}[0,1]$, $Y \equiv 1$ (i.e. Y is always 1).



Under 0/1 loss: $\hat{R}(\hat{f}) = 0$ and for *all* $f \in \mathcal{F}$, $R(f) = 1$. So estimation error does not decrease.

Error decomposition - Can we bound the risk ?

- With the notion of risk, with the bayes predictor, we can now make our questions on overfitting more precise for ERM. Assume a ERM learner produces f_S for a dataset S of size N drawn from P.
- We want classifiers with low risk
- Most importantly we want to produce good classifiers for large enough datasets.
  - **Q1- Can the estimation error vanish as N gets large ?**
    - Can we have $R(f_S) \xrightarrow{p} R(f_{\mathcal{F}})$ when $N \to \infty$? If so, the algorithm is said to be $\mathcal{F}$-**consistent.**
    - **How many training samples** do we need so that $R(f_S)$ gets arbitrarily close to $R(f_{\mathcal{F}})$?
    - Is this possible for infinite classes $\mathcal{F}$?
  - **Q2- Can the true risk converge to bayes risk as N gets large ?**
    - **No, ERM does not in general**
- Let us now answer Q1 in a simple setting

Bounding estimation error of finite hypothesis classes in the realizable case

- Let us answer Q1: can we bound the estimation error ?
- To make things simpler, we will choose a very simple case.
- *Assumption 1:* The classifier space $\mathcal{F}$ is finite, the output space $\mathcal{Y} = \{0,1\}$
- *Assumption 2:* There exists $f_{\mathcal{F}} \in \mathcal{F}$ such that $R(f_{\mathcal{F}}) = 0$ (i.e. realizable), so the true risk of h_S *is equal to the estimation error* (the approximation error and the bayes error are null).

Theorem: under these assumptions, if $\hat{f}$ is selected by ERM on a dataset of size N

then for any $\varepsilon, \delta \in]0,1[$, we have these two properties:

- if $N > \frac{1}{\varepsilon} \ln \frac{|\mathcal{F}|}{\delta}$, then with proba $1-\delta$, $R^{01}(f) < \varepsilon$
- with proba $1-\delta$, $R^{01}(\hat{f}) \leq \frac{1}{N} \ln \frac{|\mathcal{F}|}{\delta}$

Corollary: under the same assumptions, $\mathbb{E}[R^{01}(\hat{f})] \leq \frac{\ln |\mathcal{F}|}{N}$

The ERM risk of finite hypothesis classes in the realizable case

Proof: Let $Bad_\varepsilon = \{ f \in \mathcal{F} : R^{0/1}(f) > \varepsilon \}$ be the set of "bad" classifiers achieving a risk of at least ε.

Note that $\hat{R}^{0/1}(h) = 0$ iff for all $i \in 1..N$, $h(x_i) = y_i$

if $h \in Bad_\varepsilon$, then $P(\hat{R}(h)=0) = \underbrace{P(h(x_1)=y_1)}_{\le 1-\varepsilon} \times \dots \times \underbrace{P(h(x_N)=y_N)}_{\le 1-\varepsilon} \le (1-\varepsilon)^N = e^{N\log(1-\varepsilon)} \le e^{-\varepsilon N}$

We know that there is a classifier $f^* \in \mathcal{F} \setminus Bad_\varepsilon$ wich will have $\hat{R}(f^*) = 0$

If $\forall h \in Bad_\varepsilon$, $\hat{R}(h) \neq 0$, then the ERM will select $\hat{f} \notin Bad_\varepsilon$, which implies that $R(\hat{f}) < \varepsilon$

$$P(R(\hat{f}) < \varepsilon) \ge P(\forall h \in Bad_\varepsilon, \hat{R}(h) \neq 0) = 1 - P(\exists h \in Bad_\varepsilon, \hat{R}(h) = 0)$$

let us use the union bound, which states that $P(\exists i, A_i) \le \sum_i P(A_i)$

$\ge 1 - |\mathcal{F}| \times P(\hat{R}(h) = 0)$ for an arbitrary $h \in Bad_\varepsilon$

$\ge 1 - |\mathcal{F}| \times e^{-\varepsilon N}$

$\ge 1 - \delta$ with $\delta \ge |\mathcal{F}| \times e^{-\varepsilon N}$

Noting that $\delta \ge |\mathcal{F}| e^{-\varepsilon N} \iff N \ge \frac{1}{\varepsilon} \log \frac{|\mathcal{F}|}{\delta} \iff \varepsilon \ge \frac{1}{N} \log \frac{|\mathcal{F}|}{\delta}$ concludes the proof.

Outline

1. Supervised Learning: a Formal Setting
2. Empirical Risk, True risks, and the Empirical Risk Minimization Principle
    - Empirical Risk and the ERM
    - Overfitting
    - The True Risk
3. Decomposing the Risk of the ERM
    - Error Decomposition
    - The Bayes Error
    - Approximation and Estimation error
    - Warmup: Bounding the estimation error in a simple case
4. No Learner is universal: No Free Lunch thm
5. The Risk of k-Nearest Neighbors

No Free Lunch: converging to the **Bayes risk** can be arbitrarily slow (No Free Lunches)

- There are other learning algorithms than ERM (e.g. k-nearest neighbors, kernel methods, etc...)
- We know the risk of ERM can be big. Is there hope elsewhere ?

No Free Lunch Theorem

Let $\mathcal{X}$ be an arbitrary discrete domain, and let $\mathcal{Y}=\{0,1\}$. Consider **any** learning algorithm generating $h_S(\cdot)$ from a sample S. Then **for any** $N \leqslant \frac{|\mathcal{X}|}{2}$, there exists a distribution P such that the Bayes risk is null and

$$\mathbb{E}_{S\sim P^N}\left[R(h_S)\right] - R(f^*) \geqslant \frac{1}{4}$$

- This applies to *all* learners (ERM, k-NN, etc...) !

No Free Lunch Theorem (proof 1/2)

- Let $\mathcal{C} \subset \mathcal{X}$ such that $|\mathcal{C}| = 2N$.
- Let $X, X_1 \dots X_N$ be all independently and uniformly distributed over $\mathcal{C}$.
- Let $\mathcal{F}$ be the set of all 2^{2N} functions from $\mathcal{C}$ to $\mathcal{Y}$.

If there exists a target function $f \in \mathcal{F}$ such that $Y = f(X)$ with probability 1, then the risk of any function h can be written

$$R(h) = P\left(h(X) \neq f(X)\right)$$

Thus, it is sufficient to show that there exists a function $f \in \mathcal{F}$ such that in expectation over the sample $s = \{(X_i, f(X_i))\}_{i=1}^N$, we have:

$$\mathbb{E}_s P\left(h_s(X) \neq f(X)\right) = \mathbb{E}_{X_1 \dots X_N} P\left(h_s(X) \neq f(X)\right) \geqslant \frac{1}{4}$$

.

No Free Lunch Theorem (proof 2/2)

- Let F be a random function uniformly distributed over $\mathcal{F}$ and $S = \{(X_i, F(X_i))\}_{i=1}^{N}$.

$$
\begin{aligned}
\sup_{f\in\mathcal{F}} \mathbb{E}_{X_1\dots X_N} P\left(h_S(X) \neq f(X)\right) &\geqslant \mathbb{E}_F \mathbb{E}_{X_1\dots X_N} P\left(h_S(X) \neq F(X)\right) \\
&= \mathbb{E}_F \mathbb{E}_{X_1\dots X_N} \frac{1}{2N} \sum_{x\in\mathcal{C}} 1(h_S(x) \neq F(x)) \\
&\geqslant \mathbb{E}_F \mathbb{E}_{X_1\dots X_N} \frac{1}{2N} \sum_{x\in\mathcal{C}\setminus\{X_1\dots X_N\}} 1(h_S(x) \neq F(x)) \\
&= \frac{1}{2N} \mathbb{E}_{X_1\dots X_N} \sum_{x\in\mathcal{C}\setminus\{X_1\dots X_N\}} \mathbb{E}_F 1(h_S(x) \neq F(x)) \\
&= \frac{1}{2N} \mathbb{E}_{X_1\dots X_N} \sum_{x\in\mathcal{C}\setminus\{X_1\dots X_N\}} \underbrace{P_F\left(h_S(x) \neq F(x)\right)}_{=1/2} = \frac{|\mathcal{C}| - N}{4N} = \frac{1}{4}
\end{aligned}
$$

Curse of Dimensionality (No Free Lunch in $\mathcal{X} = \mathbb{R}^d$)

- What if $\mathcal{X} = \mathbb{R}^d$ and we add structure to the target function and hypothesis space ? still impossible ?
- Recall that $\eta(x)$ is c-lipschitz iff for any x_1, x_2 we have $|\eta(x_1) - \eta(x_2)| \leqslant c.dist(x_1, x_2)$.

curse of dimensionality

Let $\mathcal{X} = [0,1]^d$ and $\mathcal{Y} = \{0,1\}$. Let $c > 1$. Consider any learning algorithm generating $h_S(\cdot)$ from any sample S. Then, there exists a distribution P over $\mathcal{X} \times \mathcal{Y}$ such that $\eta(x)$ is c-Lipschitz and for any $N \leqslant \frac{(c+1)^d}{2}$, we have: $\mathbb{E}_S\left[R(h_S)\right] - R(f^*) \geqslant \frac{1}{4}$

Curse of Dimensionality (Idea of proof)

Conclusion

- With a finite classifier space, under the realizable assumption the ERM works well.
- In general, without a very strong structure (cardinality ?) of the function space OR very large amounts of data, learning is not possible.
- For ERM to work, we need to control the complexity of the function space $\mathcal{F}$.
  - either $\mathcal{F}$ is small and the approximation error is big
  - either $\mathcal{F}$ is big and the estimation error might be big
- What about infinite $\mathcal{F}$? is there hope ? (spoiler: yes)
- How about the No Free Lunches ?
- In the next lectures:
  - How do we choose the function space $\mathcal{F}$?
  - if $\mathcal{F}$ is infinite and the realizability assumption does not hold, can we have $R(h_S) \rightarrow R(f_{\mathcal{F}})$?
  - If we have a hierarchy of function classes $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \dots$ (e.g. decision trees of various depth), how can we choose the best class ?

Outline

1. Supervised Learning: a Formal Setting
2. Empirical Risk, True risks, and the Empirical Risk Minimization Principle
   - Empirical Risk and the ERM
   - Overfitting
   - The True Risk
3. Decomposing the Risk of the ERM
   - Error Decomposition
   - The Bayes Error
   - Approximation and Estimation error
   - Warmup: Bounding the estimation error in a simple case
4. No Learner is universal: No Free Lunch thm
5. The Risk of k-Nearest Neighbors

Analyzing the expected risk of the 1-nearest Neighbor

Bayes Optimal Classifier