Fundamentals of Learning and Information Processing

Session 12: Statistical Machine Learning (II)

Gang Li

Deakin University, Australia

2021-08-07

Table of Content

PAC Learning

The learner's task is to:

Input: training data $S = \{(x_1, y_1), \cdots, (x_m, y_m)\} \in (\mathscr{X} \times \mathscr{Y})^m$

Output: prediction rule $h : \mathscr{X} \to \mathscr{Y}$

Measure The error of a prediction rule $h : \mathscr{X} \to \mathscr{Y}$ can be defined as:

Generalization risk $L_{(\mathscr{D},f)}(h) \stackrel{def}{=} P_{x\sim\mathscr{D}}[h(x) \neq f(x)] \stackrel{def}{=} \mathscr{D}(\{x : h(x) \neq f(x)\})$

Empirical risk $L_S(h) \stackrel{def}{=} \frac{|\{i\in[m]:h(x_i)\neq y_i\}|}{m}$

The Statistical Learning Framework

The learner's task is to:

Input: training data $S = \{(x_1, y_1), \cdots, (x_m, y_m)\} \in (\mathscr{X} \times \mathscr{Y})^m$

Output: prediction rule $h : \mathscr{X} \to \mathscr{Y}$

Measure The error of a prediction rule $h : \mathscr{X} \to \mathscr{Y}$ can be defined as:

Generalization risk $L_{(\mathscr{D},f)}(h) \stackrel{def}{=} P_{x\sim\mathscr{D}}[h(x) \neq f(x)] \stackrel{def}{=} \mathscr{D}(\{x : h(x) \neq f(x)\})$

Empirical risk $L_S(h) \stackrel{def}{=} \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$

ERM

ERM comes up with a predictor h that minimizes $L_S(h)$

$$ERM_{\mathscr{Y}^{\mathscr{X}}}(S) \in \underset{h \in \mathscr{Y}^{\mathscr{X}}}{\operatorname{argmin}} L_S(h)$$

ERM with Inductive Bias comes up with any $h \in \mathscr{H}$ that minimizes $L_S(h)$

$$ERM_{\mathscr{H}}(S) \in \underset{h \in \mathscr{H}}{\operatorname{argmin}} L_S(h)$$

Can only be *Approximately* correct

For any training data S with m i.i.d. examples, we should not hope find an h s.t. $L_{(\mathscr{D},f)}(h) = 0$

Proof.

- For every $\epsilon \in (0,1)$ take $\mathscr{X} = \{x_1, x_2\}$ and $\mathscr{D}(\{x_1\}) = 1-\epsilon$, $\mathscr{D}(\{x_2\}) = \epsilon$
- The probability not to see x_2 at all among m i.i.d. examples in S is $(1-\epsilon)^m \approx e^{-\epsilon m}$
- So if $\epsilon \ll \frac{1}{m}$ we are likely not to see x_2 at all, but then we can not know its label.

$\square$

Can only be *Approximately* correct

For any training data S with m i.i.d. examples, we should not hope find an h s.t. $L_{(\mathscr{D},f)}(h) = 0$

Proof.

- For every $\epsilon \in (0,1)$ take $\mathscr{X} = \{x_1, x_2\}$ and $\mathscr{D}(\{x_1\}) = 1-\epsilon$, $\mathscr{D}(\{x_2\}) = \epsilon$
- The probability not to see x_2 at all among m i.i.d. examples in S is $(1-\epsilon)^m \approx e^{-\epsilon m}$
- So if $\epsilon \ll \frac{1}{m}$ we are likely not to see x_2 at all, but then we can not know its label.

□

Relaxation.

- We would be happy with $L_{(\mathscr{D},f)}(h) < \epsilon$, where ϵ is the user-specified accuracy parameter.

□

Can only be *Probably* correct

For any training data S with m i.i.d. examples, no algorithm can guarantee $L_{(\mathscr{D},f)}(h) \leq \epsilon$

Proof.

- Recall that the input to the learner is a set of randomly generated examples, there is always a (very small) chance to see the same example again and again.

□

Can only be *Probably* correct

For any training data S with m i.i.d. examples, no algorithm can guarantee $L_{(\mathscr{D},f)}(h) \le \epsilon$

Proof.

- Recall that the input to the learner is a set of randomly generated examples, there is always a (very small) chance to see the same example again and again.

□

Relaxation.

- We would allow the algorithm to fail with probability δ, where $\delta \in (0,1)$ is the user-specified confidence parameter
- Here, the probability is over the random choice of examples

□

Probably Approximately Correct (PAC) Learnability

A hypothesis class $\mathscr{H}$ is PAC learnable if there exists a function $m_{\mathscr{H}} : (0,1)^2 \to \mathscr{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathscr{D}$ over $\mathscr{X}$, and for every labelling function $f : \mathscr{X} \to \{0,1\}$, if the realizable assumption holds with respect to $\mathscr{H}$, $\mathscr{D}$ and f, then when we run the algorithm on $m \geq m_{\mathscr{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathscr{D}$ and labelled by f, the algorithm returns a hypothesis h such that, with probability of at least $(1-\delta)$, $L_{(\mathscr{D},f)}(h) \leq \epsilon$.

A hypothesis class $\mathscr{H}$ is PAC learnable if there exists a function $m_{\mathscr{H}} : (0,1)^2 \rightarrow \mathscr{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathscr{D}$ over $\mathscr{X}$, and for every labelling function $f : \mathscr{X} \rightarrow \{0,1\}$, if the realizable assumption holds with respect to $\mathscr{H}$, $\mathscr{D}$ and f, then when we run the algorithm on $m \geq m_{\mathscr{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathscr{D}$ and labelled by f, the algorithm returns a hypothesis h such that, with probability of at least $(1-\delta)$, $L_{(\mathscr{D},f)}(h) \leq \epsilon$.

Key Points.

- It is a distribution free model, i.e. no particular assumption about $\mathscr{D}$

Probably Approximately Correct (PAC) Learnability

A hypothesis class $\mathscr{H}$ is PAC learnable if there exists a function $m_{\mathscr{H}} : (0,1)^2 \to \mathscr{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathscr{D}$ over $\mathscr{X}$, and for every labelling function $f : \mathscr{X} \to \{0,1\}$, if the realizable assumption holds with respect to $\mathscr{H}$, $\mathscr{D}$ and f, then when we run the algorithm on $m \geq m_{\mathscr{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathscr{D}$ and labelled by f, the algorithm returns a hypothesis h such that, with probability of at least $(1-\delta)$, $L_{(\mathscr{D},f)}(h) \leq \epsilon$.

Key Points.

- It is a distribution free model, i.e. no particular assumption about $\mathscr{D}$
- Training and test samples are drawn according to the same $\mathscr{D}$ (otherwise transfer learning)

A hypothesis class $\mathscr{H}$ is PAC learnable if there exists a function $m_{\mathscr{H}} : (0,1)^2 \to \mathscr{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathscr{D}$ over $\mathscr{X}$, and for every labelling function $f : \mathscr{X} \to \{0,1\}$, if the realizable assumption holds with respect to $\mathscr{H}$, $\mathscr{D}$ and f, then when we run the algorithm on $m \geq m_{\mathscr{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathscr{D}$ and labelled by f, the algorithm returns a hypothesis h such that, with probability of at least $(1-\delta)$, $L_{(\mathscr{D},f)}(h) \leq \epsilon$.

Key Points.

- It is a distribution free model, i.e. no particular assumption about $\mathscr{D}$
- Training and test samples are drawn according to the same $\mathscr{D}$ (otherwise transfer learning)
- It deals with the question of learnability for $\mathscr{H}$, not a particular concept, namely the "*target labelling function*" f.

□

Probably Approximately Correct (PAC) Learnability

A hypothesis class $\mathscr{H}$ is PAC learnable if there exists a function $m_{\mathscr{H}} : (0,1)^2 \to \mathscr{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathscr{D}$ over $\mathscr{X}$, and for every labelling function $f : \mathscr{X} \to \{0,1\}$, if the realizable assumption holds with respect to $\mathscr{H}$, $\mathscr{D}$ and f, then when we run the algorithm on $m \geq m_{\mathscr{H}}(\epsilon,\delta)$ i.i.d. examples generated by $\mathscr{D}$ and labelled by f, the algorithm returns a hypothesis h such that, with probability of at least $(1-\delta)$, $L_{(\mathscr{D},f)}(h) \leq \epsilon$.

Steps.

- The learner does not know $\mathscr{D}$ and f

A hypothesis class $\mathscr{H}$ is PAC learnable if there exists a function $m_{\mathscr{H}} : (0,1)^2 \to \mathscr{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathscr{D}$ over $\mathscr{X}$, and for every labelling function $f : \mathscr{X} \to \{0,1\}$, if the realizable assumption holds with respect to $\mathscr{H}$, $\mathscr{D}$ and f, then when we run the algorithm on $m \geq m_{\mathscr{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathscr{D}$ and labelled by f, the algorithm returns a hypothesis h such that, with probability of at least $(1-\delta)$, $L_{(\mathscr{D},f)}(h) \leq \epsilon$.

Steps.

- The learner does not know $\mathscr{D}$ and f
- The learner receives the *accuracy* parameter ϵ and the *confidence* parameter δ

Probably Approximately Correct (PAC) Learnability

A hypothesis class $\mathcal{H}$ is PAC learnable if there exists a function $m_{\mathcal{H}} : (0,1)^2 \to \mathcal{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathcal{D}$ over $\mathcal{X}$, and for every labelling function $f : \mathcal{X} \to \{0,1\}$, if the realizable assumption holds with respect to $\mathcal{H}$, $\mathcal{D}$ and f, then when we run the algorithm on $m \geq m_{\mathcal{H}}(\epsilon,\delta)$ i.i.d. examples generated by $\mathcal{D}$ and labelled by f, the algorithm returns a hypothesis h such that, with probability of at least $(1-\delta)$, $L_{(\mathcal{D},f)}(h) \leq \epsilon$.

Steps.

- The learner does not know $\mathcal{D}$ and f
- The learner receives the *accuracy* parameter ϵ and the *confidence* parameter δ
- The learner can ask for training data S containing $m_{\mathcal{H}}(\epsilon,\delta)$ examples
  - the number of examples can depend on ϵ and δ, but not on depend $\mathcal{D}$ and f

Probably Approximately Correct (PAC) Learnability

A hypothesis class $\mathcal{H}$ is PAC learnable if there exists a function $m_{\mathcal{H}} : (0,1)^2 \to \mathcal{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathcal{D}$ over $\mathcal{X}$, and for every labelling function $f : \mathcal{X} \to \{0,1\}$, if the realizable assumption holds with respect to $\mathcal{H}$, $\mathcal{D}$ and f, then when we run the algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$ and labelled by f, the algorithm returns a hypothesis h such that, with probability of at least $(1-\delta)$, $L_{(\mathcal{D},f)}(h) \leq \epsilon$.

Steps.

- The learner does not know $\mathcal{D}$ and f
- The learner receives the *accuracy* parameter ϵ and the *confidence* parameter δ
- The learner can ask for training data S containing $m_{\mathcal{H}}(\epsilon, \delta)$ examples
  - the number of examples can depend on ϵ and δ, but not on depend $\mathcal{D}$ and f
- The learner should output a hypothesis h, s.t. with probability of at least $(1-\delta)$ it holds that $L_{(\mathcal{D},f)}(h) \leq \epsilon$.
  - the learner should be **P**robably (with probability at least $(1-\delta)$) **A**pproximately (up to accuracy ϵ) **C**orrect

Sample Complexity

The function $m_{\mathscr{H}} : (0,1)^2 \to \mathscr{N}$ determines the sample complexity of learning $\mathscr{H}$, namely, $m_{\mathscr{H}}(\epsilon, \delta)$ represents how many examples are required to guarantee a PAC solution:

- It is a function of the *accuracy* parameter ϵ and the *confidence* parameter δ
- It also depends on the properties of the hypothesis class $\mathscr{H}$.
  - If $\mathscr{H}$ is PAC learnable, there are many functions $m_{\mathscr{H}}$ that satisfy the requirements given in the PAC learnability definition.
  - We define the sample complexity to be the "*minimal function*"

Sample Complexity

The function $m_{\mathcal{H}} : (0,1)^2 \rightarrow \mathcal{N}$ determines the sample complexity of learning $\mathcal{H}$, namely, $m_{\mathcal{H}}(\epsilon, \delta)$ represents how many examples are required to guarantee a PAC solution:

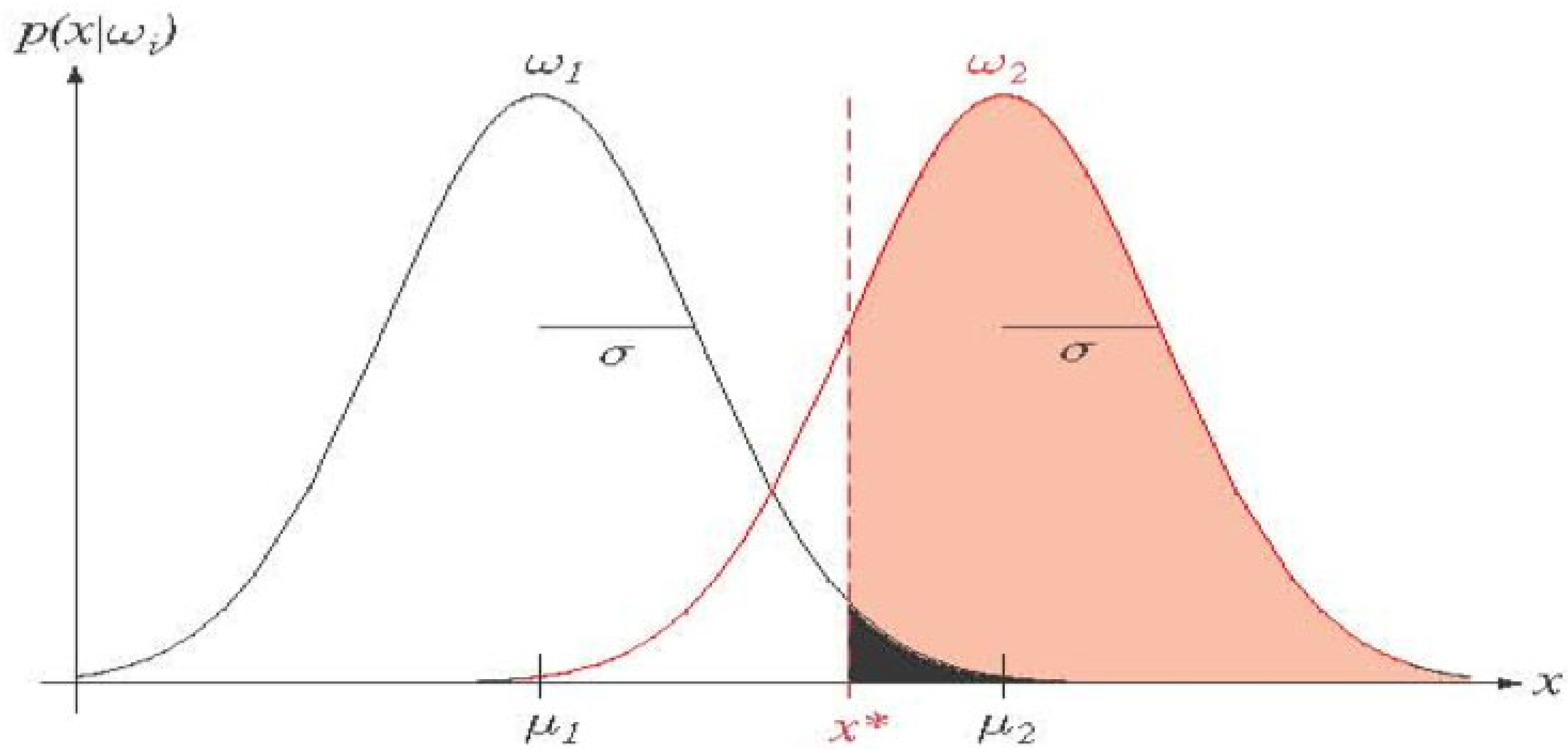
- It is a function of the *accuracy* parameter ϵ and the *confidence* parameter δ
- It also depends on the properties of the hypothesis class $\mathcal{H}$.
  - If $\mathcal{H}$ is PAC learnable, there are many functions $m_{\mathcal{H}}$ that satisfy the requirements given in the PAC learnability definition.
  - We define the sample complexity to be the "*minimal function*"

Every finite hypothesis class $\mathcal{H}$ is PAC learnable with the sample complexity:

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \rceil = \lceil \frac{1}{\epsilon}[\log(|\mathcal{H}|) + \log(\frac{1}{\delta})] \rceil$$

Is there a learner?

In many scenarios, there is no perfect learner:



General PAC Learning Model

PAC learning model can be generalized in two aspects:

PAC learning model can be generalized in two aspects:

Relaxing the Realizability Assumption

- We assume that labels are generated by some $f \in \mathscr{H}$, this assumption may be too strong.

General PAC Learning Model

PAC learning model can be generalized in two aspects:

Relaxing the Realizability Assumption

- We assume that labels are generated by some $f \in \mathcal{H}$, this assumption may be too strong.

Learning beyond Binary Classification

- Many learning tasks involve multiple class classification
- or even prediction of a real valued number.

Relaxing the Realizability Assumption:

General PAC Learning — Relaxing the Realizability Assumption

Relaxing the Realizability Assumption:

Intuition.

- Relax the realizability assumption by replacing the "*target labelling function*" f with a more flexible notion, a data-labels generating distribution.

Relaxing the Realizability Assumption:

Intuition.

- Relax the realizability assumption by replacing the "*target labelling function*" f with a more flexible notion, a data-labels generating distribution.
    - In PAC model, $\mathscr{D}$ is a distribution over $\mathscr{X}$

Relaxing the Realizability Assumption:

Intuition.

- Relax the realizability assumption by replacing the "*target labelling function*" f with a more flexible notion, a data-labels generating distribution.
  - In PAC model, $\mathscr{D}$ is a distribution over $\mathscr{X}$
  - In this aspect, $\mathscr{D}$ is a distribution over $Z = \mathscr{X} \times \mathscr{Y}$

Relaxing the Realizability Assumption:

Intuition.

- Relax the realizability assumption by replacing the "*target labelling function*" f with a more flexible notion, a data-labels generating distribution.
  - In PAC model, $\mathscr{D}$ is a distribution over $\mathscr{X}$
  - In this aspect, $\mathscr{D}$ is a distribution over $Z = \mathscr{X} \times \mathscr{Y}$
- The *Generalization risk* is then defined as:

$$L_{\mathscr{D}}(h) \stackrel{def}{=} P_{Z\sim\mathscr{D}}[h(x) \neq y] \stackrel{def}{=} \mathscr{D}(\{x : h(x) \neq y\})$$

Relaxing the Realizability Assumption:

Intuition.

- Relax the realizability assumption by replacing the "*target labelling function*" f with a more flexible notion, a data-labels generating distribution.
  - In PAC model, $\mathscr{D}$ is a distribution over $\mathscr{X}$
  - In this aspect, $\mathscr{D}$ is a distribution over $Z = \mathscr{X} \times \mathscr{Y}$
- The *Generalization risk* is then defined as:

$$L_{\mathscr{D}}(h) \stackrel{def}{=} P_{Z\sim\mathscr{D}}[h(x) \neq y] \stackrel{def}{=} \mathscr{D}(\{x : h(x) \neq y\})$$

- The notation of "*approximately correct*" is now defined as:

$$L_{\mathscr{D}}(h) \leq \min_{h^* \in \mathscr{H}} L_{\mathscr{D}}(h^*) + \epsilon$$

□

Scope of Learning Problems.

General PAC Learning — Beyond Binary Classification

Scope of Learning Problems.

Muticlass categorization $\mathscr{Y}$ is a finite set representing $|\mathscr{Y}|$ different classes.

- For example, the degree could be $\mathscr{Y} = \{Bachelor, Honours, Masters, PhD\}$

General PAC Learning — Beyond Binary Classification

Scope of Learning Problems.

Muticlass categorization $\mathscr{Y}$ is a finite set representing $|\mathscr{Y}|$ different classes.

- For example, the degree could be $\mathscr{Y} = \{Bachelor, Honours, Masters, PhD\}$

Regression $\mathscr{Y} = \mathscr{R}$

- For example, one wishes to predict the marks of a student based on the resources access pattern.

□

General PAC Learning — Loss Functions

- Let $Z = \mathscr{X} \times \mathscr{Y}$

General PAC Learning — Loss Functions

- Let $Z = \mathcal{X} \times \mathcal{Y}$
- Given hypothesis $h \in \mathcal{H}$, and an example $(x, y) \in Z$, how good is h on (x, y)?

General PAC Learning — Loss Functions

- Let $Z = \mathscr{X} \times \mathscr{Y}$
- Given hypothesis $h \in \mathscr{H}$, and an example $(x, y) \in Z$, how good is h on (x, y)?
- Loss Function:

$$l : \mathscr{H} \times Z \rightarrow \mathscr{R}_+$$

General PAC Learning — Loss Functions

- Let $Z = \mathscr{X} \times \mathscr{Y}$
- Given hypothesis $h \in \mathscr{H}$, and an example $(x, y) \in Z$, how good is h on (x, y)?
- Loss Function:

$$l : \mathscr{H} \times Z \to \mathscr{R}_+$$

0-1 loss $l(h,(x,y)) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 0 & \text{if } h(x) = y \end{cases}$

General PAC Learning — Loss Functions

- Let $Z = \mathscr{X} \times \mathscr{Y}$
- Given hypothesis $h \in \mathscr{H}$, and an example $(x, y) \in Z$, how good is h on (x, y)?
- Loss Function:

$$l : \mathscr{H} \times Z \to \mathscr{R}_+$$

0-1 loss $l(h, (x, y)) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 0 & \text{if } h(x) = y \end{cases}$

Squared loss $l(h, (x, y)) = (h(x) - y)^2$

General PAC Learning — Loss Functions

- Let $Z = \mathscr{X} \times \mathscr{Y}$
- Given hypothesis $h \in \mathscr{H}$, and an example $(x, y) \in Z$, how good is h on (x, y)?
- Loss Function:

$$l : \mathscr{H} \times Z \to \mathscr{R}_+$$

0-1 loss $l(h,(x,y)) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 0 & \text{if } h(x) = y \end{cases}$

Squared loss $l(h,(x,y)) = (h(x) - y)^2$

Absolute-value loss $l(h,(x,y)) = |h(x) - y|$

General PAC Learning — Loss Functions

- Let $Z = \mathscr{X} \times \mathscr{Y}$
- Given hypothesis $h \in \mathscr{H}$, and an example $(x, y) \in Z$, how good is h on (x, y)?
- Loss Function:

$$l : \mathscr{H} \times Z \to \mathscr{R}_+$$

0-1 loss $l(h,(x,y)) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 0 & \text{if } h(x) = y \end{cases}$

Squared loss $l(h,(x,y)) = (h(x) - y)^2$

Absolute-value loss $l(h,(x,y)) = |h(x) - y|$

Cost-sensitive loss $l(h,(x,y)) = C_{h(x),y}$, where C is $|\mathscr{Y}| \times |\mathscr{Y}|$ matrix.

Agnostic PAC Learnability

A hypothesis class $\mathscr{H}$ is agnostic PAC learnable with respect to a set Z and a loss function $l : \mathscr{H} \times Z \to \mathscr{R}_+$, if there exists a function $m_{\mathscr{H}} : (0,1)^2 \to \mathscr{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathscr{D}$ over Z, when running the algorithm on $m \geq m_{\mathscr{H}}(\epsilon,\delta)$ i.i.d. examples generated by $\mathscr{D}$, the algorithm returns a hypothesis $h \in \mathscr{H}$ such that, with probability of at least $(1-\delta)$: $\min_{h^* \in \mathscr{H}} L_{\mathscr{D}}(h^*) + \epsilon$

Agnostic PAC Learnability

A hypothesis class $\mathscr{H}$ is agnostic PAC learnable with respect to a set Z and a loss function $l : \mathscr{H} \times Z \rightarrow \mathscr{R}_+$, if there exists a function $m_{\mathscr{H}} : (0,1)^2 \rightarrow \mathscr{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathscr{D}$ over Z, when running the algorithm on $m \geq m_{\mathscr{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathscr{D}$, the algorithm returns a hypothesis $h \in \mathscr{H}$ such that, with probability of at least $(1-\delta)$: $\min_{h^* \in \mathscr{H}} L_{\mathscr{D}}(h^*) + \epsilon$

Probably (at least $(1-\delta)$ probability) **A**pproximately (up to accuracy ϵ) **C**orrect solve:

$$\min_{h \in \mathscr{H}} L_{\mathscr{D}}(h), \quad \text{where } L_{\mathscr{D}}(h) \stackrel{def}{=} E_{z \sim \mathscr{D}}[l(h,z)]$$

Agnostic PAC Learnability

A hypothesis class $\mathscr{H}$ is agnostic PAC learnable with respect to a set Z and a loss function $l : \mathscr{H} \times Z \to \mathscr{R}_+$, if there exists a function $m_{\mathscr{H}} : (0,1)^2 \to \mathscr{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathscr{D}$ over Z, when running the algorithm on $m \geq m_{\mathscr{H}}(\epsilon,\delta)$ i.i.d. examples generated by $\mathscr{D}$, the algorithm returns a hypothesis $h \in \mathscr{H}$ such that, with probability of at least $(1-\delta)$: $\min_{h^* \in \mathscr{H}} L_{\mathscr{D}}(h^*) + \epsilon$

Probably (at least $(1-\delta)$ probability) **A**pproximately (up to accuracy ϵ) **C**orrect solve:

$$\min_{h \in \mathscr{H}} L_{\mathscr{D}}(h), \quad \text{where } L_{\mathscr{D}}(h) \stackrel{def}{=} E_{z \sim \mathscr{D}}[l(h,z)]$$

- Learner knows $\mathscr{H}$, Z and l
- The learner receives the *accuracy* parameter ϵ and the *confidence* parameter δ

Agnostic PAC Learnability

A hypothesis class $\mathcal{H}$ is agnostic PAC learnable with respect to a set Z and a loss function $l : \mathcal{H} \times Z \to \mathcal{R}_+$, if there exists a function $m_{\mathcal{H}} : (0,1)^2 \to \mathcal{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathcal{D}$ over Z, when running the algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$, the algorithm returns a hypothesis $h \in \mathcal{H}$ such that, with probability of at least $(1-\delta)$: $\min_{h^* \in \mathcal{H}} L_{\mathcal{D}}(h^*) + \epsilon$

Probably (at least $(1-\delta)$ probability) **A**pproximately (up to accuracy ϵ) **C**orrect solve:

$$\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h), \quad \text{where } L_{\mathcal{D}}(h) \stackrel{def}{=} E_{z \sim \mathcal{D}}[l(h,z)]$$

- Learner knows $\mathcal{H}$, Z and l
- The learner receives the *accuracy* parameter ϵ and the *confidence* parameter δ
- The learner can decide on training set size m based on ϵ and δ.

A hypothesis class $\mathcal{H}$ is agnostic PAC learnable with respect to a set Z and a loss function $l : \mathcal{H} \times Z \to \mathcal{R}_+$, if there exists a function $m_{\mathcal{H}} : (0,1)^2 \to \mathcal{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathcal{D}$ over Z, when running the algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$, the algorithm returns a hypothesis $h \in \mathcal{H}$ such that, with probability of at least $(1-\delta)$: $\min_{h^* \in \mathcal{H}} L_{\mathcal{D}}(h^*) + \epsilon$

Probably (at least $(1-\delta)$ probability) **A**pproximately (up to accuracy ϵ) **C**orrect solve:

$$\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h), \quad \text{where } L_{\mathcal{D}}(h) \stackrel{def}{=} E_{z \sim \mathcal{D}}[l(h,z)]$$

- Learner knows $\mathcal{H}$, Z and l
- The learner receives the *accuracy* parameter ϵ and the *confidence* parameter δ
- The learner can decide on training set size m based on ϵ and δ.
- The learner does not know $\mathcal{D}$ but can sample $S \sim \mathcal{D}^m$

Agnostic PAC Learnability

A hypothesis class $\mathscr{H}$ is agnostic PAC learnable with respect to a set Z and a loss function $l : \mathscr{H} \times Z \to \mathscr{R}_+$, if there exists a function $m_{\mathscr{H}} : (0,1)^2 \to \mathscr{N}$ and a learning algorithm with the following property:

- For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathscr{D}$ over Z, when running the algorithm on $m \geq m_{\mathscr{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathscr{D}$, the algorithm returns a hypothesis $h \in \mathscr{H}$ such that, with probability of at least $(1-\delta)$: $\min_{h^* \in \mathscr{H}} L_{\mathscr{D}}(h^*) + \epsilon$

Probably (at least $(1-\delta)$ probability) **A**pproximately (up to accuracy ϵ) **C**orrect solve:

$$\min_{h \in \mathscr{H}} L_{\mathscr{D}}(h), \quad \text{where } L_{\mathscr{D}}(h) \stackrel{def}{=} E_{z \sim \mathscr{D}}[l(h,z)]$$

- Learner knows $\mathscr{H}$, Z and l
- The learner receives the *accuracy* parameter ϵ and the *confidence* parameter δ
- The learner can decide on training set size m based on ϵ and δ.
- The learner does not know $\mathscr{D}$ but can sample $S \sim \mathscr{D}^m$
- Using S the learner outputs some hypothesis $h \in \mathscr{H}$, with probability of at least $(1-\delta)$ it holds that $L_{\mathscr{D}}(h) \leq \min_{h^* \in \mathscr{H}} L_{\mathscr{D}}(h^*) + \epsilon$.

PAC versus Agnostic PAC Learning

Table 1: Comparison of PAC and Agnostic PAC

	PAC	**Agnostic PAC**
Distribution	$\mathscr{D}$ over $\mathscr{X}$	$\mathscr{D}$ over $\mathscr{X} \times \mathscr{Y}$
Truth	$f \in \mathscr{H}$	not in class or does not exist
Risk	$L_{(\mathscr{D},f)}(h) = \mathscr{D}(\{x : h(x) \neq f(x)\})$	$L_{\mathscr{D}}(h) = \mathscr{D}(\{x : h(x) \neq y\})$
Training set	$(x_1, \cdots, x_m) \sim \mathscr{D}^m$, $\forall i,\ y_i = f(x_i)$	$((x_1, y_1), \cdots, (x_m, y_m)) \sim \mathscr{D}^m$
Goal	$L_{(\mathscr{D},f)}(h) \leq \epsilon$	$L_{\mathscr{D}}(h) \leq \min_{h^* \in \mathscr{H}} L_{\mathscr{D}}(h^*) + \epsilon$

$\mathscr{X}$: Domain $\mathscr{Y}$: Range $\mathscr{H}$: Hypothesis Class
L: Loss function ϵ: accuracy parameter m: sample size

Agnostic Learning Finite Hypothesis Classes

Representative Sample

A training set S is called ϵ-representative w.r.t. domain Z, hypothesis class $\mathcal{H}$, loss function l and distribution $\mathcal{D}$, if

$$\forall h \in \mathcal{H}, \quad |L_S(h) - L_{\mathcal{D}}(h)| \leq \epsilon$$

Representative Sample

A training set S is called ϵ-representative w.r.t. domain Z, hypothesis class $\mathscr{H}$, loss function l and distribution $\mathscr{D}$, if

$$\forall h \in \mathscr{H}, \quad |L_S(h) - L_{\mathscr{D}}(h)| \leq \epsilon$$

Intuition.

- The hope is that an h that minimizes the empirical risk with respect to the sample S is a risk minimizer, or has risk close to the minimum, with respect to the true data probability distribution $\mathscr{D}$.

Representative Sample

A training set S is called ϵ-representative w.r.t. domain Z, hypothesis class $\mathcal{H}$, loss function l and distribution $\mathcal{D}$, if

$$\forall h \in \mathcal{H}, \quad |L_S(h) - L_{\mathcal{D}}(h)| \leq \epsilon$$

Intuition.

- The hope is that an h that minimizes the empirical risk with respect to the sample S is a risk minimizer, or has risk close to the minimum, with respect to the true data probability distribution $\mathcal{D}$.
- This concept ensures that: uniformly over *all hypotheses* in the hypothesis class $\mathcal{H}$, the empirical risk will be *close to the true* risk.

□

Representative Sample

Assume that a training set S is $\frac{\epsilon}{2}$-representative w.r.t. domain Z, hypothesis class $\mathscr{H}$, loss function l and distribution $\mathscr{D}$, then, any output of $ERM_{\mathscr{H}}(S)$, namely any $h_S \in \operatorname{argmin}_{h \in \mathscr{H}} L_S(h)$

$$L_{\mathscr{D}}(h_S) \leq \min_{h^* \in \mathscr{H}} L_{\mathscr{D}}(h^*) + \epsilon$$

Representative Sample

Assume that a training set S is $\frac{\epsilon}{2}$-representative w.r.t. domain Z, hypothesis class $\mathscr{H}$, loss function l and distribution $\mathscr{D}$, then, any output of $ERM_{\mathscr{H}}(S)$, namely any $h_S \in \text{argmin}_{h\in\mathscr{H}} L_S(h)$

$$L_{\mathscr{D}}(h_S) \leq \min_{h^*\in\mathscr{H}} L_{\mathscr{D}}(h^*) + \epsilon$$

Proof.

- $L_{\mathscr{D}}(h_S) \leq L_S(h_S) + \frac{\epsilon}{2}$

Representative Sample

Assume that a training set S is $\frac{\epsilon}{2}$-representative w.r.t. domain Z, hypothesis class $\mathscr{H}$, loss function l and distribution $\mathscr{D}$, then, any output of $ERM_{\mathscr{H}}(S)$, namely any $h_S \in \operatorname{argmin}_{h \in \mathscr{H}} L_S(h)$

$$L_{\mathscr{D}}(h_S) \leq \min_{h^* \in \mathscr{H}} L_{\mathscr{D}}(h^*) + \epsilon$$

Proof.

- $L_{\mathscr{D}}(h_S) \leq L_S(h_S) + \frac{\epsilon}{2}$
- $L_S(h^*) \leq L_{\mathscr{D}}(h^*) + \frac{\epsilon}{2}$

Assume that a training set S is $\frac{\epsilon}{2}$-representative w.r.t. domain Z, hypothesis class $\mathcal{H}$, loss function l and distribution $\mathcal{D}$, then, any output of $ERM_{\mathcal{H}}(S)$, namely any $h_S \in \mathrm{argmin}_{h\in\mathcal{H}} L_S(h)$

$$L_{\mathcal{D}}(h_S) \le \min_{h^*\in\mathcal{H}} L_{\mathcal{D}}(h^*) + \epsilon$$

Proof.

- $L_{\mathcal{D}}(h_S) \le L_S(h_S) + \frac{\epsilon}{2}$
- $L_S(h^*) \le L_{\mathcal{D}}(h^*) + \frac{\epsilon}{2}$
- Combine them together, we have

$$\begin{aligned} L_{\mathcal{D}}(h_S) &\le L_S(h_S) + \frac{\epsilon}{2} \\ &\le L_S(h^*) + \frac{\epsilon}{2} \\ &\le L_{\mathcal{D}}(h^*) + \frac{\epsilon}{2} + \frac{\epsilon}{2} \\ &= L_{\mathcal{D}}(h^*) + \epsilon \end{aligned}$$

□

Uniform Convergence

A hypothesis class $\mathscr{H}$ has the uniform convergence property if there exists a function $m_{\mathscr{H}}^{UC} : (0,1)^2 \rightarrow \mathscr{N}$, such that for every $\epsilon, \delta \in (0,1)$, and every distribution $\mathscr{D}$, we have: if S is a sample with $m \geq m_{\mathscr{H}}^{UC}(\epsilon, \delta)$ examples drawn i.i.d. according to $\mathscr{D}$, then with probability of at least $(1-\delta)$, S is ϵ-representative.

Uniform Convergence

A hypothesis class $\mathcal{H}$ has the uniform convergence property if there exists a function $m_{\mathcal{H}}^{UC} : (0,1)^2 \to \mathcal{N}$, such that for every $\epsilon, \delta \in (0,1)$, and every distribution $\mathcal{D}$, we have: if S is a sample with $m \geq m_{\mathcal{H}}^{UC}(\epsilon,\delta)$ examples drawn i.i.d. according to $\mathcal{D}$, then with probability of at least $(1-\delta)$, S is ϵ-representative.

If a class $\mathcal{H}$ has the *uniform convergence property* with the sample complexity $m_{\mathcal{H}}^{UC}$, then $\mathcal{H}$ is *agnostically PAC learnable* with the sample complexity

$$m_{\mathcal{H}}(\epsilon,\delta) \leq m_{\mathcal{H}}^{UC}(\frac{\epsilon}{2},\delta)$$

Furthermore, $ERM_{\mathcal{H}}$ paradigm is a successful agnostic PAC learner for $\mathcal{H}$.

Uniform Convergence

A hypothesis class $\mathcal{H}$ has the uniform convergence property if there exists a function $m_{\mathcal{H}}^{UC} : (0,1)^2 \to \mathcal{N}$, such that for every $\epsilon, \delta \in (0,1)$, and every distribution $\mathcal{D}$, we have: if S is a sample with $m \geq m_{\mathcal{H}}^{UC}(\epsilon,\delta)$ examples drawn i.i.d. according to $\mathcal{D}$, then with probability of at least $(1-\delta)$, S is ϵ-representative.

If a class $\mathcal{H}$ has the *uniform convergence property* with the sample complexity $m_{\mathcal{H}}^{UC}$, then $\mathcal{H}$ is *agnostically PAC learnable* with the sample complexity

$$m_{\mathcal{H}}(\epsilon,\delta) \leq m_{\mathcal{H}}^{UC}(\frac{\epsilon}{2},\delta)$$

Furthermore, $ERM_{\mathcal{H}}$ paradigm is a successful agnostic PAC learner for $\mathcal{H}$.

- $m_{\mathcal{H}}^{UC}$ measures the minimal sample complexity of obtaining the uniform convergence.

Agnostic Learning Finite Hypothesis Classes

Assume $\mathscr{H}$ is finite and the range of the loss function is $[0,1]$, then $\mathscr{H}$ is agnostic PAC learnable using the $ERM_{\mathscr{H}}$ algorithm with sample complexity:

$$m_{\mathscr{H}}(\epsilon,\delta) \leq \lceil \frac{2\log(2|\mathscr{H}|/\delta)}{\epsilon^2} \rceil = \lceil \frac{2}{\epsilon^2}[\log(2|\mathscr{H}|) + \log(\frac{1}{\delta})] \rceil$$

Agnostic Learning Finite Hypothesis Classes

Assume $\mathscr{H}$ is finite and the range of the loss function is $[0,1]$, then $\mathscr{H}$ is agnostic PAC learnable using the $ERM_{\mathscr{H}}$ algorithm with sample complexity:

$$m_{\mathscr{H}}(\epsilon,\delta) \leq \lceil \frac{2\log(2|\mathscr{H}|/\delta)}{\epsilon^2} \rceil = \lceil \frac{2}{\epsilon^2}[\log(2|\mathscr{H}|) + \log(\frac{1}{\delta})] \rceil$$

Proof. It suffices to show that $\mathscr{H}$ has the uniform convergence property with

$$m^{UC}_{\mathscr{H}}(\epsilon,\delta) \leq \lceil \frac{\log(2|\mathscr{H}|/\delta)}{2\epsilon^2} \rceil$$

Assume $\mathcal{H}$ is finite and the range of the loss function is $[0,1]$, then $\mathcal{H}$ is agnostic PAC learnable using the $ERM_{\mathcal{H}}$ algorithm with sample complexity:

$$m_{\mathcal{H}}(\epsilon,\delta) \leq \lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2} \rceil = \lceil \frac{2}{\epsilon^2}[\log(2|\mathcal{H}|)+\log(\frac{1}{\delta})] \rceil$$

Proof. It suffices to show that $\mathcal{H}$ has the uniform convergence property with

$$m_{\mathcal{H}}^{UC}(\epsilon,\delta) \leq \lceil \frac{\log(2|\mathcal{H}|/\delta)}{2\epsilon^2} \rceil$$

1. To show uniform convergence, we need: $\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) < \delta$

Agnostic Learning Finite Hypothesis Classes

Assume $\mathcal{H}$ is finite and the range of the loss function is $[0,1]$, then $\mathcal{H}$ is agnostic PAC learnable using the $ERM_{\mathcal{H}}$ algorithm with sample complexity:

$$m_{\mathcal{H}}(\epsilon,\delta) \le \lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2} \rceil = \lceil \frac{2}{\epsilon^2}[\log(2|\mathcal{H}|) + \log(\frac{1}{\delta})] \rceil$$

Proof. It suffices to show that $\mathcal{H}$ has the uniform convergence property with

$$m_{\mathcal{H}}^{UC}(\epsilon,\delta) \le \lceil \frac{\log(2|\mathcal{H}|/\delta)}{2\epsilon^2} \rceil$$

1. To show uniform convergence, we need: $\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) < \delta$
2. From the union bound, we have:

$$\begin{aligned} &\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) \\ &= \mathcal{D}^m(\bigcup_{h\in\mathcal{H}} \{S : |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) \\ &\le \sum_{h\in\mathcal{H}} \mathcal{D}^m(\{S : |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) \end{aligned}$$

$\square$

Agnostic Learning Finite Hypothesis Classes

Assume $\mathcal{H}$ is finite and the range of the loss function is $[0,1]$, then $\mathcal{H}$ is agnostic PAC learnable using the $ERM_{\mathcal{H}}$ algorithm with sample complexity:

$$m_{\mathcal{H}}(\epsilon,\delta) \leq \lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2} \rceil = \lceil \frac{2}{\epsilon^2}[\log(2|\mathcal{H}|) + \log(\frac{1}{\delta})] \rceil$$

Proof.

3. $L_{\mathcal{D}}(h) = E_{z\sim\mathcal{D}}[l(h,z)]$ and $L_S(h) = \frac{1}{m}\sum_{i=1}^{m} l(h,z_i)$, let $\theta_i = l(h,z_i)$

Assume $\mathscr{H}$ is finite and the range of the loss function is $[0,1]$, then $\mathscr{H}$ is agnostic PAC learnable using the $ERM_{\mathscr{H}}$ algorithm with sample complexity:

$$m_{\mathscr{H}}(\epsilon,\delta) \leq \lceil \frac{2\log(2|\mathscr{H}|/\delta)}{\epsilon^2} \rceil = \lceil \frac{2}{\epsilon^2}[\log(2|\mathscr{H}|) + \log(\frac{1}{\delta})] \rceil$$

Proof.

3. $L_{\mathscr{D}}(h) = E_{z\sim\mathscr{D}}[l(h,z)]$ and $L_S(h) = \frac{1}{m}\sum_{i=1}^{m} l(h,z_i)$, let $\theta_i = l(h,z_i)$
4. For all i, $E[\theta_i] = L_{\mathscr{D}}(h)$

Assume $\mathcal{H}$ is finite and the range of the loss function is $[0,1]$, then $\mathcal{H}$ is agnostic PAC learnable using the $ERM_{\mathcal{H}}$ algorithm with sample complexity:

$$m_{\mathcal{H}}(\epsilon,\delta) \leq \lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2} \rceil = \lceil \frac{2}{\epsilon^2}[\log(2|\mathcal{H}|) + \log(\frac{1}{\delta})] \rceil$$

Proof.

3. $L_{\mathcal{D}}(h) = E_{z\sim\mathcal{D}}[l(h,z)]$ and $L_S(h) = \frac{1}{m}\sum_{i=1}^{m} l(h,z_i)$, let $\theta_i = l(h,z_i)$
4. For all i, $E[\theta_i] = L_{\mathcal{D}}(h)$
5. From *Hoeffding's inequality*:

$$\mathcal{D}^m(\{S : |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) \leq 2e^{-2m\epsilon^2}$$

Agnostic Learning Finite Hypothesis Classes

Assume $\mathscr{H}$ is finite and the range of the loss function is $[0,1]$, then $\mathscr{H}$ is agnostic PAC learnable using the $ERM_{\mathscr{H}}$ algorithm with sample complexity:

$$m_{\mathscr{H}}(\epsilon,\delta) \le \lceil \frac{2\log(2|\mathscr{H}|/\delta)}{\epsilon^2} \rceil = \lceil \frac{2}{\epsilon^2}[\log(2|\mathscr{H}|) + \log(\frac{1}{\delta})] \rceil$$

Proof.

3. $L_{\mathscr{D}}(h) = E_{z\sim\mathscr{D}}[l(h,z)]$ and $L_S(h) = \frac{1}{m}\sum_{i=1}^{m} l(h,z_i)$, let $\theta_i = l(h,z_i)$
4. For all i, $E[\theta_i] = L_{\mathscr{D}}(h)$
5. From *Hoeffding's inequality*:

$$\mathscr{D}^m(\{S : |L_S(h) - L_{\mathscr{D}}(h)| > \epsilon\}) \le 2e^{-2m\epsilon^2}$$

6. We have:

$$\mathscr{D}^m(\{S : \exists h \in \mathscr{H}, |L_S(h) - L_{\mathscr{D}}(h)| > \epsilon\}) \le \sum_{h\in\mathscr{H}} \mathscr{D}^m(\{S : |L_S(h) - L_{\mathscr{D}}(h)| > \epsilon\}) \le 2|\mathscr{H}|e^{-2m\epsilon^2}$$

Agnostic Learning Finite Hypothesis Classes

Assume $\mathcal{H}$ is finite and the range of the loss function is $[0,1]$, then $\mathcal{H}$ is agnostic PAC learnable using the $ERM_{\mathcal{H}}$ algorithm with sample complexity:

$$m_{\mathcal{H}}(\epsilon,\delta) \leq \lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2} \rceil = \lceil \frac{2}{\epsilon^2}[\log(2|\mathcal{H}|) + \log(\frac{1}{\delta})] \rceil$$

Proof.

3. $L_{\mathcal{D}}(h) = E_{z\sim\mathcal{D}}[l(h,z)]$ and $L_S(h) = \frac{1}{m}\sum_{i=1}^m l(h,z_i)$, let $\theta_i = l(h,z_i)$
4. For all i, $E[\theta_i] = L_{\mathcal{D}}(h)$
5. From *Hoeffding's inequality*:

$$\mathcal{D}^m(\{S : |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) \leq 2e^{-2m\epsilon^2}$$

6. We have:

$$\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) \leq \sum_{h\in\mathcal{H}} \mathcal{D}^m(\{S : |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) \leq 2|\mathcal{H}|e^{-2m\epsilon^2}$$

7. So if $m \geq \frac{\log(2|\mathcal{H}|/\delta)}{2\epsilon^2}$, we have the right hand side is at most δ as required. $\square$

The Discretization Trick

- Suppose $\mathscr{H}$ is parametrized by d numbers.

The Discretization Trick

- Suppose $\mathscr{H}$ is parametrized by d numbers.
- Suppose we are happy with a representation of each number using b bits

The Discretization Trick

- Suppose $\mathscr{H}$ is parametrized by d numbers.
- Suppose we are happy with a representation of each number using b bits
- Then $|\mathscr{H}| \leq 2^{db}$, and so

$$m_{\mathscr{H}}(\epsilon, \delta) \leq \left\lceil \frac{2db + 2\log(2/\delta)}{\epsilon^2} \right\rceil$$

The Discretization Trick

- Suppose $\mathscr{H}$ is parametrized by d numbers.
- Suppose we are happy with a representation of each number using b bits
- Then $|\mathscr{H}| \leq 2^{db}$, and so

$$m_{\mathscr{H}}(\epsilon, \delta) \leq \left\lceil \frac{2db + 2\log(2/\delta)}{\epsilon^2} \right\rceil$$

- While not very elegant, it is a great tool for upper bounding *sample complexity*.

Quiz

Theoretical analysis:

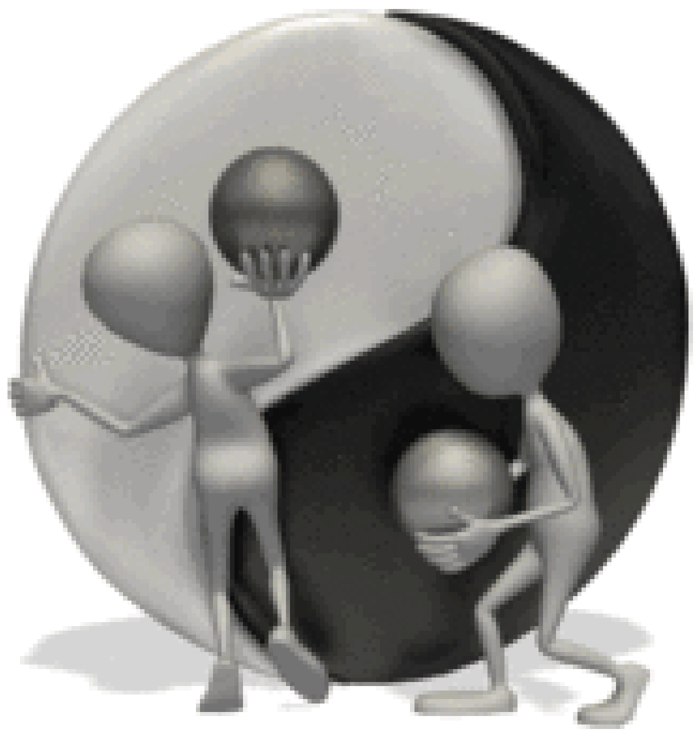
1. If the range of the loss function is $[a, b]$, then the sample complexity satisfies:

$$m_{\mathcal{H}}(\epsilon, \delta) \leq m_{\mathcal{H}}^{UC}(\epsilon/2, \delta) \leq \lceil \frac{2 \log 2|\mathcal{H}/\delta|(b-a)^2}{\epsilon^2} \rceil .$$

Theoretical analysis:

1. If the range of the loss function is $[a, b]$, then the sample complexity satisfies:
$$m_{\mathscr{H}}(\epsilon, \delta) \le m_{\mathscr{H}}^{UC}(\epsilon/2, \delta) \le \lceil \frac{2 \log 2|\mathscr{H}/\delta|(b-a)^2}{\epsilon^2} \rceil .$$
2. Given any probability distribution $\mathscr{D}$ over $\mathscr{X} \times \{0, 1\}$, the *Bayes Optimal Predictor* is defined as: $f_{\mathscr{D}}(x) = \begin{cases} 1 & \text{if } P[y=1|x] \ge \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$ Show that for every probability distribution $\mathscr{D}$, the *Bayes Optimal Predictor* $f_{\mathscr{D}}$ is optimal, in the sense that for every classifier g from $\mathscr{X}$ to $\{0, 1\}$, we have $L_{\mathscr{D}}(f_{\mathscr{D}}) \le L_{\mathscr{D}}(g)$.

Questions?



Contact Information

Associate Professor **Gang Li**
School of Information Technology
Deakin University
Geelong, Victoria 3216, Australia

gangli@tulip.org.au
Open Resources of TULIP-Lab
Team for Universal Learning and Intelligent Processing