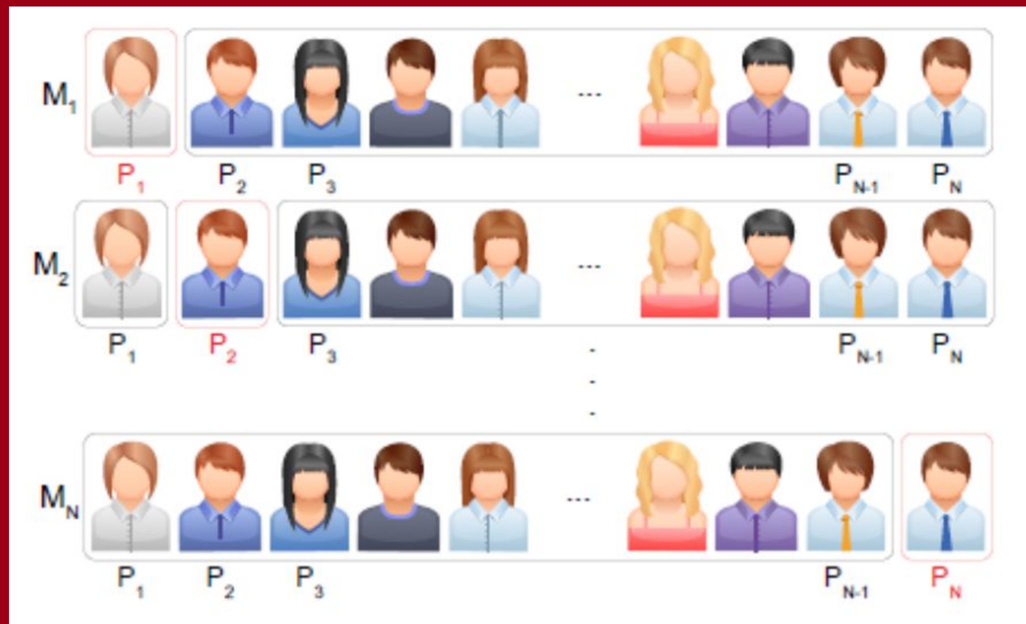
DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

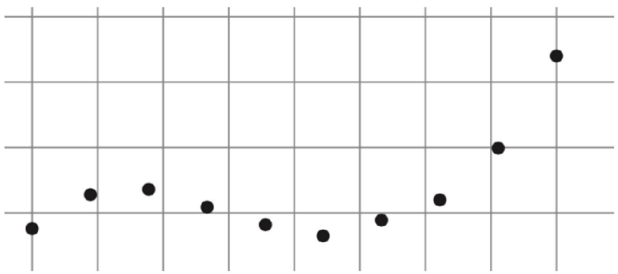


Model Selection and Validation

Machine Learning 2022-23
UML book chapter 11

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Choosing the Right Model



- There are different algorithms
- Algorithms have parameters/design choices

How to select the best algorithm or params?

Example→ *Hyp. class*: Polynomial Regression

Hyper-parameter: degree of polynomial

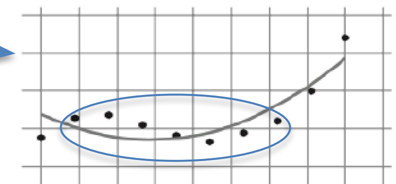
Two approaches on the book

- Structural Risk Minimization (SRM)

  Not part of the course (impractical)

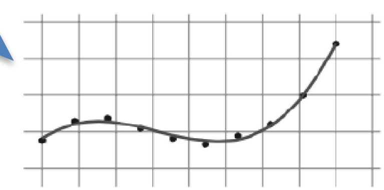
- ***Use a Validation set***

Degree 2



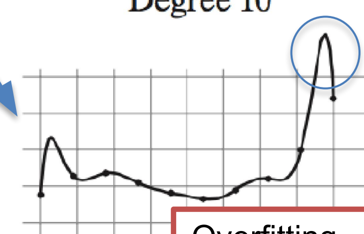
High Empirical Risk

Degree 3



Good Choice?

Degree 10



Overfitting

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Validation Set

Idea: divide the training set in 2 parts, use the first to pick an hypothesis, and the second (*not used to train*) to estimate its true error

Assume we have picked a predictor h (e.g., by ERM rule on $\mathcal{H}_d$)

- $V = \left((x_1, y_1), \dots, (x_{m_v}, y_{m_v})\right)$: set of m_v samples from D not used for training (*validation set*)
- L_V : loss computed on *V* (loss in [0,1])

Theorem :
For every $\delta \in (0,1)$, with probability $\geq 1 - \delta$ (over the choice of V), we have:

$$|L_D(h) - L_V(h)| \leq \sqrt{\frac{\log\left(\frac{2}{\delta}\right)}{2m_v}}$$

Idea of demonstration: similar to law of large numbers, with more samples average gets closer to expectation

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Bounds Comparison

$$L_D(h) \le L_S(h) + \sqrt{C\frac{d + \log(\frac{1}{\delta})}{m}}$$

*From quantitative version fundamental theorem statistical learning**

$$L_D(h) \le L_V(h) + \sqrt{\frac{\log(\frac{2}{\delta})}{2m_v}}$$

With validation set

The bound based on the validation set is more accurate:

- Depends on validation set size, not on the training set size
- Does not depend on VC-dimension
  - *Why ? → The validation samples have not been used for training*
- Choose final hypotheses by ERM over the validation set

$$*: m \le C\frac{d+\log\left(\frac{1}{\delta}\right)}{\epsilon^2} \rightarrow \epsilon^2 \le C\frac{d+\log\left(\frac{1}{\delta}\right)}{m} \rightarrow \epsilon \le \sqrt{C\frac{d+\log\left(\frac{1}{\delta}\right)}{m}}$$

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Validation for Model Selection (1)

Train different algorithms or the same algorithm with different hyper-parameters on the training set

1. For each algorithm or parameter set there is a different hypothesis class $\mathcal{H}_i = \{h_{i1}, h_{i2}, \dots, h_{iI}\}$ where $\mathrm{I} = |\mathcal{H}_i|$
2. Train the ML algorithm on each hypothesis class independently, call h_i^{ERM} the found ERM solution
3. Collect all the ERM solutions h_i^{ERM} into a new hypothesis class $\mathcal{H}'$
$$\mathcal{H}' = \{h_1^{ERM}, h_2^{ERM}, \dots, h_r^{ERM}\}$$
4. Select inside $\mathcal{H}'$ as final output the predictor h^* that minimizes the error on the validation set

Validation for Model Selection

- Train different algorithms or the same algorithm with different hyper-parameters on the training set obtaining a set of ERM predictors $\mathcal{H}' = \{h_1^{ERM}, h_2^{ERM}, \dots, h_r^{ERM}\}$
- Choose the predictor h^* that minimizes the error on the validation set
- $\mathcal{H}'$ Similar to a finite hypothesis class where $\mathcal{H}$ is not fixed ahead but depends on training set
- *Theorem message:* the validation error is a good approximation of the true error if we do not try too many methods (otherwise going back to the "standard" case and there is risk of overfitting)

Let $\mathcal{H}' = \{h_1^{ERM}, \dots, h_r^{ERM}\}$ be an arbitrary set of predictors and assume that the loss is in [0,1]. Assume that a validation set V of size m_v is sampled independent of $\mathcal{H}'$. Then, with probability at least $1-\delta$ over the choice of V we have:

$$\forall h \in \mathcal{H}': |L_D(h^*) - L_V(h^*)| \leq \sqrt{\frac{\log\left(\frac{2|\mathcal{H}'|}{\delta}\right)}{2m_v}}$$

Size of output predictor set
(*not of the hypothesis classes used for training*)

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Demonstration

1. Apply previous theorem to each h_i^{ERM}: $\forall h_i^{ERM}$: $P_{bad-valid} \leq \delta$ (*)
2. Repeat for $|\mathcal{H}'|$ times: from union bound (**)

$$P_{bad-valid-all} \leq \sum_{|\mathcal{H}'|} \delta = |\mathcal{H}'|\delta$$

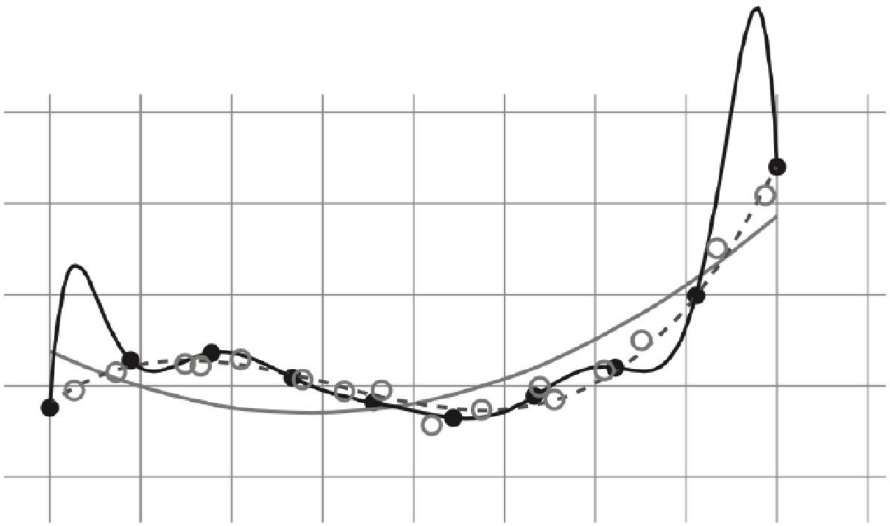
3. To have $P_{bad-valid-all} \leq \delta_{all}$ set $\delta' = \frac{\delta}{|\mathcal{H}'|} \rightarrow \delta_{all} = \sum_{|\mathcal{H}'|} \delta' = |\mathcal{H}'| \frac{\delta}{|\mathcal{H}'|}$

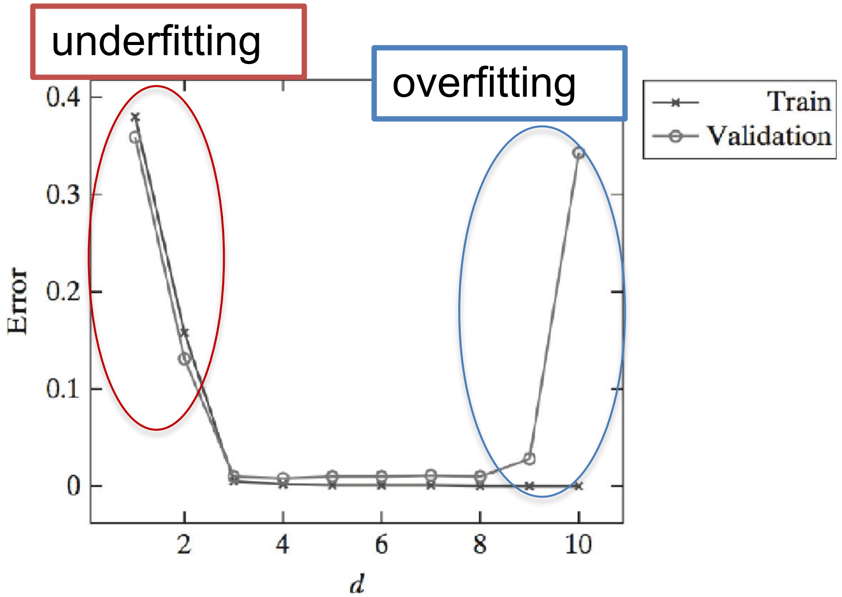
$$\forall h \in H: |L_D(h^*) - L_V(h^*)| \leq \sqrt{\frac{\log\left(\frac{2}{\delta'}\right)}{2m_V}} = \sqrt{\frac{\log\left(\frac{2|\mathcal{H}'|}{\delta}\right)}{2m_V}}$$

(*) $P_{bad-valid} = P\left(|L_D(h) - L_V(h)| > \sqrt{\frac{\log\left(\frac{2}{\delta}\right)}{2m_v}}\right)$

(**) Recall union bound: $P(\bigcup_i A_i) \leq \sum_i P(A_i)$

Example





- Empty circles: validation samples
- The fitting is done using only the training samples (full circles)
- Notice how high order polynomial does not fit well over the validation ones (specially on the left and right sides)
- The right plot is sometimes called «*model selection curve*»

Grid Search for Multiple Parameters

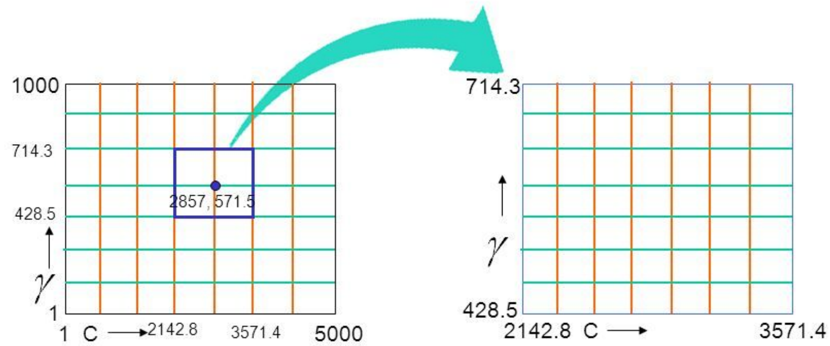


image from Tata research center

What if we have one or more parameters with values in $\mathbb{R}$?

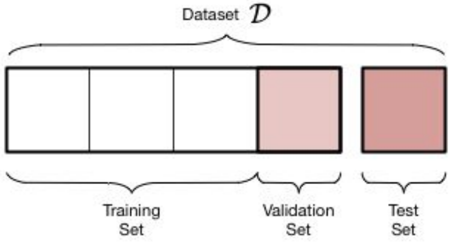
1. Start with a rough grid of values
2. Plot the corresponding model-selection curve
3. Based on the curve, zoom in to the correct region
4. Restart from step 1 with a finer grid

- The empirical risk on the validation set is not an estimate of the true risk, *in particular if we choose among many models* !
- Furthermore grid search does not always find the global optimum for the set of parameters, but it is a reasonable approximation
- Question: how can we estimate the true risk after model selection ?

Train, Validation and Test sets

We have to choose among multiple possible hypotheses set $\mathcal{H}_i$

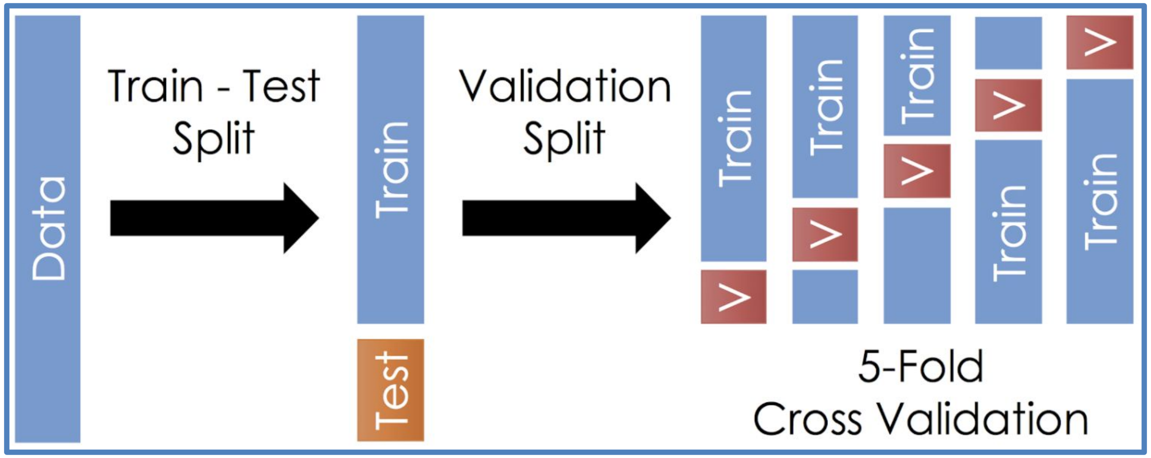
Approach → split the data in 3 parts:



1. *Training set*: used to learn the best model h_i^{ERM} inside each class $\mathcal{H}_i$
2. *Validation set*: used to pick one hypothesis h^* from $h_1, h_2, \ldots$.
3. *Test set*: used to estimate the true risk $L_D(h^*)$

- The estimate from the test set has the guarantees provided by the proposition on estimate of $L_D(h^*)$ for one class
- The test set is not involved in the choice of h^*
- if after using the test set to estimate $L_D(h^*)$ we decide to choose another hypothesis (because we have seen $L_D(h^*)$ ...) we cannot use the test set again to estimate $L_D(h^*)$!

k-Fold Cross Validation



When data is not plentiful, we cannot afford to drop part of it to build the validation set → use k-fold cross validation

k-fold cross validation:

1. Partition (training) set of *m* samples into *k* folds of size *m/k*
2. For each fold:
   - train on the union of the other folds
   - estimate error (for learned hypothesis) on the selected fold
3. Estimate of the true error as the average of the estimated errors

Example: Gesture Recognition

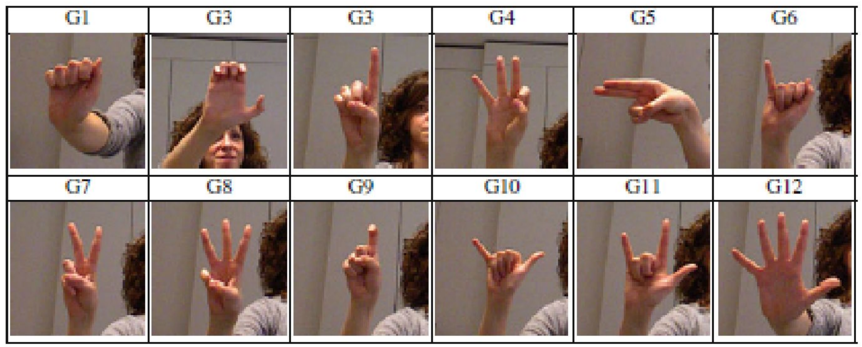
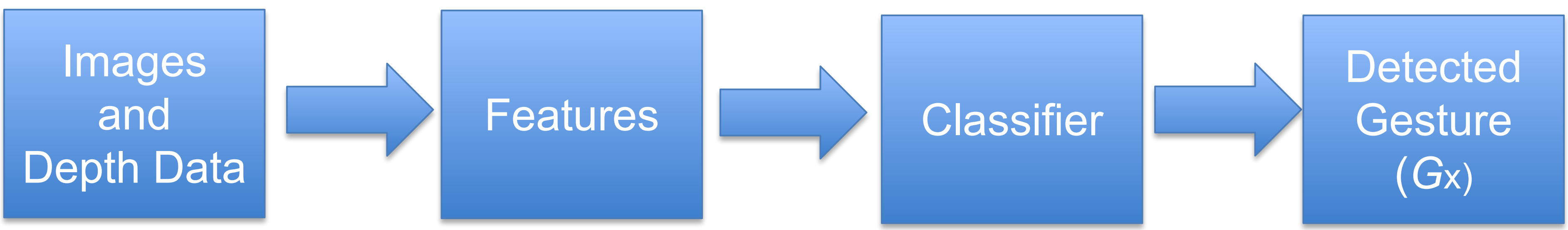
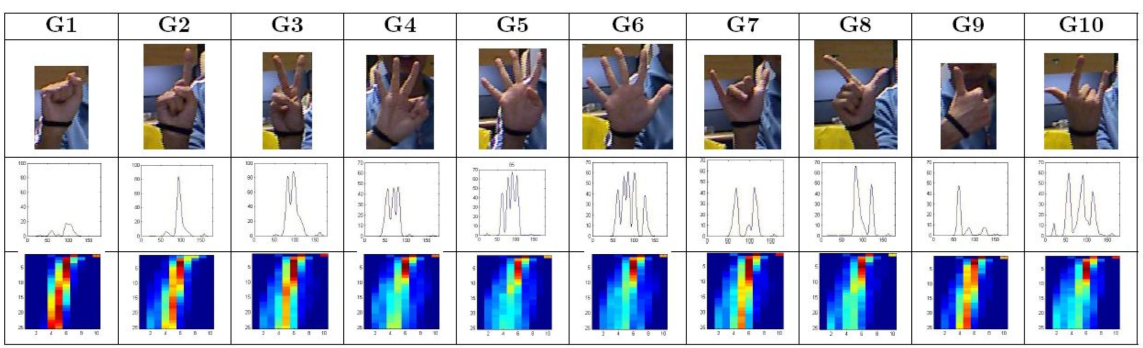


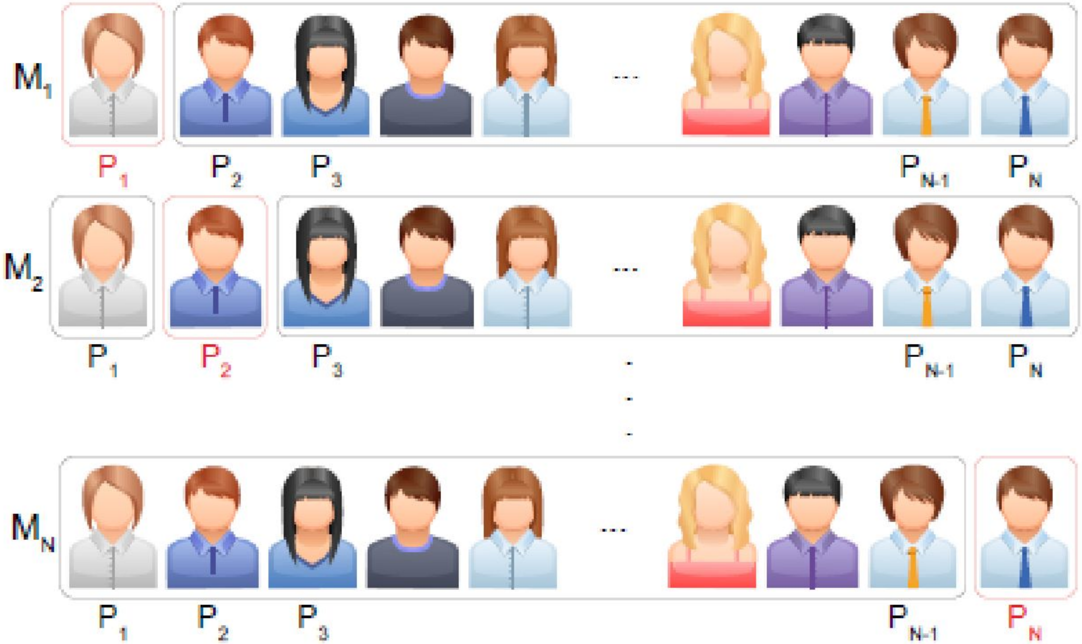
Fig. 11.9 Gestures from the American manual alphabet contained in the experimental dataset





DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Example: Gesture Recognition



- Dataset: 12 gestures, 14 people, 10 repetitions -> 1680 samples
- Low number of samples -> use *k-fold cross-validation*
- Maximize Training/Validation diversity (better generalization properties): in this case leave out all the examples from a single person (same person always does the gestures in a very similar way)

Model Selection with Cross Validation

k-Fold Cross Validation for Model Selection

input:
  training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
  set of parameter values Θ
  learning algorithm A
  integer k
partition S into $S_1, S_2, \ldots, S_k$
foreach $\theta \in \Theta$
  for $i = 1 \ldots k$
    $h_{i,\theta} = A(S \setminus S_i; \theta)$
  $\text{error}(\theta) = \frac{1}{k} \sum_{i=1}^{k} L_{S_i}(h_{i,\theta})$
output
  $\theta^\star = \text{argmin}_\theta \left[\text{error}(\theta)\right]$
  $h_{\theta^\star} = A(S; \theta^\star)$

remove i-th fold..

..and use it for testing

- Often cross validation is used for model selection
- In this case after selecting the model, the final hypothesis is obtained from training on the entire training set

Error Decomposition

Recall:

- $L_D(h^*)$ Approximation error (true error of best hypothesis in $\mathcal{H}$)
- $L_D(h_s) - L_D(h^*)$ Estimation error (difference between the true error of best hypothesis in $\mathcal{H}$ and true error of ERM solution)
- $L_S(h_s)$ Training error (empirical error of ERM solution on training set S)
- $L_V(h_s)$ Validation error (error on validation set V of ERM solution)

Decompose error:

- Approximation and estimation error (aready discussed)

$$L_D(h_s) = L_D(h^*) + (L_D(h_s) - L_D(h^*))$$

- Using train and validation errors

$$L_D(h_s) = \big(L_D(h_s) - L_V(h_s)\big) + \big(L_V(h_s) - L_S(h_s)\big) + L_S(h_s)$$

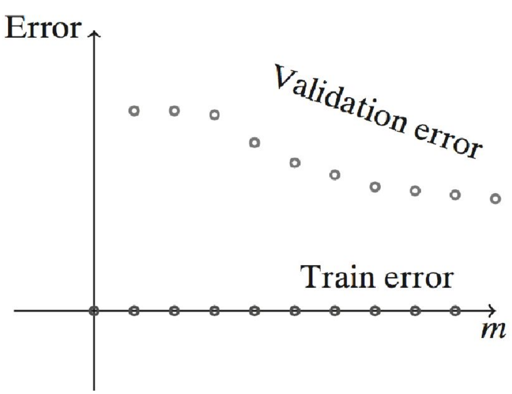
When Learning Fails…

Good training/validation errors… but results on test set are bad !

$L_S(h_S)$

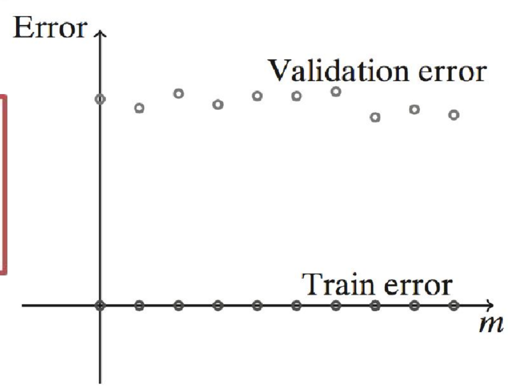
Large (underfitting) → Large approximation error $L_D(h^*)$ large

Small (overfitting) $L_D(h^*)$?



$\Rightarrow$ H may have a good approximation error but maybe we do not have enough data

$L_D(h^*)$?
estimate approximation error by comparing the 2 curves



$\Rightarrow$ There is no evidence that the approximation error of $\mathcal{H}$ is good

Summary

Some potential steps to follow if learning fails:

1. If you have tuned parameters, plot model-selection curve to make sure they are tuned appropriately
2. If training error is excessively large consider:
   - enlarge hypothesis class $\mathcal{H}$
   - change hypothesis class $\mathcal{H}$
   - change feature representation of the data
3. If training error is small, use learning curves to understand whether problem is approximation error or estimation error

   - if validation error seems to decrease (the error is large but the two curves get closer) :
     - get more data (if possible)
     - otherwise reduce complexity of $\mathcal{H}$
   - if the validation error remains large:
     - change $\mathcal{H}$
     - change feature representation of the data