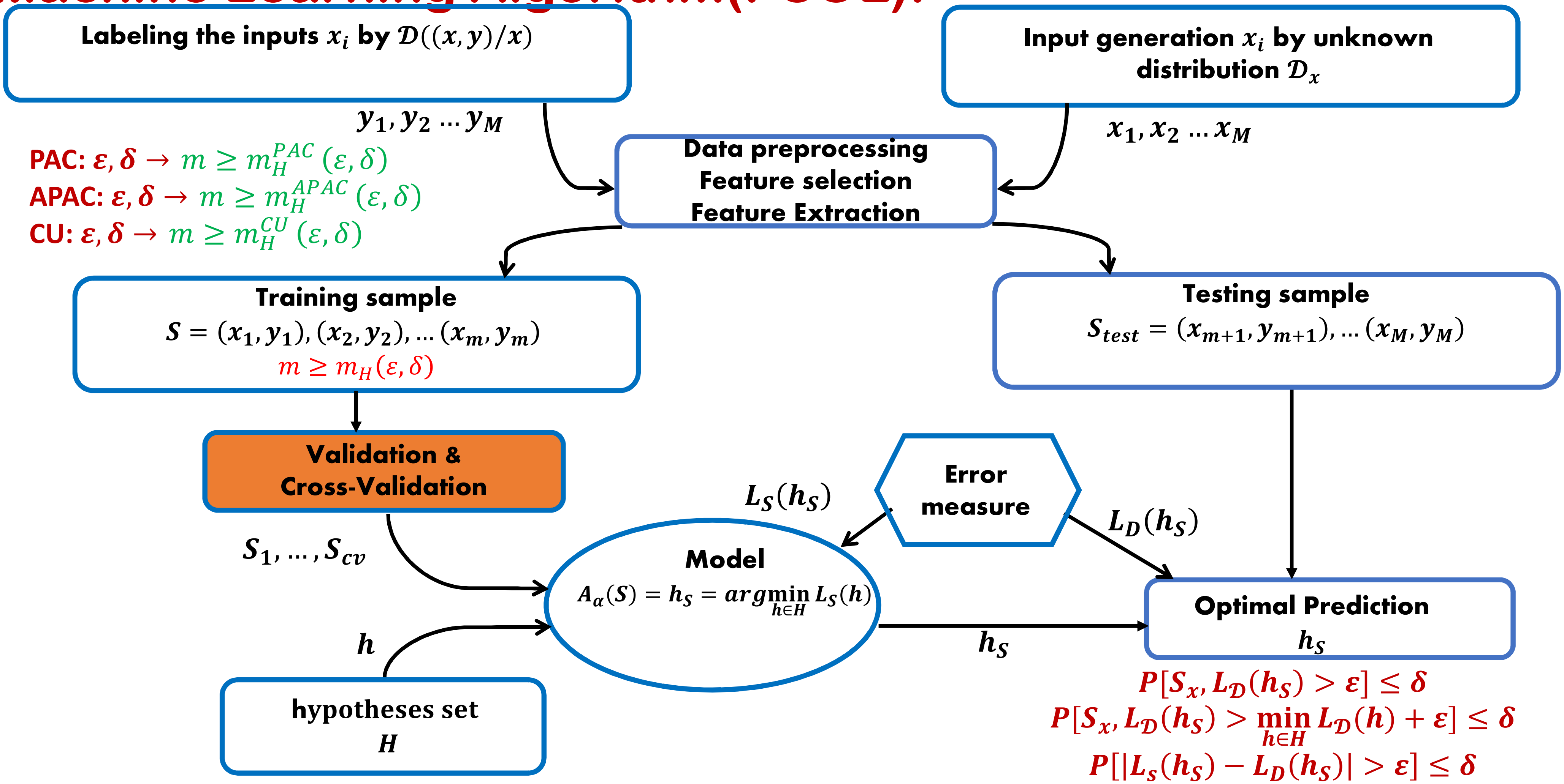
Part 3: Overfitting-Underfitting

1. **Validation/Cross-Validation:**
    1. **Validation set**
    2. **Model selection**
    3. **Cross-validation: k-fold method**
2. **Regularization**

Machine Learning Algorithm(PSOL):



Reminder

Definition: APAC learning model

H follows agnostic PAC learning, if there exist $m_H\colon\ (0,1)^2\ \rightarrow \mathbb{N}$ and A_α.

Having the following property: $\forall \varepsilon, \delta \in (0,1), \forall\ \mathcal{D}$ on $X \times Y$.

Then, if we run A_α on $m \geq m_H(\varepsilon, \delta)$ generated $(i.i.d.)$ such that S is selected with a probability at least $(1-\delta)$, A_α will generate the hypothesis h_S such that:

$$P_{S \sim \mathcal{D}^m}\left[L_{\mathcal{D}}(h_S) \leq \min_{h \in H} L_{\mathcal{D}}(h) + \varepsilon\right] \geq 1 - \delta \quad \forall\, m \geq m_H(\varepsilon, \delta)$$

In other words:

$$P_{S \sim \mathcal{D}^m}\left[L_{\mathcal{D}}(h_S) > \min_{h \in H} L_{\mathcal{D}}(h) + \varepsilon\right] \leq \delta \ \ \forall\ \ m \geq m_H(\varepsilon, \delta)$$

Motivation

- If $L_S(\boldsymbol{h_S})$ is too big, we have the Underfitting problem. So, we should variate the model's configuration
- Variate the whole model, and select the best one that have the smaller $L_D(\boldsymbol{h_S})$.

Objective:

- How to select the best Model/Configuration ?

Tools:

- Validation (sufficient training data points)
- Cross-validation (unsufficient training data points).

Validation & Cross-Validation

1. Validation Set

Definition : Validation

The validation consists in extracting from the training set another set named: validation set. This is for two objectives :

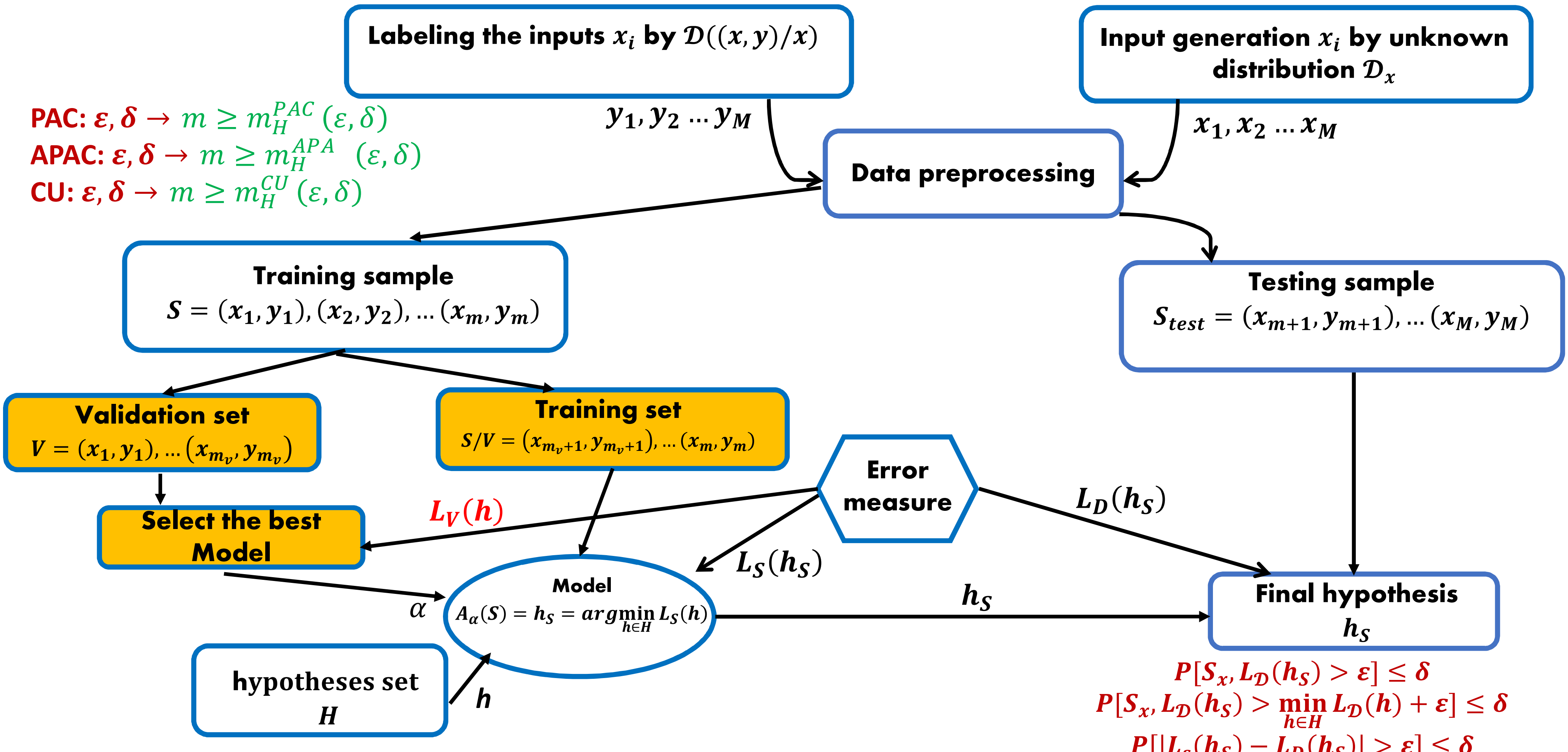
- Select the best model (algorithm or hyper-parameter).
- Better estimation of the generalization error.

Let's consider the following validation set :

$$V = (x_1, y_1), \dots, (x_{m_v} y_{m_v})$$

Whose data points are sampled according to the distribution D independently from the m data of the training set.

Machine Learning Algorithm(PSOL):Validation set



1. Validation Set

Theorem :

Consider the hypothesis h and the cost function belonging to $[0,1]$. So $\forall\delta \in [0,1]$:

$$P_{V\sim D^{m_v}}\left[|L_V(h) - L_D(h)| \leq \varepsilon = \sqrt{\frac{\log\left(\frac{2}{\delta}\right)}{2m_v}}\right] \geq 1-\delta$$

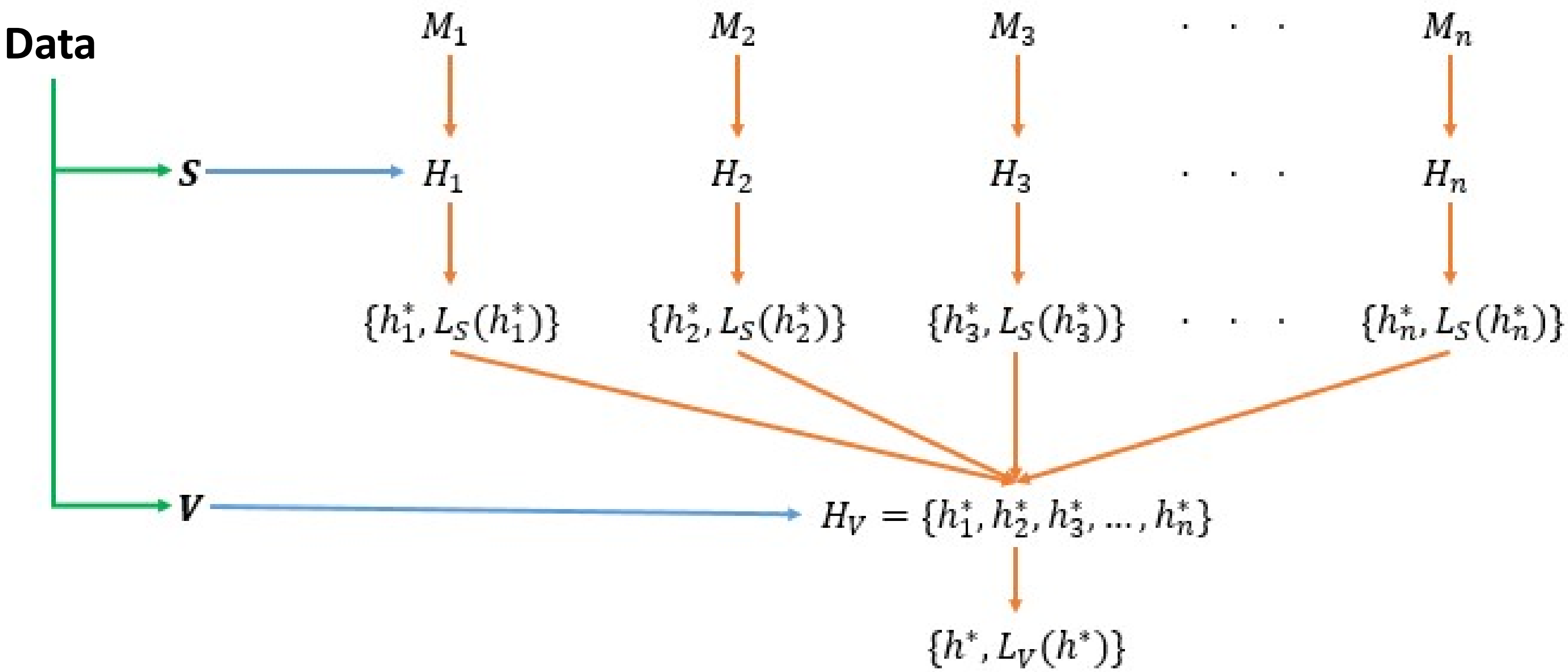
Notice : $|\boldsymbol{H}| \cong \infty$

- This generalization bound doesn't depend neither on the learning algorithm nor on the training set.
- The generalization bound of the validation set is better than that of the training set:

$$\boldsymbol{L_D(h) - L_S(h) \leq \sqrt{C\frac{d_{VC}(H) + log(\frac{1}{\delta})}{m}}}$$

With: C is a constant.

2. Model Selection



$$Model = \{learning\ algorithm; hyperparameters\}$$

If $h^* \in H_i$ so M_i is the best model, with $i = 1, \dots, n$.

2. Model Selection

Theorem : $|H_V| < \infty$

Let's consider the hypothesis set $H_V = \{h_1^*, h_2^*, \dots, h_n^*\}$ and the cost function belonging to $[0,1]$. Let's consider the validation set V of size m_v sampled independently from H_V. So $\forall \delta \in [0,1]$ and $\forall h^* \in H_V$:

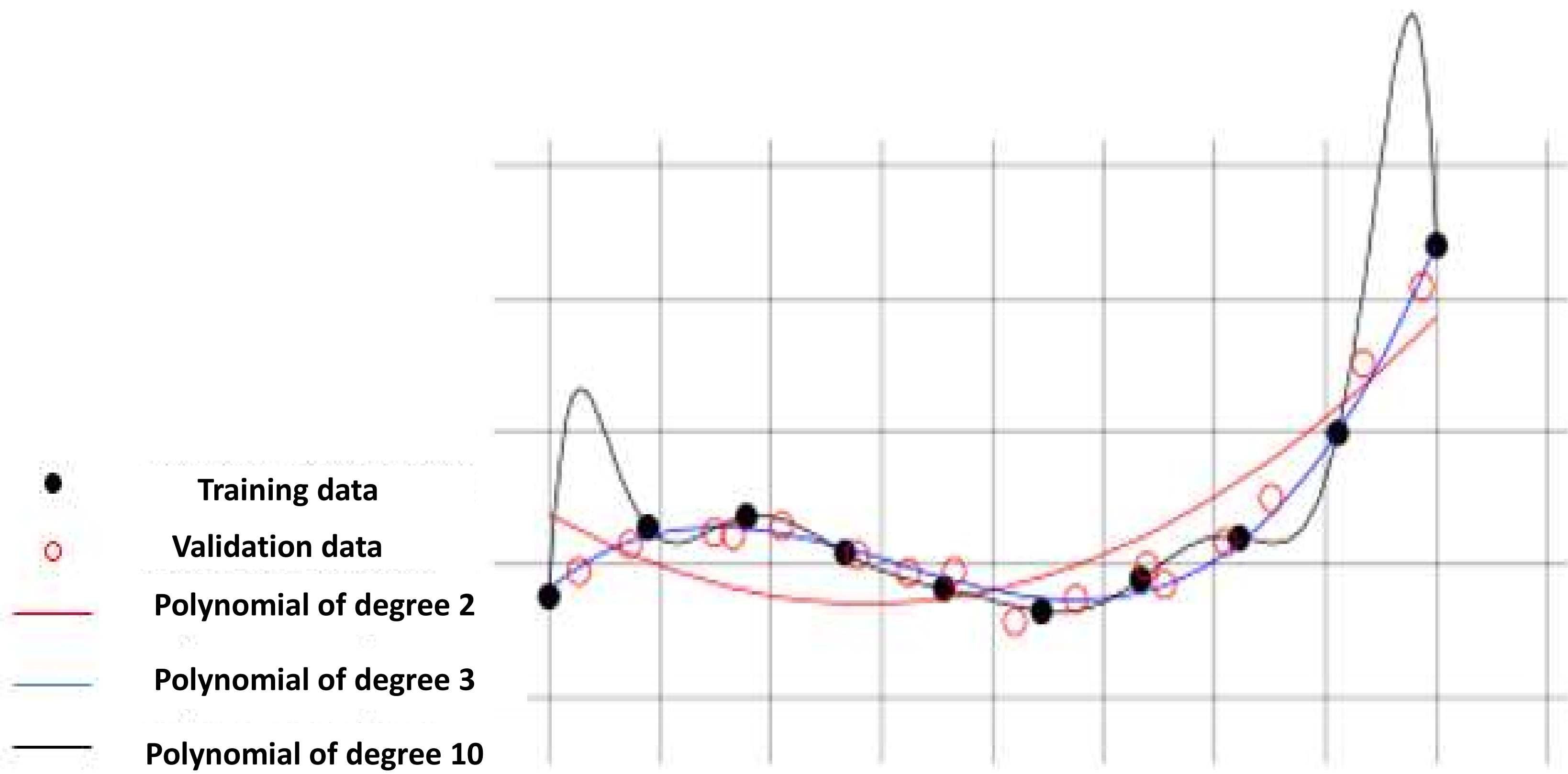
$$P_{V \sim D^{m_v}}\left[|L_V(h^*) - L_D(h^*)| \leq \varepsilon = \sqrt{\frac{\log\left(\frac{2|H_V|}{\delta}\right)}{2m_v}} \right] \geq 1 - \delta$$

Notice :

The generalization bound based on the validation set is better than that of the training set under the condition that the size of H_V is small.

2. Model Selection

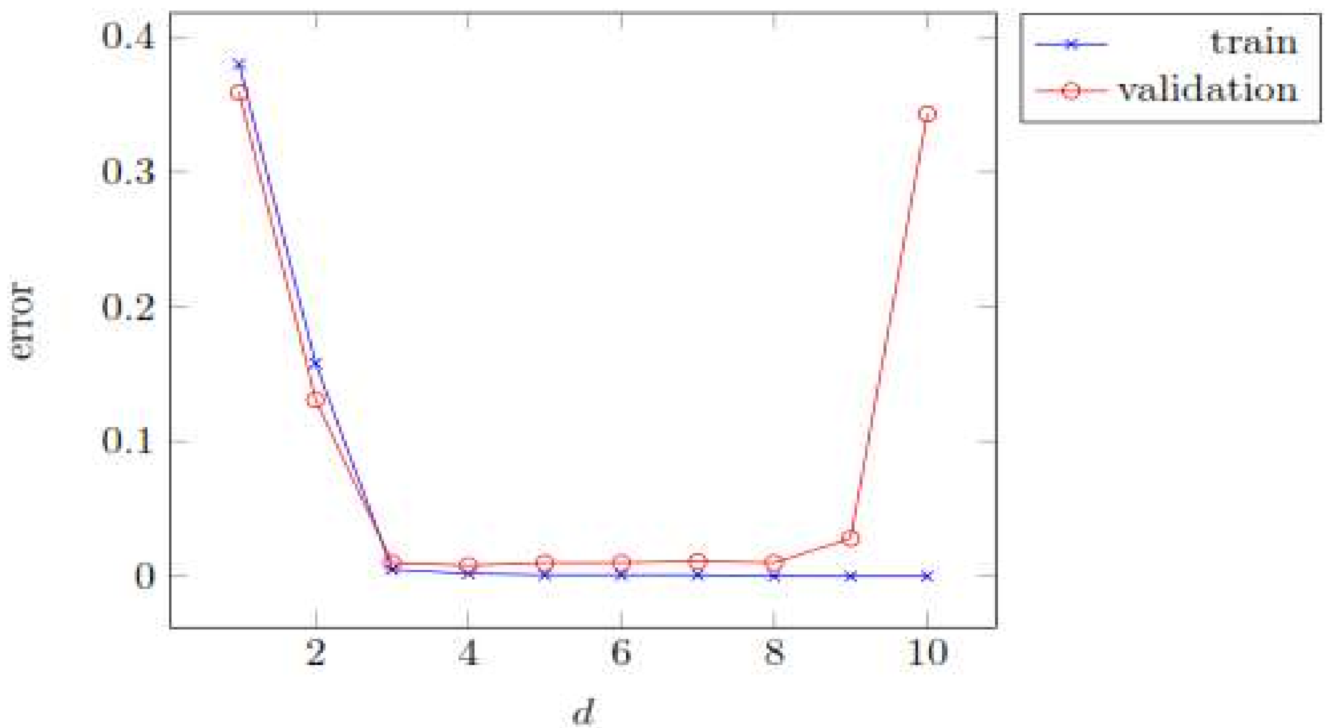
What is the best polynomial?



2. Model Selection

Definition :

The model selection curve presents the training error and the validation error in function of the model complexity.



3. Cross-validation: k-fold method

Definition : k-Fold cross-validation

The training set is partitioned on k subsets (folds) of size $\frac{m_H(\varepsilon,\delta)=m}{k}$.

For each subset S_i $i = 1,..,k$, the algorithm A_α is trained on the union of the remaining subsets, then the estimation of the validation error of $A_\alpha(h)$ is made on S_i.

Finally, the total validation error of the model is the mean of all validation errors of the subsets S_i.

Definition : Leave-one-out cross-validation

Leave-one-out cross-validation is a particular case of the $\boldsymbol{k}$-Fold cross-validation with $k = 1$.

3. Cross-validation: k-fold method

ALGORITHM : « k-Fold Cross Validation »

INPUT : Training set : $\mathrm{S} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$

Parameter values set : Θ

Learning Model A

Integer k

BEGIN:

Partition S into $S_1, \ldots, S_k$

FOREACH $\theta \in \Theta$

FOR $i = 1 \ldots k$

$h_{i,\theta} = A(S \backslash \mathrm{S}_i; \theta)$

Compute the error estimation $L_{S_i}(h_{i,\theta})$

ENDFOR

error(θ) = $\frac{1}{k} \sum_{i=1}^{k} L_{S_i}(h_{i,\theta})$

END

OUTPUT : Optimal parameter : $\theta^* = \mathrm{argmin}_{\theta \in \Theta}(\mathrm{error}(\theta))$

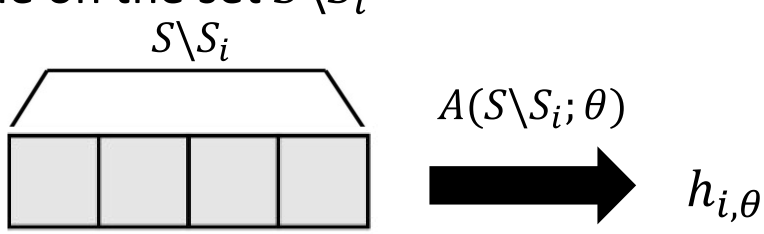
Optimal hypothesis : $h_{\theta^*} = A(S; \theta^*)$

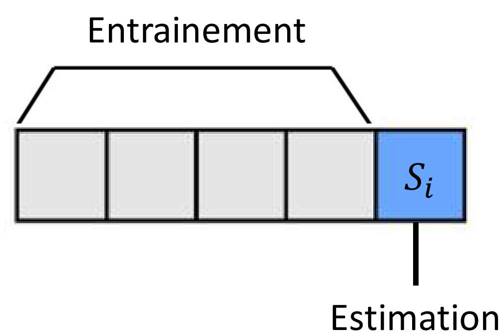
3. Cross-validation: k-fold method

Example: $k = 5$

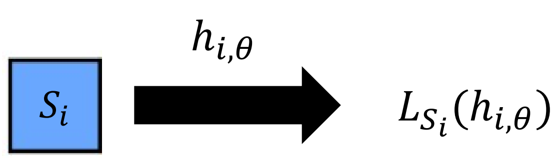
- Partition S into 5 subsets $S_1, \dots, S_5$

- For each $i = 1, \dots, 5$:
  - The training is done on the set $S \backslash S_i$



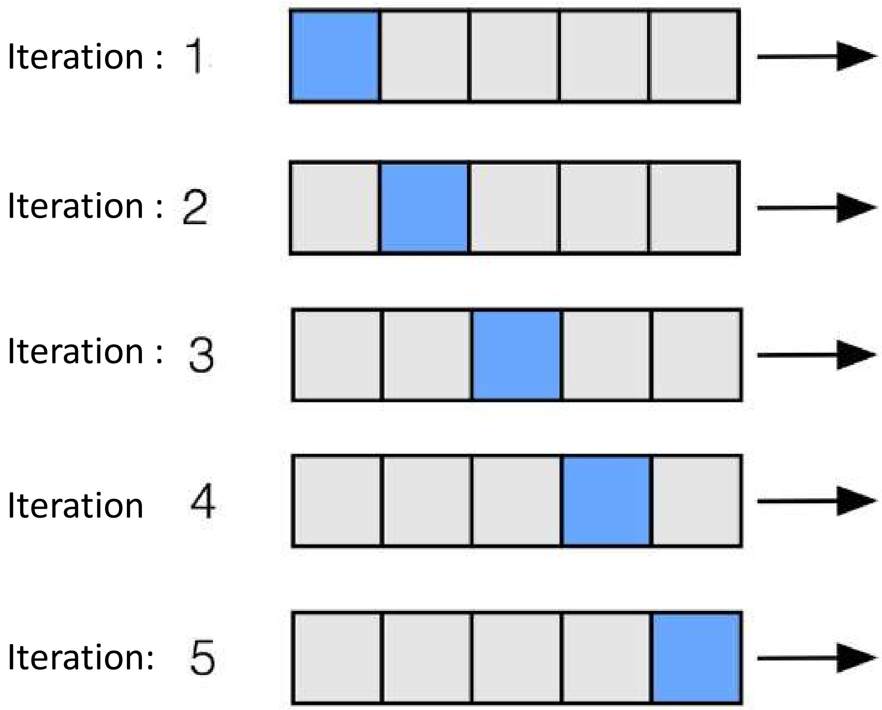


  - The error estimation $L_{S_i}(h_{i,\theta})$ is done on S_i :



3. Cross-validation: k-fold method

- For $\forall i = 1, \ldots, 5: m_{S_i} < m_H = m$



$L_{S_1}(h_{1,\theta})$

$L_{S_2}(h_{2,\theta})$

$L_{S_3}(h_{3,\theta})$

$L_{S_4}(h_{4,\theta})$

$L_{S_5}(h_{5,\theta})$

$$\text{error}(\theta) = \frac{1}{5}\sum_{i=1}^{5} L_{S_i}(h_{i,\theta})$$