Probably Approximately Correct Learning

Concepts,
Instance Space, Hypothesis Space,
Risks,
the PAC Learning Model,
Rectangle Learning

Joachim M. Buhmann

December 17, 2020

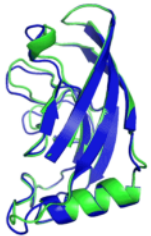
Statistical Learning Theory: the Setting

Mathematics and computer science has made remarkable breakthroughs in the last century.

- Better understanding of the world.
- Higher computing power.
- Outperforming humans in different tasks.

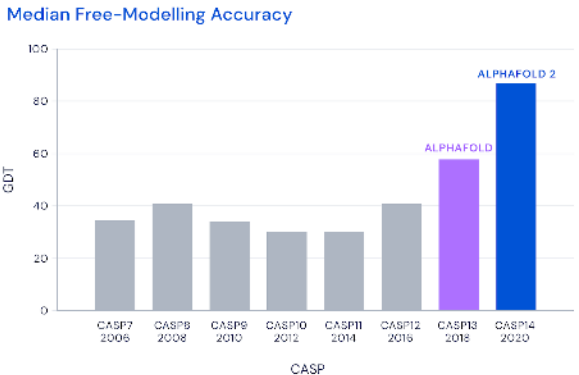
Alpha Fold 2: Folding proteins better than humans (DeepMind, 30 Nov 2020)

T1037 / 6vr4
90.7 GDT
(RNA polymerase domain)



T1049 / 6y4f
93.3 GDT
(adhesin tip)

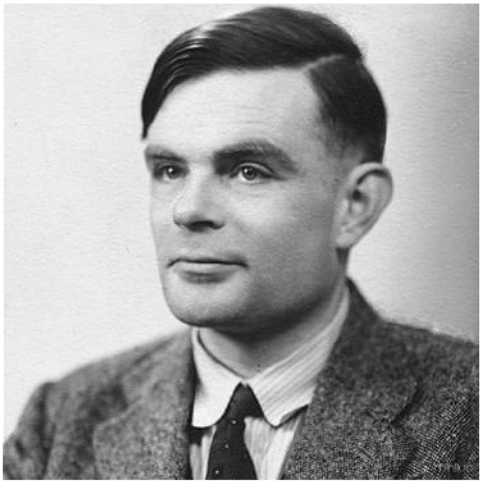
● Experimental result
● Computational prediction



https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology

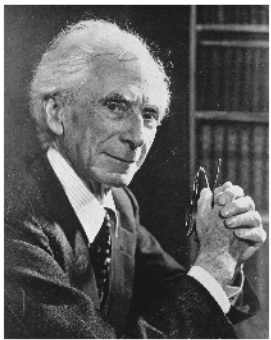
Capabilities of computers

- Can machines compute everything?
- No!
  Turing's halting problem,
  Post's correspondence problem.

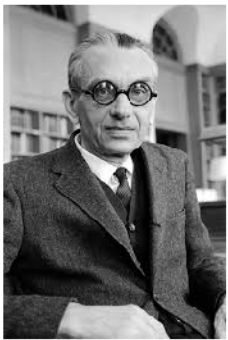


The breakthroughs of mathematics

- **Bertrand Russell** together with Alfred North Whitehead starts writing his Principia Mathematica
- Can formal logic prove everything?
- No! **Gödel**'s incompleteness theorem: there are infinitely many truths about arithmetic that cannot be proven formally.



Bertrand Russel



Kurt Gödel

The breakthroughs of machine learning

- Can machines learn everything?
- Can we learn a function to arbitrary precision with high probability?

Vladimir Vapnik

Alexey Chervonenkis

Statistical learning theory

Statistical learning theory is a framework for machine learning that aims at learning functions from data.

Probably Approximately Correct learning (PAC) is a subfield of Machine Learning that concerns with the following questions:

- What is “learnable”? Can we learn anything?
- If something is “learnable”, how much can we learn it by empirically minimizing a “cost function”?

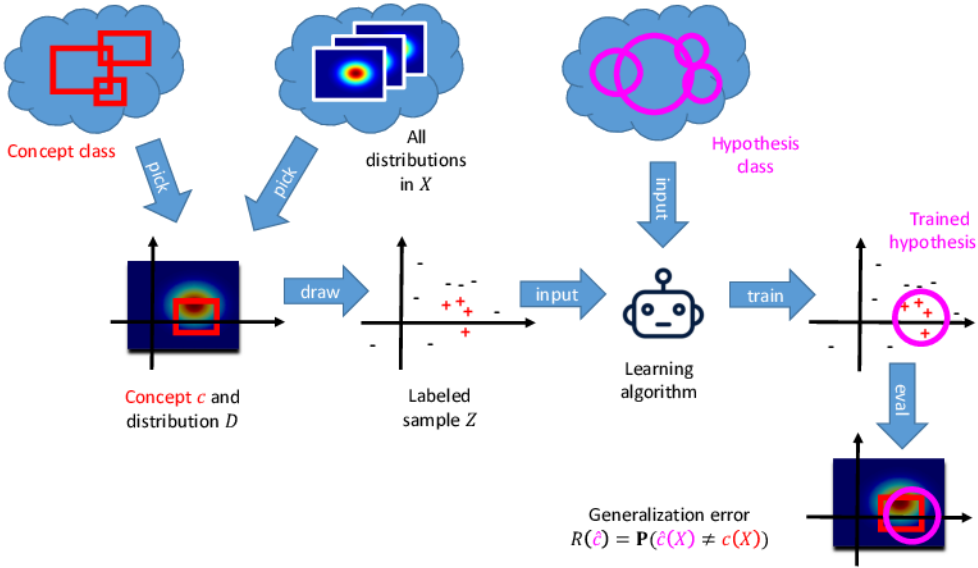
The following material is from the books

- Mohri, Rostamizadeh, and Talwalkar. Foundations of machine learning.
- Devroye, Györfi, and Lugosi. A probabilistic theory of pattern recognition.

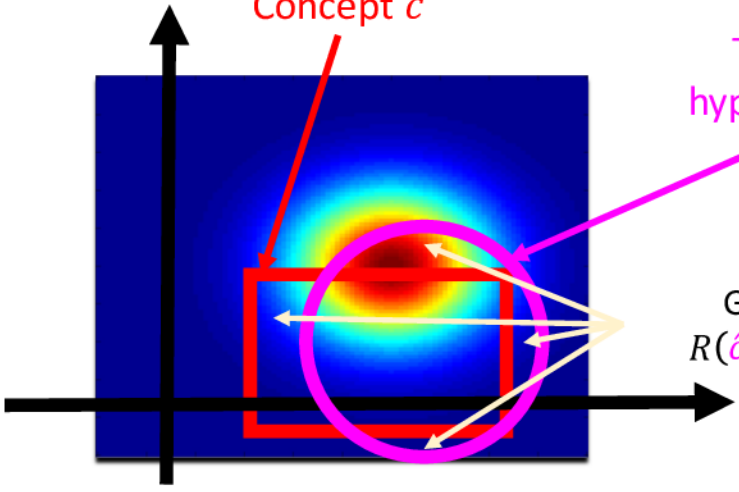
Agenda

1. Motivation for statistical learning theory
2. Basic concepts
3. What is learnable.
4. Example of “learnable” concepts.
5. Useful inequalities from statistical learning theory.

The learning problem



The learning problem



Generalization error

$$R(\hat{c}) = \mathbf{P}(\hat{c}(X) \neq c(X))$$

Generalization error and empirical error

Generalization error Not computable by the learner:

$$\mathcal{R}(\hat{c}) := \mathbf{P}\left(\hat{c}(X) \neq c(X)\right)$$

Empirical error Computable by the learner:

$$\hat{\mathcal{R}}_n(\hat{c}) := \frac{1}{n} \sum_{i \leq n} \mathbf{1}_{\hat{c}(x_i) \neq c(x_i)}$$

One can show that $\mathbb{E}_{X, X_1, \ldots, X_n}\left[\hat{\mathcal{R}}_n(\hat{c}(X))\right] = \mathcal{R}(\hat{c})$.

Notions from statistical learning theory

Instance space $\mathcal{X}$: think of $\mathcal{X}$ as being a set of instances or objects in the learner's world.

Concept: A concept is a subset c of $\mathcal{X}$ (we sometimes think of c as a function $c : \mathcal{X} \to \{0, 1\}$).

Concept class: A set of concepts we wish to learn.

Hypothesis class: Another set of concepts that we use to learn a target concept from the concept class.

No additional prior knowledge on the distribution on $\mathcal{X}$ is available.

Observe that this differs from Bayesian approaches, which require a prior on $\mathcal{X}$.

The PAC Learning Model

Definition

Let $\mathcal{H}$ and c be a hypothesis class and a concept. A learning algorithm is an algorithm that receives as input a labeled sample $\mathcal{Z} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ with $\forall i, y_i = c(x_i)$ and outputs a hypothesis $\hat{c} \in \mathcal{H}$.

The PAC Learning Model

A learning algorithm $\mathcal{A}$ can learn a concept c from $\mathcal{C}$ if, given as input a sufficiently large sample, it outputs a hypothesis that generalizes well with high probability.

Definition

A learning algorithm $\mathcal{A}$ can learn a concept c if there is a polynomial function $poly(\cdot, \cdot, \cdot)$ such that

1. for any distribution $\mathcal{D}$ on $\mathcal{X}$ and
2. for any $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$,

if $\mathcal{A}$ receives as input a sample $\mathcal{Z}$ of size $n \geq poly(1/\epsilon, 1/\delta, size(c))$, then $\mathcal{A}$ outputs $\hat{c}$ such that

$$\mathbf{P}_{\mathcal{Z} \sim \mathcal{D}^n} \left(\mathcal{R}\left(\hat{c}\right) \leq \epsilon \right) \geq 1 - \delta.$$

This probability is taken over $\mathcal{Z}$ and any internal randomization of $\mathcal{A}$. The value $size(c)$ indicates the size of the representation of concept c.

The PAC Learning Model

Definition

A concept class $\mathcal{C}$ is PAC learnable from a hypothesis class $\mathcal{H}$ if there is an algorithm that can learn any concept in $\mathcal{C}$.

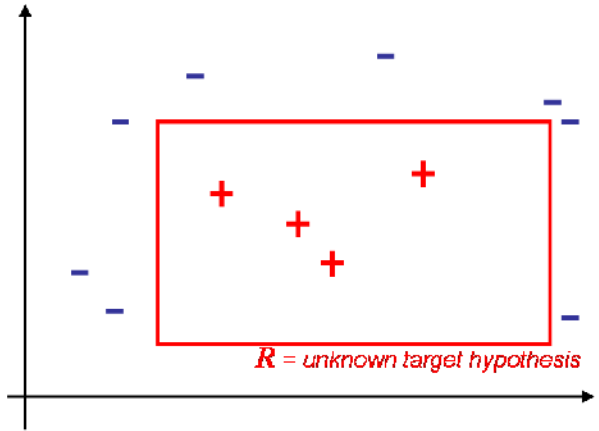
Efficient PAC learning: If $\mathcal{A}$ runs in time polynomial in $1/\epsilon$ and $1/\delta$, we say that $\mathcal{C}$ is efficiently PAC learnable.

- The input ϵ is called the **error** parameter, the parameter δ denotes the **confidence** value.

Remark: No assumptions on the distribution of instances are made!

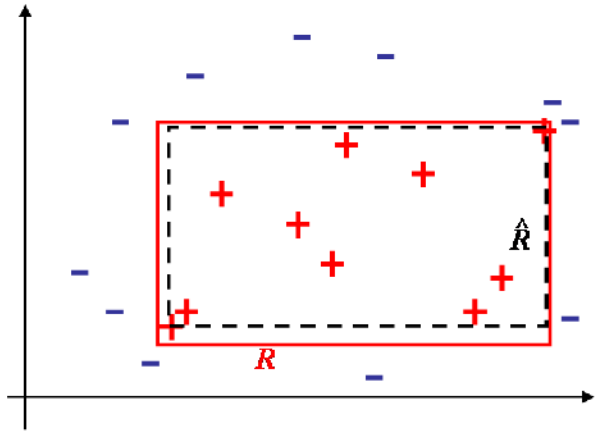
Example: Axis-aligned rectangles are PAC learnable

Let $\mathcal{C}$ be the concept of all axis-aligned rectangles. We show that $\mathcal{C}$ can be learned from $\mathcal{H} = \mathcal{C}$.



Example: Axis-aligned rectangles are PAC learnable

Consider the algorithm $\mathcal{A}$ that outputs the smallest rectangle $\hat{R}$ containing all positively labeled points. We show that $\mathcal{A}$ can learn any concept $R \in \mathcal{C}$.



How do we prove that $\mathcal{A}$ learns rectangles?

We show that there exists $poly(\cdot, \cdot, \cdot)$ such that

- for any rectangle $R \in \mathcal{C}$,
- for any distribution $\mathcal{D}$ on $\mathbb{R}^2$,
- for any $\epsilon > 0$ and $\delta > 0$,

if $\mathcal{A}$ receives a sample $\mathcal{Z}$ of size $n \geq poly(1/\epsilon, 1/\delta, size(R) = 4)$, then

$$\mathbf{P}\left(\mathcal{R}(\hat{R}) \leq \epsilon\right) \geq 1 - \delta.$$

For a rectangle R, $size(R) = 4$ since we need four coordinates to define it.

How do we prove that $\mathcal{A}$ learns rectangles?

We define next an event called $\hat{\mathcal{R}}IG$ (which stands for "$\hat{\mathcal{R}}$ is good enough") such that

$$\mathbf{P}\left(\mathcal{R}(\hat{R}) \leq \epsilon\right) \geq \mathbf{P}\left(\hat{\mathcal{R}}IG\right) \geq 1 - 4\exp\left(-\frac{n\epsilon}{4}\right).$$

Observe that we just need to ensure that $1 - 4\exp\left(-\frac{n\epsilon}{4}\right) \geq 1 - \delta$ or equivalently that

$$n \geq \frac{4}{\epsilon}\ln\frac{4}{\delta}.$$

We can ensure this by letting

$$n \geq \underbrace{\frac{4}{\epsilon} \times \frac{4}{\delta}}_{poly(1/\epsilon,1/\delta,4)} \geq \frac{4}{\epsilon}\ln\frac{4}{\delta}.$$

Tasks to do

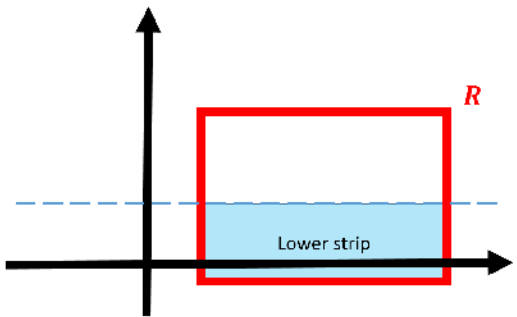
1. Define $\hat{\mathcal{R}}IG$
2. Prove that

$$\mathbf{P}\left(\mathcal{R}(\hat{R}) \leq \epsilon\right) \geq \mathbf{P}\left(\hat{\mathcal{R}}IG\right) \geq 1 - 4\exp\left(-\frac{n\epsilon}{4}\right).$$

The event $\hat{\mathcal{R}}IG$

Consider the two rectangles that result from drawing a line through R that is parallel to the x-axis. We call the upper rectangle an upper strip.

In an analogous way, we define lower, left, and right strips.

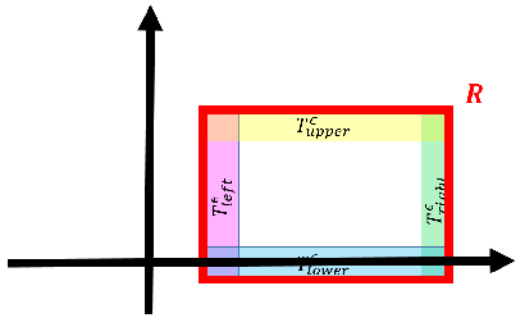


The event $\hat{\mathcal{R}}IG$

Let T^{ϵ}_{upper} the upper strip such that $\mathbf{P}\left(T^{\epsilon}_{upper}\right) = \epsilon/4$.

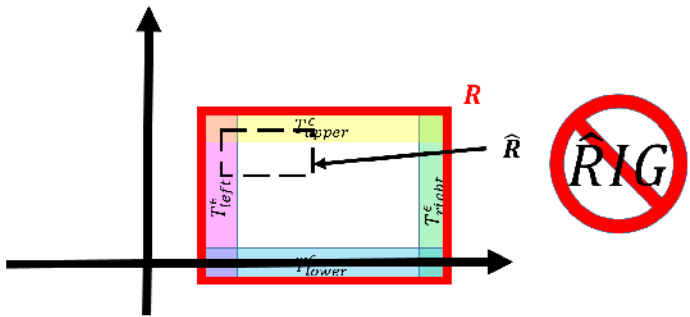
In an analogous way, we define T^{ϵ}_{lower}, T^{ϵ}_{left}, T^{ϵ}_{right}. We let

$$T^{\epsilon} := \bigcup_i T^{\epsilon}_i = T^{\epsilon}_{upper} \cup \ldots \cup T^{\epsilon}_{right}.$$



The event $\hat{\mathcal{R}}IG$

The event $\hat{\mathcal{R}}IG$ is the event that $\hat{R}$ intersects all four strips.



Tasks to do

1. Define $\hat{\mathcal{R}}IG$
2. Prove that

$$\mathbf{P}\left(\mathcal{R}(\hat{R}) \leq \epsilon\right) \geq \mathbf{P}\left(\hat{\mathcal{R}}IG\right) \geq 1 - 4\exp\left(-\frac{n\epsilon}{4}\right).$$

Proving the bound

See black board.

The universal concept class is not PAC-learnable

Let $\mathcal{X} = \{0, 1\}^*$ be the set of all finite binary sequences. The concept class $\mathcal{C}$ formed by all subsets of $\mathcal{X}$ is not PAC-learnable from $\mathcal{C}$.

The proof is hard though.

Consistent hypothesis and finite hypothesis classes

Let $\mathcal{C}$ be a finite concept class and assume that $\mathcal{H} = \mathcal{C}$. Let $\mathcal{A}$ be an algorithm that returns a consistent hypothesis $\hat{c}$ (i.e., $\forall n < \infty : \hat{\mathcal{R}}_n(\hat{c}) = 0$) for any target concept $c \in \mathcal{C}$ and any i.i.d. sample $\mathcal{Z}$. For any $\epsilon, \delta > 0$, if

$$n \geq \frac{1}{\epsilon}\left(\log|\mathcal{H}| + \log\frac{1}{\delta}\right),$$

then the success probability is determined by

$$\mathbf{P}\left(\mathcal{R}(\hat{c}) \leq \epsilon\right) \geq 1 - \delta;$$

or the error probability is bounded by

$$\mathbf{P}\left(\mathcal{R}(\hat{c}) > \epsilon\right) \leq \delta.$$

Proof

See black board.

The general stochastic setting

In general, an instance's label is not determined by the underlying concept. This is modeled with a distribution $\mathcal{D}$ on $\mathcal{X} \times \{0, 1\}$. It reflects the fact that two instances with identical features may have different labels, like when two patients with very similar features show different reactions to the same drug.

The training dataset is therefore a sample $\mathcal{Z} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ from $\mathcal{D}$.

The goal then is to find a hypothesis $\hat{c} \in \mathcal{H}$ with small generalization error

$$\mathcal{R}(\hat{c}) = \mathbf{P}_{x,y\sim\mathcal{D}}(\hat{c}(x) \neq y) = \mathbb{E}_{x,y\sim\mathcal{D}}\left(\mathbf{1}_{\hat{c}(x)\neq y}\right).$$

If the Bayes optimal classifier is not an element of the hypothesis class $\mathcal{C}$, then it is impossible to attain $\forall\, 0 < \epsilon \leq \frac{1}{2} : \mathcal{R}(\hat{c}) \leq \epsilon$. Instead, we aim to attain the best solution given the hypothesis class, i.e.,

$$\mathcal{R}(\hat{c}) - \inf_{c\in\mathcal{C}} \mathcal{R}(c) \leq \epsilon\,.$$

The general PAC learning model

A learning algorithm $\mathcal{A}$ can learn a concept class $\mathcal{C}$ from $\mathcal{H}$ if, given as input a sufficiently large sample, it outputs a hypothesis that generalizes well with high probability.

Definition

A learning algorithm $\mathcal{A}$ can learn a concept class $\mathcal{C}$ from $\mathcal{H}$ if there is a polynomial function *poly* $(\cdot, \cdot, \cdot)$ such that

1. for any distribution $\mathcal{D}$ on $\mathcal{X} \times \{0, 1\}$ and
2. for any $0 < \epsilon < 1/2$ and $0 < \delta < 1/2$,

if $\mathcal{A}$ receives as input a sample $\mathcal{Z}$ of size $n \geq$ *poly* $(1/\epsilon, 1/\delta, dim\,(\mathcal{X}))$, then $\mathcal{A}$ outputs $\hat{c} \in \mathcal{H}$ such that

$$\mathbf{P}_{\mathcal{Z} \sim \mathcal{D}^n} \left(\mathcal{R}\left(\hat{c}\right) - \inf_{c \in \mathcal{C}} \mathcal{R}\left(c\right) \leq \epsilon \right) \geq 1 - \delta.$$

Efficient PAC learning: If $\mathcal{A}$ runs in time polynomial in $1/\epsilon$ and $1/\delta$, we say that $\mathcal{A}$ is an efficient PAC learning algorithm.

Error bounds for finite and infinite hypothesis classes

Let $\epsilon > 0$. For a sample $\mathcal{Z} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, let $\hat{c}_n^*$ be the hypothesis obtained by empirical risk minimization:

$$\hat{c}_n^* = \arg\min_{c \in \mathcal{C}} \frac{1}{n} \left| \{(x_i, y_i) : c(x_i) \neq y_i, i \leq n\} \right|.$$

- If $\mathcal{C}$ is finite ($|\mathcal{C}| < \infty$),

$$\mathbf{P}\left(\mathcal{R}\left(\hat{c}_n^*\right) - \inf_{c \in \mathcal{C}} \mathcal{R}\left(c\right) > \epsilon \right) \leq 2 \left|\mathcal{C}\right| \exp\left(-2n\epsilon^2\right).$$

- Uncountably large hypothesis classes: the VC-dimension $VC_{\mathcal{C}}$ of a concept class $\mathcal{C}$ is a complexity measure for $\mathcal{C}$. If $VC_{\mathcal{C}} > 2$, then

$$\mathbf{P}\left(\mathcal{R}\left(\hat{c}_n^*\right) - \inf_{c \in \mathcal{C}} \mathcal{R}\left(c\right) > \epsilon \right) \leq 9 n^{VC_{\mathcal{C}}} \exp\left(-\frac{n\epsilon^2}{32}\right).$$

Concept classes with finite VC dimension are effective for learning

If $\mathcal{C}$ has a finite VC-dimension, then

$$\mathbf{P}\left(\mathcal{R}\left(\hat{c}_n^*\right) - \inf_{c\in\mathcal{C}} \mathcal{R}\left(c\right) > \epsilon\right) \leq 9n^{VC_{\mathcal{C}}} \exp\left(-\frac{n\epsilon^2}{32}\right) \to 0, \text{as } n \to \infty.$$

VC-dimension

Definition

The set A of instances can be shattered by a concept class $\mathcal{C}$ if for every subset $S \subseteq A$, there is a concept $c_S \in \mathcal{C}$, such that $S = c_S \cap A$.

Examples:

- Any set of two points in $\mathbb{R}^2$ can be shattered by the class of axis-aligned rectangles.
- No set of three numbers in $\mathbb{R}$ can be shattered by the class of intervals.
- There is a set of three points in $\mathbb{R}^3$ that can be shattered by the class of axis-alligned rectangles. However, some sets of three points cannot be shattered by this class.

VC-dimension

Definition

The VC-dimension $VC_{\mathcal{C}}$ of a concept class $\mathcal{C}$ is computed as follows:

1. $n \leftarrow 1$
2. Is there a set of $n+1$ instances in $\mathcal{X}$ that can be shattered by $\mathcal{C}$?
3. If yes, then $n \leftarrow n+1$ and go to step 2.
4. Otherwise, $VC_{\mathcal{C}} = n$.

Contest

Open the Kahoot! website or download the app.

Link to the contest: `https://play.kahoot.it/#/k/908011b5-e067-4798-b16c-3644472aa64b`.

Conclusion

1. What is learnable?
2. PAC-learnability.
3. Useful inequalities to bound the probability $\mathbf{P}\left(\mathcal{R}\left(\hat{c}_n^*\right) - \inf_{c \in \mathcal{C}} \mathcal{R}\left(c\right) > \epsilon\right)$.

Bayes error and Bayes classifier

The Bayes error is

$$\mathcal{R}^* := \inf_f \mathcal{R}(f),$$

where f ranges over all measurable functions from $\mathcal{X}$ to $\{0, 1\}$.

A Bayes classifier c^{Bayes} for $\mathcal{R}$ is a measurable function (not necessarily in the concept nor the hypothesis class) such that

$$\mathcal{R}\left(c^{Bayes}\right) = \mathcal{R}^*.$$

For $0/1$ loss, it holds that

$$c^{Bayes}(x) := \arg\max_{y \in \{0,1\}} \mathbf{P}(y \mid x).$$

Strong and Weak Learning

Assumption: Restricted classifier $c \in \mathcal{C}$ (hypothesis class).
Classification Error for empirically determined classifier $\hat{c}$

$$\mathbf{P}\left\{\mathcal{R}(\hat{c}) - \inf_{c \in \mathcal{C}} \mathcal{R}(c) > \epsilon\right\} < \delta \qquad (*)$$

Strong PAC Learning - Demand arbitrarily small error ϵ with high probability $1 - \delta$.
Weak PAC Learning - Demand that $(*)$ holds for 'large' (but not trivial) error ϵ

- Example: Binary classification, require that

$$\epsilon \leq \frac{1}{2} - \gamma \qquad (\gamma > 0)$$

Weak learners are necessary to build ensemble classifiers, e.g., bagging classifiers as random forests or boosting methods like Adaboost.

It is often assumed that $\mathcal{H} = \mathcal{C}$ in this setting. We make this assumption for the rest of the slides.