

# Machine learning theory

## Model Selection

Hamid Beigy

Sharif University of Technology

May 29, 2023





1. Introduction
2. Universal learners
3. Estimation and approximation errors
4. Empirical risk minimization
5. Structural risk minimization
6. Cross-validation
7. n-Fold cross-validation
8. Regularization-based algorithms
9. Reading

## Introduction

---



1. The training data can mislead the learner and results in overfitting? **how?**
2. To overcome this problem, we restricted the search space to some hypothesis class  $H$ .
3. This hypothesis class can be viewed as reflecting some prior knowledge that the learner has about the task.
4. Is such prior knowledge really necessary for the success of learning?
5. Maybe there exists some kind of **universal learner** (a learner who has no prior knowledge about a certain task and is ready to be challenged by any task?)

## Universal learners

---



1. The **no-free lunch theorem** states that **no such universal learner exists**.
2. This theorem states that for binary classification prediction tasks, for every learner there exists a distribution on which it fails.

### Theorem (No-free lunch)

Let  $A$  be any learning algorithm for the task of binary classification with respect to the 0-1 loss over a domain  $\mathcal{X}$ . Let  $m$  be any number smaller than  $\frac{|\mathcal{X}|}{2}$ , representing a training set size. Then, there exists a distribution  $\mathcal{D}$  over  $\mathcal{X} \times \{0, 1\}$  such that:

2.1 There exists a function  $h : \mathcal{X} \mapsto \{0, 1\}$  with  $\mathbf{R}(h) = 0$ .

2.2 With probability of at least  $\frac{1}{7}$  over the choice of  $S \sim \mathcal{D}^m$ , we have that  $\mathbf{R}(A(S)) \geq \frac{1}{7}$ .

3. This theorem states that for every learner, there exists a task on which it fails, even though that task can be successfully learned by another learner.
4. In other words, the theorem states that no learner can succeed on all learnable tasks, every learner has tasks on which it fails while other learners succeed.



1. How does the **No-Free-Lunch** result relate to the need for **prior knowledge**?

**Theorem**

*Let  $\mathcal{X}$  be an infinite domain set and let  $H$  be the set of all functions from  $\mathcal{X}$  to  $\{0, 1\}$ . Then,  $H$  is not PAC learnable.*

2. How can we prevent such failures?
3. We can escape the hazards by using our prior knowledge about a specific learning task, to avoid the distributions that will cause us to fail when learning that task.
4. Such prior knowledge can be expressed by **restricting our hypothesis class**.
5. But **how should we choose a good hypothesis class**?
6. We want to believe that this class includes the hypothesis that has no error at all (in the PAC setting), or at least that the smallest error achievable by a hypothesis from this class is indeed rather small (in the agnostic setting).
7. We have just seen that we cannot simply choose the richest class (the class of all functions over the given domain).
8. How can we have such **trade off**?

## Estimation and approximation errors

---





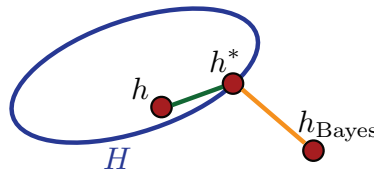
1. The answer to the trade off is to decompose  $\mathbf{R}(h)$ .
2. Let  $H$  be a family of functions mapping  $\mathcal{X}$  to  $\{0, 1\}$ .
3. The **excess error** of a hypothesis  $h$  chosen from  $H$  ( $\mathbf{R}(h) - \mathbf{R}^*$ ) can be decomposed as

$$\mathbf{R}(h) - \mathbf{R}^* = \underbrace{\left( \mathbf{R}(h) - \inf_{h' \in H} \mathbf{R}(h') \right)}_{\text{estimation error}} + \underbrace{\left( \inf_{h' \in H} \mathbf{R}(h') - \mathbf{R}^* \right)}_{\text{approximation error}}$$

4. The **estimation error** depends on the **hypothesis  $h$**  selected.
5. The **approximation error** measures how well the **Bayes error** can be **approximated using  $H$** . It is a property of the **hypothesis set  $H$** , a measure of its richness.



1. The **excess error** can be shown as



2. Model selection consists of choosing  $H$  with a **favorable trade-off** between the **approximation and estimation errors**.
3. The **approximation error is not accessible**, since in general the underlying distribution  $\mathcal{D}$  needed to determine  $\mathbf{R}^*$  is not known.
4. The **estimation error** of an algorithm  $A$ , that is, the estimation error of the hypothesis  $h$  returned after training on a sample  $S$ , can sometimes be bounded using generalization bounds.

## Empirical risk minimization

---



1. A standard algorithm for which the estimation error can be bounded is **empirical risk minimization (ERM)**.
2. ERM seeks to minimize the error on the training sample.

$$h_{erm} = \arg \min_{h \in H} \hat{\mathbf{R}}(h).$$

3. If there exists multiple hypotheses with minimal error on the training sample, then ERM returns an arbitrary one.

### **Theorem (ERM error bound)**

For any sample  $S$ , the following inequality holds for the hypothesis returned by ERM.

$$\mathbb{P} \left[ \mathbf{R}(h_{erm}) - \inf_{h \in H} \mathbf{R}(h) > \epsilon \right] \leq \mathbb{P} \left[ \sup_{h \in H} |\mathbf{R}(h_{erm}) - \hat{\mathbf{R}}(h)| > \frac{\epsilon}{2} \right].$$



**Proof.**

1. By definition of  $\inf_{h \in H} \mathbf{R}(h)$ , we mean for any  $\epsilon > 0$ , there exists  $h_\epsilon$  such that  $\mathbf{R}(h_\epsilon) \leq \inf_{h \in H} \mathbf{R}(h) + \epsilon$ .
2. By definition of ERM, we have  $\hat{\mathbf{R}}(h_{erm}) \leq \hat{\mathbf{R}}(h_\epsilon)$  and hence

$$\begin{aligned}
 \mathbf{R}(h_{erm}) - \inf_{h \in H} \mathbf{R}(h) &= \mathbf{R}(h_{erm}) - \mathbf{R}(h_\epsilon) + \mathbf{R}(h_\epsilon) - \inf_{h \in H} \mathbf{R}(h) \\
 &\leq \mathbf{R}(h_{erm}) - \mathbf{R}(h_\epsilon) + \epsilon && \text{from def. given in step 1} \\
 &= \mathbf{R}(h_{erm}) - \hat{\mathbf{R}}(h_{erm}) + \hat{\mathbf{R}}(h_{erm}) - \mathbf{R}(h_\epsilon) + \epsilon \\
 &\leq \mathbf{R}(h_{erm}) - \hat{\mathbf{R}}(h_{erm}) + \hat{\mathbf{R}}(h_\epsilon) - \mathbf{R}(h_\epsilon) + \epsilon && \text{from def. of ERM} \\
 &\leq 2 \sup_{h \in H} |\mathbf{R}(h) - \hat{\mathbf{R}}(h)| + \epsilon.
 \end{aligned}$$

3. Since the inequality holds for all  $\epsilon > 0$ , it implies

$$\mathbf{R}(h_{erm}) - \inf_{h \in H} \mathbf{R}(h) \leq 2 \sup_{h \in H} |\mathbf{R}(h) - \hat{\mathbf{R}}(h)|.$$

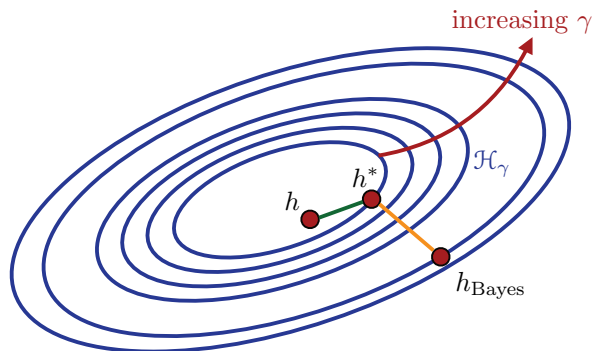
□

## Structural risk minimization

---



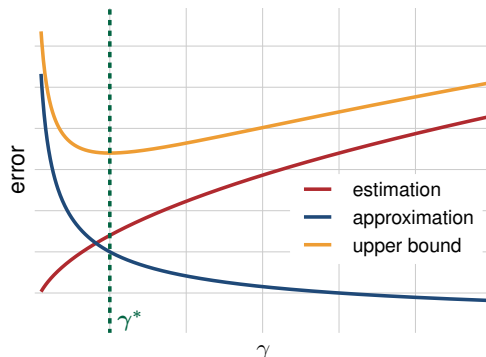
1. We showed that the estimation error can be bounded or estimated.
2. Since the approximation error cannot be estimated, how should we choose  $H$ ?
3. One way is to choose a very complex family  $H$  with no approximation error or a very small one.
4.  $H$  may be too rich for generalization bounds to hold for  $H$ .
5. Suppose we can decompose  $H$  as a union of increasingly  $\bigcup_{\gamma \in \Gamma} H_\gamma$  increasing with  $\gamma$  for some set  $\Gamma$ .



6. The problem then consists of selecting the parameter  $\gamma^* \in \Gamma$  and thus the hypothesis set  $H_{\gamma^*}$  with the most favorable trade-off between estimation and approximation errors.



1. Since estimation and approximation errors are not known, instead, a uniform upper bound on their sum can be used.

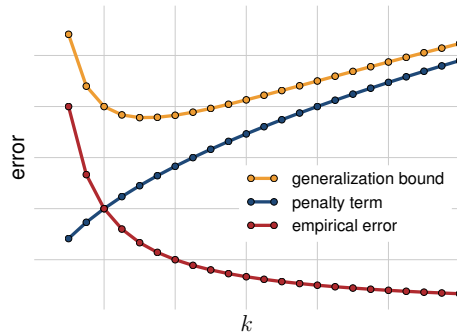


2. This is the idea behind the structural risk minimization (SRM) method.
3. For SRM,  $H$  is assumed to be decomposable into a countable set, thus, we write it as  $H = \bigcup_{k \geq 1} H_k$ .
4. Also, the hypothesis sets are nested, i.e.  $H_k \subset H_{k+1}$  for all  $k \geq 1$ .
5. However, many of the results presented here also hold for non-nested hypothesis sets.
6. SRM consists of choosing the index  $k^* \geq 1$  and the ERM hypothesis  $h \in H_{k^*}$  that minimize an upper bound on the excess error.





1. The hypothesis set for SRM:  $H = \bigcup_{k \geq 1} H_k$  with  $H_1 \subset H_2 \subset \dots \subset H_k \subset \dots$
2. Solution of SRM is  $h^* = \arg \min_{h \in H_k, k \geq 1} \hat{R}(h) + \text{pen}(k, m)$ .





### Definition (SRM)

- $H_{k(h)}$  is the simplest hypothesis set containing  $h$ .
- $h_{srm}$  is the hypothesis returned by SRM.

$$h_{srm} = \arg \min_{h \in H_k, k \geq 1} \hat{\mathbf{R}}(h) + \mathcal{R}_m(H_k) + \sqrt{\frac{\log k}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{2m}} = F_k(h)$$

- Assume that there exists  $h^* \in H$  such that  $h^* = \inf_{h \in H} \hat{\mathbf{R}}(h)$

### Theorem (SRM learning guarantee)

For any  $\delta > 0$ , with probability at least  $1 - \delta$  over sample  $S \in \mathcal{D}^m$ , the generalization error of the hypothesis  $h_{srm}$  returned by the SRM is bounded as follows:

$$\mathbf{R}(h_{srm}) \leq \hat{\mathbf{R}}(h) + 2\mathcal{R}_m(H_{k(h)}) + \sqrt{\frac{\log k(h)}{m}} + \sqrt{2 \frac{\log \frac{3}{\delta}}{m}}.$$



### Proof. (SRM learning guarantee)

1. Using the union bound, the following general inequality holds:

$$\begin{aligned}
 \mathbb{P} \left[ \sup_{h \in H} \mathbf{R}(h) - F_{k(h)}(h) > \epsilon \right] &= \mathbb{P} \left[ \sup_{k \geq 1} \sup_{h \in H_k} \mathbf{R}(h) - F_k(h) > \epsilon \right] \\
 &\leq \sum_{k=1}^{\infty} \mathbb{P} \left[ \sup_{h \in H_k} \mathbf{R}(h) - F_k(h) > \epsilon \right] \\
 &= \sum_{k=1}^{\infty} \mathbb{P} \left[ \sup_{h \in H_k} \mathbf{R}(h) - \hat{\mathbf{R}}(h) - \mathcal{R}_m(H_k) > \epsilon + \sqrt{\frac{\log k}{m}} \right] \\
 &\leq \sum_{k=1}^{\infty} \exp \left( -2m \left[ \epsilon + \sqrt{\frac{\log k}{m}} \right]^2 \right) \\
 &\leq \sum_{k=1}^{\infty} \exp \left( -2m\epsilon^2 \right) \exp \left( -2 \log k \right) \\
 &= \exp \left( -2m\epsilon^2 \right) \sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6} \exp \left( -2m\epsilon^2 \right) \\
 &\leq 2 \exp \left( -2m\epsilon^2 \right).
 \end{aligned}$$



### Proof. (SRM learning guarantee (Cont.))

2. For two random variables  $X_1$  and  $X_2$ , if  $X_1 + X_2 > \epsilon$ , then either  $X_1 > \frac{\epsilon}{2}$  or  $X_2 > \frac{\epsilon}{2}$ .

3. Let  $g(h) = \mathbf{R}(h_{srm}) - \mathbf{R}(h) - 2\mathcal{R}_m(H_k(h)) - \sqrt{\frac{\log k(h)}{m}}$ .

4. We also have  $F_{k(h_{srm})}(h_{srm}) \leq F_{k(h)}(h)$  for all  $h \in H$  and for all  $h \in H$ , we have.

$$\begin{aligned}
 \mathbb{P}[g(h) > \epsilon] &\leq \mathbb{P}\left[\mathbf{R}(h_{srm}) - F_{k(h_{srm})}(h) > \frac{\epsilon}{2}\right] + \mathbb{P}\left[F_{k(h_{srm})}(h_{srm}) - \mathbf{R}(h) - 2\mathcal{R}_m(H_k(h)) - \sqrt{\frac{\log k(h)}{m}} > \frac{\epsilon}{2}\right] \\
 &\leq 2 \exp\left(-\frac{m\epsilon^2}{2}\right) + \mathbb{P}\left[F_{k(h_{srm})}(h_{srm}) - \mathbf{R}(h) - 2\mathcal{R}_m(H_k(h)) - \sqrt{\frac{\log k(h)}{m}} > \frac{\epsilon}{2}\right] \\
 &= 2 \exp\left(-\frac{m\epsilon^2}{2}\right) + \mathbb{P}\left[\hat{\mathbf{R}}(h) - \mathbf{R}(h) - \mathcal{R}_m(H_k(h)) > \frac{\epsilon}{2}\right] \\
 &= 2 \exp\left(-\frac{m\epsilon^2}{2}\right) + \exp\left(-\frac{m\epsilon^2}{2}\right) \\
 &= 3 \exp\left(-\frac{m\epsilon^2}{2}\right).
 \end{aligned}$$

5. Setting the right-hand side to  $\delta$  and solving it, the proof will be completed. □



1. This bound is similar to learning bound when  $k(h^*)$  is known!
2. Can be extended if approximation error assumed to be small or zero.
3. if  $H$  contains the Bayes classifier, only finitely many hypothesis sets need to be considered in practice.
4. **Restriction:**  $H$  decomposed as countable union of families with converging Rademacher complexity.
5. **Issues:**
  - SRM typically computationally intractable;
  - how should we choose  $H$ s and  $h^*$ ?

## Cross-validation

---



1. An alternative method for model selection is **cross-validation**.
2. In **cross-validation**, we use some fraction of training set as **validation set** to select a **hypothesis set**  $H_k$ .
3. In **cross-validation**,  $S$  is divided into a sample  $S_1$  of size  $(1 - \alpha)m$  and a sample  $S_2$  of size  $\alpha m$ , with  $\alpha \in (0, 1)$ .
4. For any  $k \in \mathbb{N}$ , let  $h_{erm}^{S_1, k}$  be the solution of ERM run on  $S_1$  using the hypothesis set  $H_k$ .
5. The hypothesis  $h_{cv}$  returned by cross-validation is the ERM solution  $h_{erm}^{S_1, k}$  with the best performance on  $S_2$ .

$$h_{cv} = \arg \min_{h \in \{h_{erm}^{S_1, k} : k \geq 1\}} \hat{\mathbf{R}}_{S_2}(h)$$

### Theorem (Cross-validation bound)

For any  $\alpha > 0$  and any sample size  $m \geq 1$ , we have

$$\mathbb{P} \left[ \sup_{k \geq 1} \left| \mathbf{R}(h_{erm}^{S_1, k}) - \hat{\mathbf{R}}(h_{erm}^{S_1, k}) \right| > \epsilon + \sqrt{\frac{\log k}{\alpha m}} \right] \leq 4 \exp(-2\alpha m \epsilon^2)$$

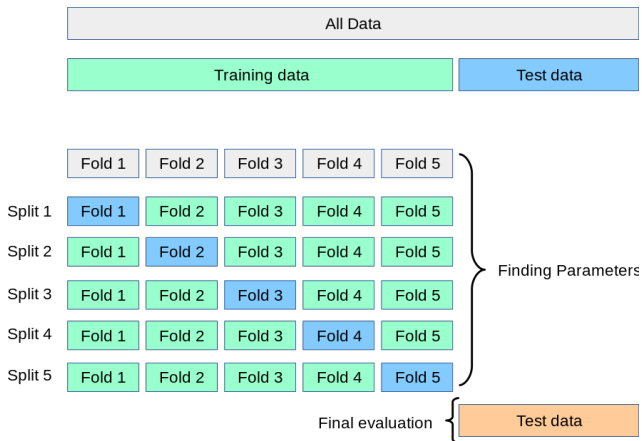
## **n-Fold cross-validation**

---





1. In practice, the amount of labeled data available is often too small to set aside a validation sample.
2. Instead, a widely adopted method known as **n-fold cross-validation** is used to exploit the labeled data both for model selection and for training.



3. The special case of **n-fold cross-validation** where  $n = m$  is called **leave-one-out cross-validation**.

## Regularization-based algorithms

---



1. A broad family of algorithms inspired by the SRM method is that of **regularization-based algorithm**.
2. This consists of selecting a **very complex family  $H$**  that is **an uncountable union of nested hypothesis sets  $H = \bigcup_{\gamma>0} H_\gamma$** .
3.  $H$  is often chosen to be dense in the space of continuous functions over  $\mathcal{X}$ .
4. For example,  $H$  may be chosen to be the set of all linear functions in some high-dimensional space and  $H_\gamma$  the subset of those functions whose norm is bounded by

$$H_\gamma = \{\mathcal{X} \mapsto \langle \mathbf{w}, \phi(x) \rangle \mid \|\mathbf{w}\| \leq \gamma\}.$$



1. Given a labeled sample  $S$ , the extension of the SRM method to an uncountable union would then suggest selecting  $h$  based on the following optimization problem:

$$h_{reg} = \arg \min_{\gamma > 0, h \in H_\gamma} \hat{\mathbf{R}}(h) + \mathcal{R}(H_\gamma) + \sqrt{\frac{\log \gamma}{m}}$$

2. Often, there exists a function  $R : H \mapsto \mathbb{R}$  such that, for any  $\gamma > 0$ , the constrained optimization problem  $\arg \min_{\gamma > 0, h \in H_\gamma} \hat{\mathbf{R}}(h) + \text{pen}(\gamma, m)$  can be equivalently written as the unconstrained optimization problem.

$$h_{reg} = \arg \min_{h \in H} \hat{\mathbf{R}}(h) + \lambda R(h)$$

3.  $\lambda > 0$  is called regularization parameter and  $R(h)$  is called regularization term.



## Reading

---



1. Chapter 5 of [Understanding machine learning : From theory to algorithms](#) (Shalev-Shwartz and Ben-David 2014).
2. Chapter 4 of [Foundations of machine learning](#) (Mohri, Rostamizadeh, and Talwalkar 2018).



-  Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar (2018). *Foundations of Machine Learning*. Second Edition. MIT Press.
-  Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

Questions?