



THE UNIVERSITY OF  
SYDNEY

# Advanced Machine Learning

(COMP 5328)

Week 5 Tutorial:  
Hypothesis Complexity and Generalisation

Anjin Liu  
[Anjin.liu@sydney.edu.au](mailto:Anjin.liu@sydney.edu.au)

# Tutorial Contents

- Review (20min):

Lecture 3: Hypothesis Complexity and Generalization

- Tutorial exercise (40min):

# Key points

- Hypothesis Complexity
- PAC learning framework
- VC dimension

# Hypothesis class

Recall that a machine learning algorithm is a mapping to find a hypothesis to fit the data

$$\mathcal{A} : S \in (\mathcal{X} \times \mathcal{Y})^n \mapsto h_S \in H.$$

Here  $H$  is the predefined hypothesis class.

The mapping is an optimisation procedure that picks a hypothesis from the predefined hypothesis class to minimise or maximise the objective.

$$\arg \min_{h \in H} R_S(h).$$

# Hypothesis class example

**Example:** build a model to find the best apple in a basket

**Objective function:** the sweetest apple is the best

**Training data:** I have 200 baskets, 100 for training, 100 for testing

You might define a set of rules (the **hypothesis class**):

- **Color rule:** the reddest apple is the sweetest.
- **Size rule:** the largest apple is the sweetest.
- **Weight rule:** the heaviest apple is the sweetest.
- **Combined rule:** the apple that is both red, large, and heavy is the sweetest.

Each individual rule is a **hypothesis**.

The collection of these rules is the **hypothesis class**.

# Notation: risks

- Empirical risk

$$R_S(h) = \frac{1}{n} \sum_{i=1}^n \ell(X_i, Y_i, h)$$

where  $S = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$  is the training sample.

- Expected risk

$$R(h) = \mathbb{E}[R_S(h)] = \mathbb{E}[\ell(X, Y, h)]$$

# Hypothesis class example

- Empirical risk:

the error rate of the 100 baskets

- Expected risk:

the error rate of the 200 baskets

(the 100 training baskets are observable; however, the  
100 testing baskets are unknown)

**200 baskets, 100 for training, 100 for testing**

# Notation

- The best hypothesis in the universal function space (target concept):

$$c = \arg \min_h R(h).$$

- The optimal (best) hypothesis in the predefined hypothesis class:

$$h^* = \arg \min_{h \in H} R(h).$$

- The hypothesis we can learn from data:

$$h_S = \arg \min_{h \in H} R_S(h) = \arg \min_{h \in H} \frac{1}{n} \sum_{i=1}^n \ell(X_i, Y_i, h).$$

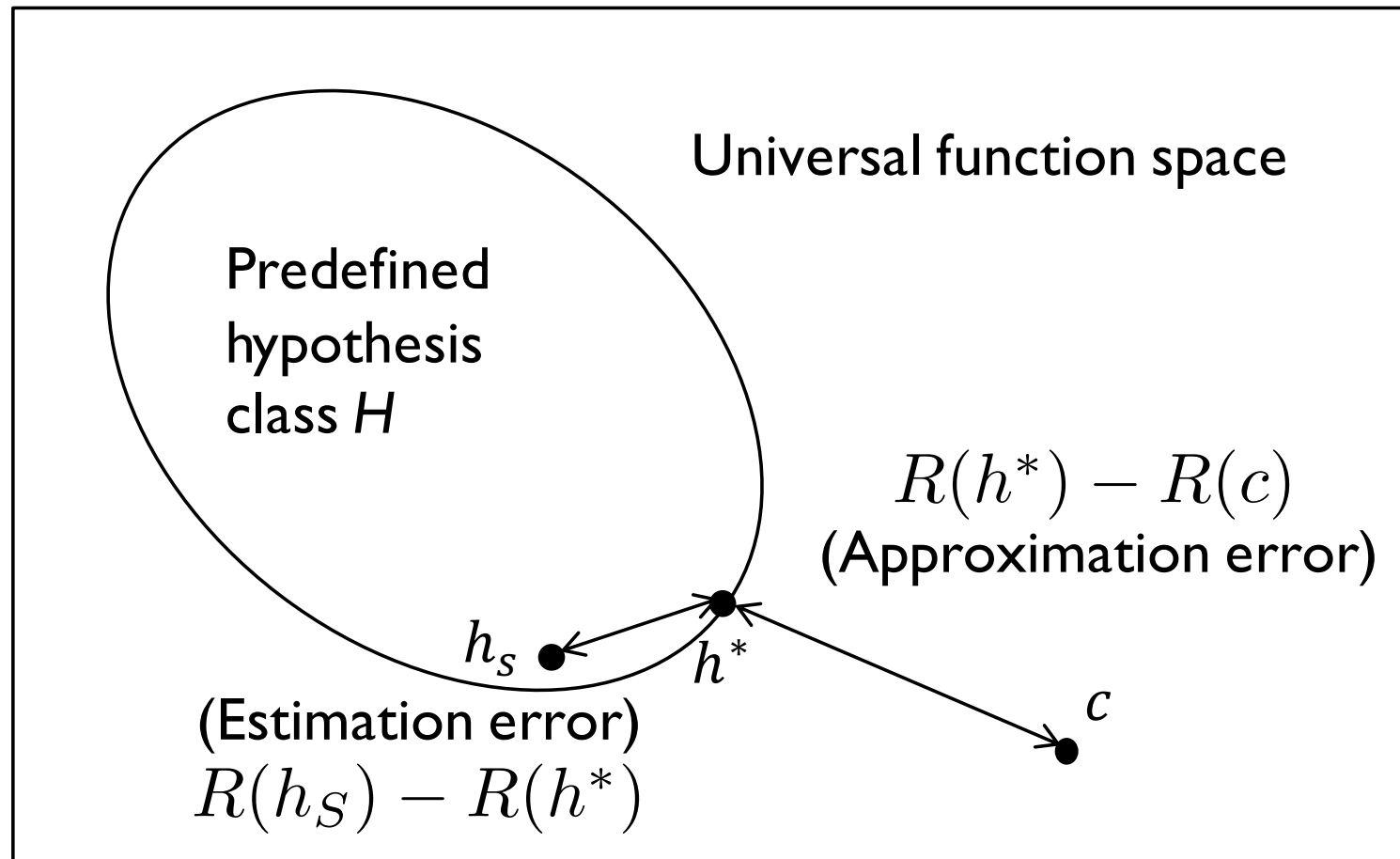


# Hypothesis

- $H = \{color\ rule, size\ rule, weight\ rule, combined\ rule\}$
- $c$  the ideal best hypothesis of universal function (may not in the predefined hypothesis class). (**shape rule** is the most accurate on the 200 baskets, but we didn't include it in our  $H$ )
- $h^*$  the ideal best hypothesis of the predefined hypothesis class. (**color rule** is the most accurate on the 200 baskets within  $H$ )
- $h_S$  the best hypothesis of the predefined hypothesis based on the training data. (**weight rule** is the most accurate on the training 100 baskets)

# Notation

What are the differences between  $c$ ,  $h^*$ , and  $h_S$ ?



- Approximation error is caused by the difference between  $h^*$  and  $c$
- Estimation error is caused by the difference between  $h_S$  and  $h^*$

# Hypothesis class

If the target  $c$  is within the predefined hypothesis class  $H$ , the approximation error will be zero.

It seems we should choose a large enough predefined hypothesis class to contain the target  $c$ . Does this help?

Trade-off

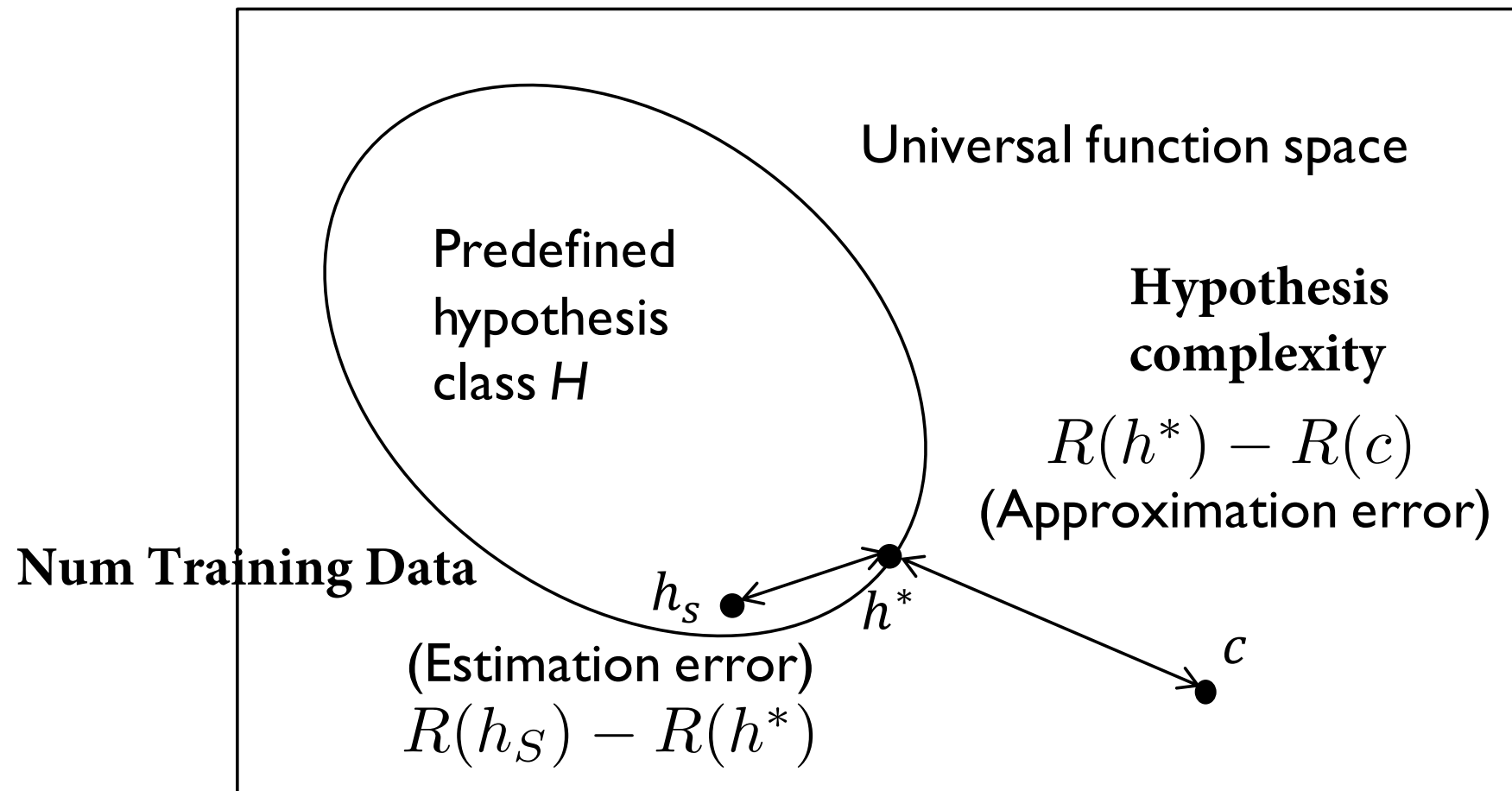
Large and complex hypothesis class would make it hard to learn.

The estimation error will become large!

To explain this, we need to introduce the PAC learning framework!

# Notation

What are the differences between  $c$ ,  $h^*$ , and  $h_S$ ?



- Approximation error is caused by the difference between  $h^*$  and  $c$
- Estimation error is caused by the difference between  $h_S$  and  $h^*$

# PAC learning framework

Probably approximately correct learning (PAC learning) is a framework for mathematical analysis of machine learning. It was proposed in 1984 by Leslie Valiant.

The PAC learning framework explains how many training examples are needed to learn the best hypothesis in the predefined class.

# PAC learning framework


## Definition:

A hypothesis class  $H$  is said to be PAC (probably approximately correct)-learnable if there exists a learning algorithm  $\mathcal{A}$  and a polynomial function  $\text{poly}(\cdot, \cdot)$  such that for any  $\epsilon > 0$  and  $\delta > 0$ , for all distribution  $D$  on  $X \times Y$ , the following holds for any sample of size  $n > \text{poly}(1/\delta, 1/\epsilon)$  and the hypothesis  $h_S$  learned by  $\mathcal{A}$ :

$$p \left\{ R(h_S) - \min_{h \in H} R(h) \leq \epsilon \right\} \geq 1 - \delta.$$

# PAC learning framework

learned hypothesis                      approximately                      probably


$$p \left\{ R(h_S) - \min_{h \in H} R(h) \leq \epsilon \right\} \geq 1 - \delta.$$

If the training sample size is large enough, e.g.,  $n > \text{poly}(1/\delta, 1/\epsilon)$  with a high probability, the learned hypothesis  $h_S$  can be an approximation of the best one in the predefined hypothesis class for any task.

# PAC learning checking

To check if a given hypothesis class  $H$  is PAC learnable, we need to find a learning algorithm  $\mathcal{A}$  and a polynomial function  $poly(\cdot, \cdot)$  such that for any  $\epsilon > 0$  and  $\delta > 0$ , for all distribution  $D$  on  $X \times Y$ , the following holds for any sample of size  $n > poly(1/\delta, 1/\epsilon)$  and hypothesis  $h_S$  learned by  $\mathcal{A}$ :

$$p \left\{ R(h_S) - \min_{h \in H} R(h) \leq \epsilon \right\} \geq 1 - \delta.$$



# PAC learning checking

If the hypothesis class is of finite hypotheses, it is PAC learnable. Because

$$p \left\{ R(h_S) - \min_{h \in H} R(h) \leq 2M \sqrt{\frac{\log|H| + \log(2/\delta)}{2n}} \right\} \geq 1 - \delta.$$

We can find that if the hypothesis class  $H$  is large, to find a good hypothesis with a small prediction error, we need a large training sample size  $n$ .

# PAC learning proof

Proof sketch:

Hoeffding's inequality

$$1. p\{|R(h) - R_S(h)| \geq \epsilon\} \leq 2\exp\left(\frac{-2n\epsilon^2}{M^2}\right);$$

Union bound

$$2. p\{\sup_{h \in H} |R(h) - R_S(h)| \geq \epsilon\} \leq 2|H| \exp\left(\frac{-2n\epsilon^2}{M^2}\right);$$

$$p\{\sup_{h \in H} |R(h) - R_S(h)| \leq \epsilon\} \geq 1 - 2|H| \exp\left(\frac{-2n\epsilon^2}{M^2}\right);$$

Uniform Convergence Lemma

$$3. R(h_S) - \min_{h \in H} R(h) \leq 2\sup_{h \in H} |R(h) - R_S(h)|.$$

$$p\left\{R(h_S) - \min_{h \in H} R(h) \leq 2\sup_{h \in H} |R(h) - R_S(h)| \leq 2\epsilon\right\} \geq 1 - 2|H| \exp\left(\frac{-2n\epsilon^2}{M^2}\right);$$

$$\text{Let } \delta = 2|H| \exp\left(\frac{-2n\epsilon^2}{M^2}\right).$$

$$p\left\{R(h_S) - \min_{h \in H} R(h) \leq 2M \sqrt{\frac{\log|H| + \log(2/\delta)}{2n}}\right\} \geq 1 - \delta;$$

# PAC learning proof

Proof sketch:

$$\text{Let } \delta = 2|H| \exp\left(\frac{-2n\epsilon^2}{M^2}\right).$$

$$p\left\{R(h_S) - \min_{h \in H} R(h) \leq 2M \sqrt{\frac{\log|H| + \log(2/\delta)}{2n}}\right\} \geq 1 - \delta;$$

$$p\left\{R(h_S) - \min_{h \in H} R(h) \leq \epsilon\right\} \geq 1 - \delta;$$

$$\text{Let } \epsilon = 2M \sqrt{\frac{\log|H| + \log(2/\delta)}{2n}}.$$

$$n = \frac{2M^2}{\epsilon^2} \log\left(\frac{2|H|}{\delta}\right), n > \text{poly}\left(\frac{1}{\epsilon}, \frac{1}{\delta}\right).$$

# PAC-learnable

## Intuition in plain English

Imagine we want to learn to identify cats in photos:

- We don't need 100% correct identification; 98% is fine ( $\epsilon=0.02$ ).
- We don't want a method that sometimes fails completely; we want it to succeed almost all the time, say 95% ( $\delta=0.05$ ).
- PAC-learning says: "If we have  $n > p_{\text{loy}}(\frac{1}{\delta}, \frac{1}{\epsilon})$  photos, we can train a model that meets these goals."

# VC dimension

If the predefined hypothesis class  $H$  has infinite many hypotheses, how can we upper bound

$$\sup_{h \in H} |R_S(h) - R(h)|?$$

Hint: we consider binary classifier  $s$  and group the hypothesis

$$H = \{(h_1^1, \dots, h_{n_1}^1), (h_1^2, \dots, h_{n_2}^2), \dots, (h_1^G, \dots, h_{n_G}^G)\}.$$

# VC dimension

$$H = \{(h_1^1, \dots, h_{n_1}^1), (h_1^2, \dots, h_{n_2}^2), \dots, (h_1^G, \dots, h_{n_G}^G)\}.$$

Although the predefined hypothesis class  $H$  has infinitely many hypotheses, we can group them into **finite groups**, where the hypotheses in each group having the same value of

$$h(X_1), h(X_2), \dots, h(X_n)$$

Let  $h^1, \dots, h^G$  be the representatives of each group, we have a new set of representatives:

$$H' = \{h^1, \dots, h^G\}.$$

# VC dimension

How to find  $H'$  ?

## **Definition:**

*Growth function*

The growth function  $\Pi_H : \mathbb{N} \rightarrow \mathbb{N}$  for a hypothesis class  $H$  is defined by

$$\forall n \in \mathbb{N}, \Pi_H(n) = \max_{X_1, \dots, X_n} |\{h(X_1), \dots, h(X_n) : h \in H\}|$$

The maximum group that have the same predictions.

# VC dimension

## **Definition:**

### *Shattering*

The data points  $\{X_1, \dots, X_n\}$  is said to be shattered by a hypothesis class  $H$  when  $H$  realises all possible binary predictions. That is  $\Pi_H(n) = 2^n$ .



# VC dimension

## **Definition:**

*VC dimension*

The VC dimension of a hypothesis class  $H$  is the size of the largest set that can be fully shattered by  $H$ :

$$\text{VC dimension}(H) = \max_n \{n : \Pi_H(n) = 2^n\}.$$

# VC dimension

Let  $H$  be a hypothesis set with VC dimension( $H$ ) =  $d$  then  
for all  $n \geq d$

$$\Pi_H(n) \leq \left(\frac{en}{d}\right)^d.$$

The proof is in Chapter 3 of the book “Foundations of ML”

# PAC learning checking

If the hypothesis class is of finite VC dimension, it is PAC learnable. Because

$$p \left\{ R(h_S) - \min_{h \in H} R(h) \leq 2 \sup_{h \in H} |R(h) - R_S(h)| \leq 2M \sqrt{\frac{32(d \log(en/d) + \log(8/\delta))}{n}} \right\} \geq 1 - \delta.$$

Since  $\delta = 8 \left( \frac{en}{d} \right)^d \exp(-n\epsilon^2/32M^2)$ , we have

$$n = \frac{32M^2}{\epsilon^2} (d \log(en/d) + \log(8/\delta)).$$

$$n > \text{poly}(1/\delta, 1/\epsilon)$$

# Key points

- Hypothesis Complexity
- PAC learning framework
- VC dimension