An Introduction to Statistical Machine Learning - Theoretical Aspects -

Samy Bengio

bengio@idiap.ch

Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP)

CP 592, rue du Simplon 4

1920 Martigny, Switzerland

http://www.idiap.ch/~bengio

Statistical Learning Theory

1. The Data
2. The Function Space
3. The Loss Function
4. The Risk and the Empirical Risk
5. The Training Error
6. The Capacity
7. The Bias-Variance Dilemma
8. Regularization
9. Estimation of the Risk
10. Model Selection
11. Methodology

The Data

- Available training data:
  - $D_n = \{z_1, z_2, \cdots, z_n\} \in \mathcal{Z}$,
  - independently and identically distributed (iid),
  - drawn from unknown distribution $p(Z)$
- Various forms of the data:
  - Classification: $Z = (X, Y) \in \mathbb{R}^d \times \{-1, 1\}$

    objective: given a new x, estimate $P(Y|X = x)$
  - Regression: $Z = (X, Y) \in \mathbb{R}^d \times \mathbb{R}$

    objective: given a new x, estimate $E[Y|X = x]$
  - Density estimation: $Z \in \mathbb{R}^d$

    objective: given a new z, estimate $p(z)$

The Function Space

- Learning: search for a good function in a function space $\mathcal{F}$
- Examples of functions $f(\cdot;\theta) \in \mathcal{F}$:
  - Regression:

$$\hat{y} = f(x; a, b, c) = a \cdot x^2 + b \cdot x + c$$

  - Classification:

$$\hat{y} = f(x; a, b, c) = \text{sign}(a \cdot x^2 + b \cdot x + c)$$

  - Density estimation

$$\hat{p}(z) = f(z; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{|z|}{2}}\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(z-\mu)^T\Sigma^{-1}(z-\mu)\right)$$

The Loss Function

- Learning: search for a good function in a function space $\mathcal{F}$
- Examples of loss functions $L : \mathcal{Z} \times \mathcal{F}$
  - Regression:

$$L(z, f) = L((x, y), f) = (f(x) - y)^2$$

  - Classification:

$$L(z, f) = L((x, y), f) = \begin{cases} 0 & \text{if } f(x) = y \\ 1 & \text{otherwise} \end{cases}$$

  - Density estimation:

$$L(z, f) = -\log f(z)$$

The Risk and the Empirical Risk

- Learning: search for a good function in a function space $\mathcal{F}$
- Minimize the Expected Risk on $\mathcal{F}$, defined for a given f as

$$R(f) = E_Z[L(z, f)] = \int_Z L(z, f)p(z)dz$$

- Induction Principle:
  - select $f^* = \arg\min_{f \in \mathcal{F}} R(f)$
  - problem: $p(z)$ is unknown!!!
- Empirical Risk:

$$\hat{R}(f, D_n) = \frac{1}{n} \sum_{i=1}^{n} L(z_i, f)$$

The Risk and the Empirical Risk

- The empirical risk is an unbiased estimate of the risk:

$$E_D[\hat{R}(f, D)] = R(f)$$

- The principle of empirical risk minimization:

$$f^*(D_n) = \arg\min_{f \in \mathcal{F}} \hat{R}(f, D_n)$$

- Training error:

$$\hat{R}(f^*(D_n), D_n) = \min_{f \in \mathcal{F}} \hat{R}(f, D_n)$$

- Is the training error a biased estimate of the risk?

The Training Error

- Is the training error biased? yes.

$$E[R(f^*(D_n)) - \hat{R}(f^*(D_n), D_n)] \geq 0$$

- The solution $f^*(D_n)$ found by minimizing the training error is better on D_n than on any other set $D_n^{'}$ drawn from $p(Z)$.
- Can we bound the difference between the training error and the generalization error?

$$|R(f^*(D_n)) - \hat{R}(f^*(D_n), D_n)| \leq ?$$

- Answer: under certain conditions on $\mathcal{F}$, yes.
- These conditions depend on the notion of capacity h of $\mathcal{F}$.

The Capacity

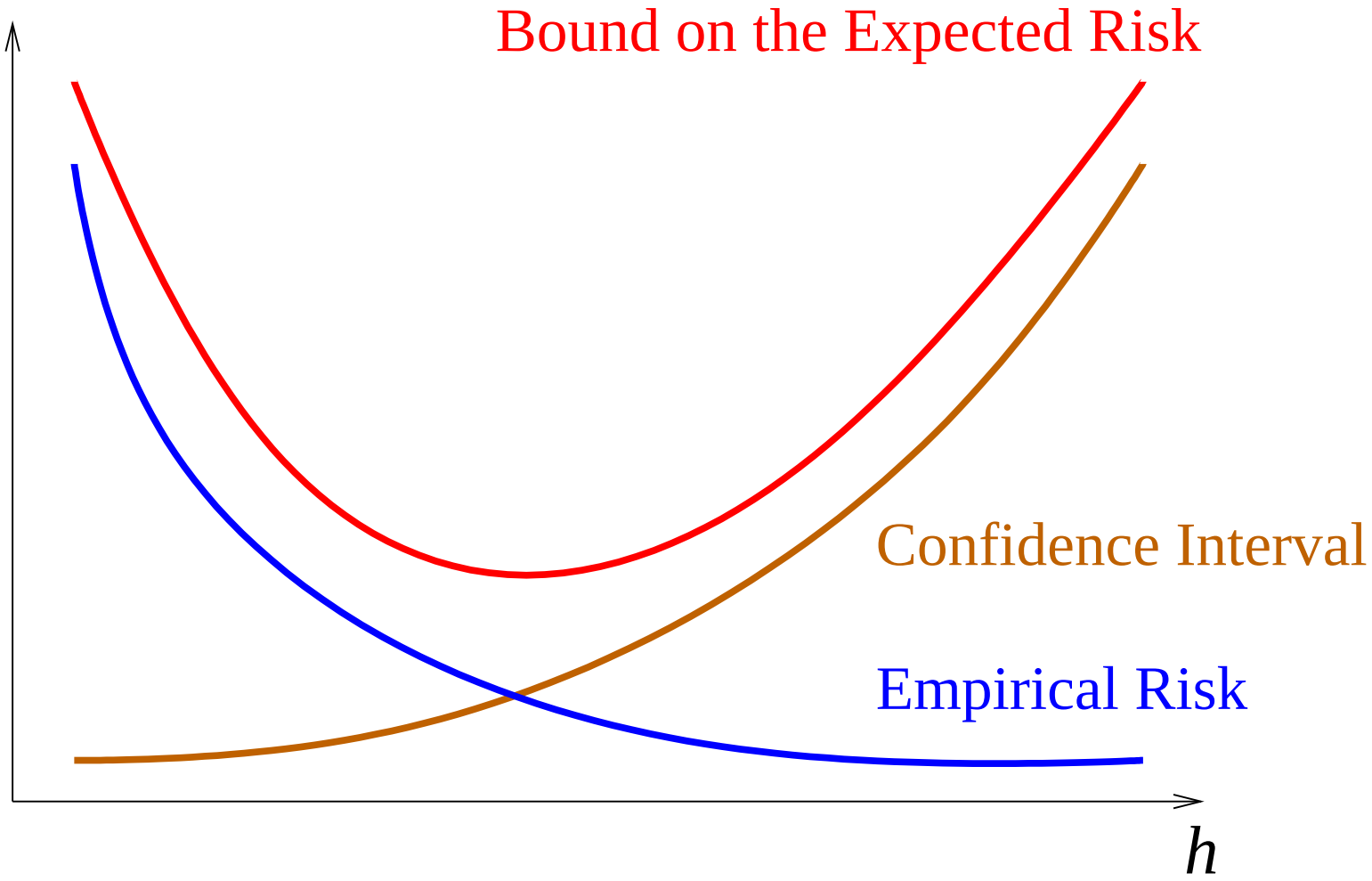
- The capacity $h(\mathcal{F})$ is a measure of its size, or complexity.
- Classification:

  The capacity $h(\mathcal{F})$ is the largest n such that there exist a set of examples D_n such that one can always find an $f \in \mathcal{F}$ which gives the correct answer for all examples in D_n, for any possible labeling.

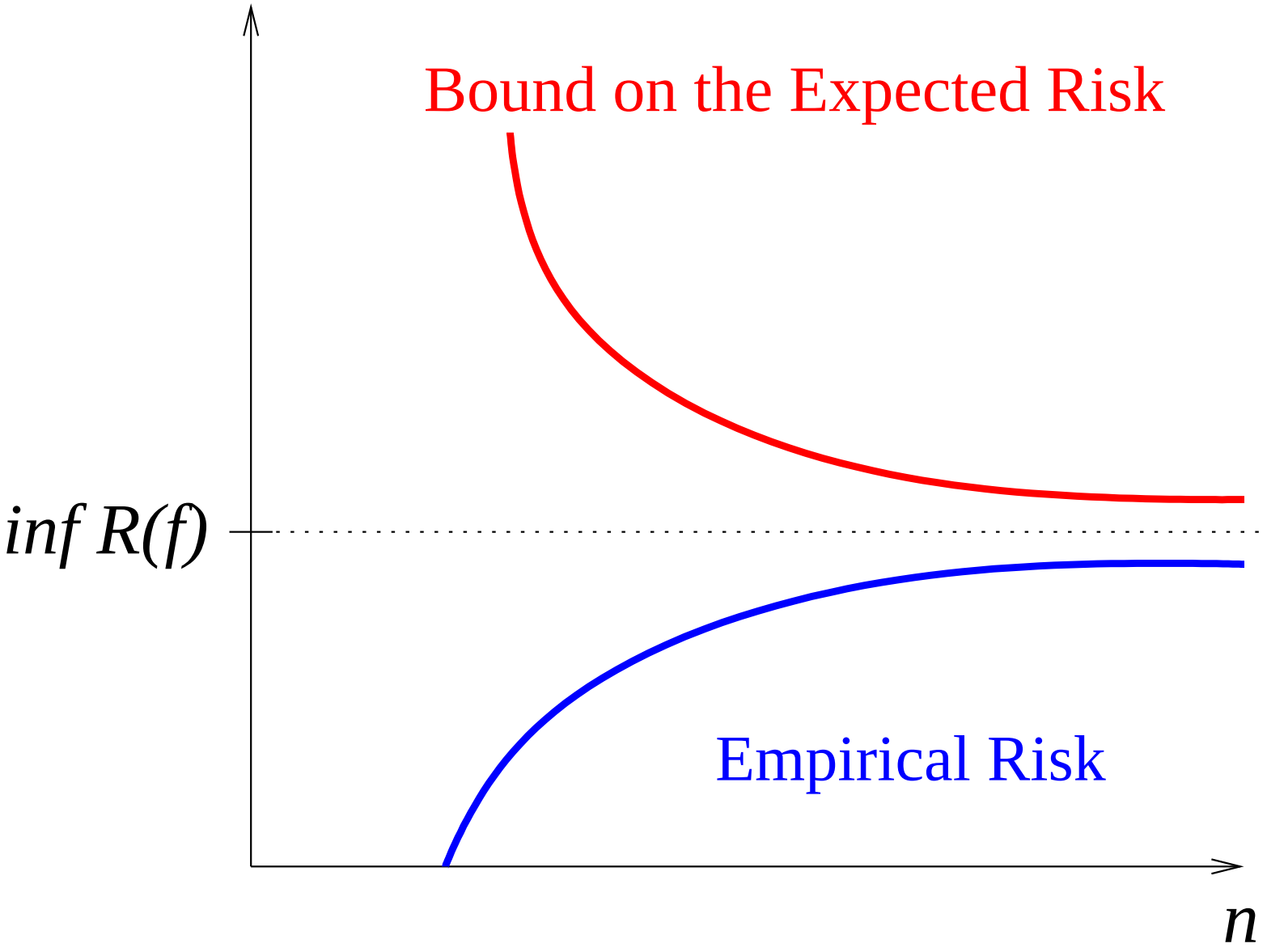
- Regression and density estimation: capacity exists also, but more complex to derive (for instance, we can always reduce a regression problem to a classification problem).
- Bound on the expected risk: let $\tau = \sup L - \inf L$

$$P\left(\sup_{f\in\mathcal{F}} |R(f) - \hat{R}(f, D_n)| \leq 2\tau \sqrt{\frac{h\left(\ln\frac{2n}{h}+1\right) - \ln\frac{\eta}{9}}{n}}\right) \geq 1-\eta$$

Theoretical Curves



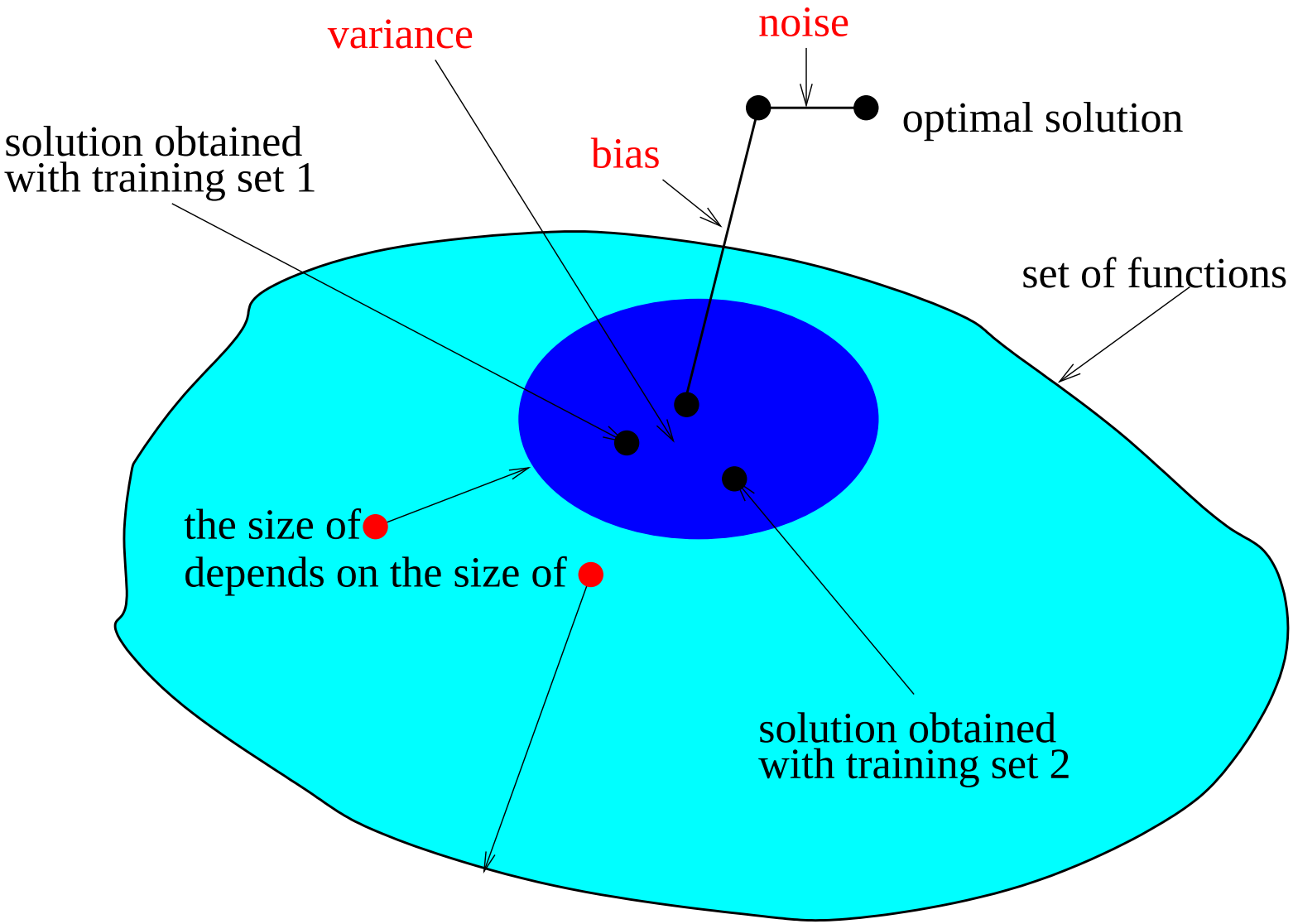
Theoretical Curves



The Bias-Variance Dilemma

- The generalization error can be decomposed into 3 parts:
  - the bias: due to the fact that the set of functions $\mathcal{F}$ does not contain the optimal solution,
  - the variance: due to the fact that if we had been using another set $D_n^{'}$ drawn from the same distribution $p(Z)$, we would have obtained a different solution,
  - the noise: even the optimal solution could be wrong! (for instance if for a given x there are more than one possible y)
- Intrinsic dilemma: when the capacity $h(\mathcal{F})$ grows, the bias goes down, but the variance goes up!

The Bias-Variance Dilemma (Graphical View)



Regularization

- We have seen that learning = searching in a set of functions
- This set should not be too small (underfitting)
- This set should not be too large (overfitting)
- One solution: regularization
- Penalize functions f according to a prior knowledge
- For instance, penalize functions that have very large parameters

$$f^*(D_n) = \arg\min_{f \in \mathcal{F}} \hat{R}(f, D_n) + H(f)$$

with $H(f)$ a function that penalizes according to your prior

- For example, in some models:

small parameters $\rightarrow$ simpler solutions $\rightarrow$ less capacity

Early Stopping

- Another method for regularization: early stopping.
- Works when training is an iterative process.
- Instead of selecting the function that minimizes the empirical risk on D_n, we can do:
  - divide your training set D_n into two parts
    - train set $D^{tr} = \{z_1, z_2, \cdots, z_{tr}\}$
    - validation set $D^{va} = \{z_{va+1}, z_{tr+2}, \cdots, z_{tr+va}\}$
    - $tr + va = n$
  - let $f^t(D^{tr})$ be the current function found at iteration t
  - let $\hat{R}(f^t(D^{tr}), D^{va}) = \frac{1}{va} \sum_{z_i \in D^{va}} L(z_i, f^t(D^{tr}))$
  - stop training at iteration t^* such that

$$t^* = \arg\min_t \hat{R}(f^t(D^{tr}), D^{va})$$

and return function $f(D_n) = f^{t^*}(D^{tr})$

Methodology

- First: identify the goal! It could be
  1. to give the best model you can obtain given a training set?
  2. to give the expected performance of a model obtained by empirical risk minimization given a training set?
  3. to give the best model and its expected performance that you can obtain given a training set?
- If the goal is (1): use need to do model selection
- If the goal is (2), you need to estimate the risk
- If the goal is (3): use need to do both!
- There are various methods that can be used for either risk estimation or model selection:
  - simple validation
  - cross validation (k-fold, leave-one-out)
  - sequential validation (for time series)

Model Selection - Validation

- Select a family of functions with hyper-parameter θ
- Divide your training set D_n into two parts
  - $D^{tr} = \{z_1, z_2, \cdots, z_{tr}\}$
  - $D^{te} = \{z_{tr+1}, z_{tr+2}, \cdots, z_{tr+te}\}$
  - $tr + te = n$
- For each value θ_m of the hyper-parameter θ
  - select $f^*_{\theta_m}(D^{tr}) = \arg \min_{f \in \mathcal{F}_{\theta_m}} \hat{R}(f, D^{tr})$
  - let $\hat{R}(f^*_{\theta_m}, D^{te}) = \frac{1}{te} \sum_{z_i \in D^{te}} L(z_i, f^*_{\theta_m}(D^{tr}))$
- select $\theta^*_m = \arg \min_{\theta_m} \hat{R}(f^*_{\theta_m}, D^{te})$
- return $f^*(D_n) = \arg \min_{f \in \mathcal{F}_{\theta^*_m}} \hat{R}(f, D_n)$

Model Selection - Cross-validation

- Select a family of functions with hyper-parameter θ
- Divide your training set D_n into k distinct and equal parts $D^1, \cdots, D^k$
- For each value θ_m of the hyper-parameter θ
  - For each part D^j (and its counterpart $\bar{D}^j$)
    - select $f^*_{\theta_m}(\bar{D}^j) = \arg \min_{f \in \mathcal{F}_{\theta_m}} \hat{R}(f, \bar{D}^j)$
    - let $\hat{R}(f^*_{\theta_m}(\bar{D}^j), D^j) = \frac{1}{|D^j|} \sum_{z_i \in D^j} L(z_i, f^*_{\theta_m}(\bar{D}^j))$
  - estimate $\hat{R}_{\theta_m}(f) = \frac{1}{k} \sum_j \hat{R}(f^*_{\theta_m}(\bar{D}^j), D^j)$
- select $\theta^*_m = \arg \min_{\theta_m} \hat{R}_{\theta_m}(f)$
- return $f^*(D_n) = \arg \min_{f \in \mathcal{F}_{\theta^*_m}} \hat{R}(f, D_n)$

Estimation of the Risk - Validation

- Divide your training set D_n into two parts
  - $D^{tr} = \{z_1, z_2, \cdots, z_{tr}\}$
  - $D^{te} = \{z_{tr+1}, z_{tr+2}, \cdots, z_{tr+te}\}$
  - $tr + te = n$
- select $f^*(D^{tr}) = \arg\min_{f \in \mathcal{F}} \hat{R}(f, D^{tr})$

  (this optimization process could include model selection)
- estimate $R(f) = \hat{R}(f^*(D^{tr}), D^{te}) = \frac{1}{te} \sum_{z_i \in D^{te}} L(z_i, f^*(D^{tr}))$

Estimation of the Risk - Cross-validation

- Divide your training set D_n into k distinct and equal parts $D^1, \cdots, D^k$
- For each part D^j
  - let $\bar{D}^j$ be the set of examples that are in D_n but not in D^j
  - select $f^*(\bar{D}^j) = \arg\min_{f \in \mathcal{F}} \hat{R}(f, \bar{D}^j)$

    (this process could include model selection)
  - let $\hat{R}(f^*(\bar{D}^j), D^j) = \frac{1}{|D^j|} \sum_{z_i \in D^j} L(z_i, f^*(\bar{D}^j))$
- estimate $R(f) = \frac{1}{k} \sum_j \hat{R}(f^*(\bar{D}^j), D^j)$
- When $k = n$: leave-one-out cross-validation

Estimation of the Risk - Sequential Validation

- When data is sequential in nature (time series)
- Divide your training set D_n into k distinct and equal sequential blocks $D^1, \cdots, D^k$
- For $j = 1 \rightarrow k-1$
  - select $f^*(D^{1 \rightarrow j}) = \arg\min_{f \in \mathcal{F}} \hat{R}(f, \bigcup_{i=1}^{j} D^i)$

    (this process could include model selection)
  - let $\hat{R}(f^*(D^{1 \rightarrow j}), D^{j+1}) = \frac{1}{|D^{j+1}|} \sum_{z_i \in D^{j+1}} L(z_i, f^*(D^{1 \rightarrow j}))$
- estimate $R(f) = \frac{1}{k-1} \sum_{j=1}^{k-1} \hat{R}(f^*(D^{1 \rightarrow j}), D^{j+1})$

Estimation of the Risk - Bootstrap

- Is our estimate of the risk really accurate?
- Let us use Bootstrap to estimate the accuracy of a given statistics:
  - Let us create N bootstraps of D_n
  - For each bootstrap B_i, get an estimate of the risk R_i (using cross-validation for instance)
  - You can now compute estimates of the mean and the standard deviation of your estimates of the risk:

$$\bar{R} = \frac{1}{N} \sum_{j=1}^{N} R_j$$

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} \left(R_i - \bar{R}\right)^2}$$

Bootstrap

- Given a data set D_n with n examples drawn from $p(Z)$
- A bootstrap B_i of D_n also contains n examples:
- For $j = 1 \rightarrow n$, the j^{th} example of B_i is drawn independently with replacement from D_n
- Hence,
  - some examples from D_n are in multiple copies in B_i
  - and some examples from D_n are not in B_i
- Hypothesis: the examples were iid drawn from $p(Z)$
- Hence, the datasets B_i are as plausible as D_n, but drawn from D_n instead of $p(Z)$.

Estimation of the Risk and Model Selection

- When you want both the best model and its expected risk.
- You then need to merge the methods already presented.
  For instance:
  - train-validation-test: 3 separate data sets are necessary
  - cross-validation + test: cross-validate on train set, then test on separate set
  - double-cross-validation: for each subset, need to do a second cross-validation with the $k-1$ other subsets
- Other important methodological aspects:
  - compare your results with other methods!!!!
  - use statistical tests to verify significance
  - verify your model on other datasets

Train - Validation - Test

- Select a family of functions with hyper-parameter θ
- Divide your training set D_n into three parts D^{tr}, D^{va}, and D^{te}
- For each value θ_m of the hyper-parameter θ
  - select $f^*_{\theta_m}(D^{tr}) = \arg\min_{f \in \mathcal{F}_{\theta_m}} \hat{R}(f, D^{tr})$
  - let $\hat{R}(f^*_{\theta_m}, D^{va}) = \frac{1}{va} \sum_{z_i \in D^{va}} L(z_i, f^*_{\theta_m}(D^{tr}))$
- select $\theta^*_m = \arg\min_{\theta_m} \hat{R}(f^*_{\theta_m}, D^{va})$
- select $f^*(D^{tr} \cup D^{va}) = \arg\min_{f \in \mathcal{F}_{\theta^*_m}} \hat{R}(f, D^{tr} \cup D^{va})$
- estimate $R(f) = \frac{1}{te} \sum_{z_i \in D^{te}} L(z_i, f^*(D^{tr} \cup D^{va}))$

Cross-validation + Test

- Select a family of functions with hyper-parameter θ
- Divide you dataset D_n into two parts:

  a training set D^{tr} and a test set D^{te}
- For each value θ_m of the hyper-parameter θ

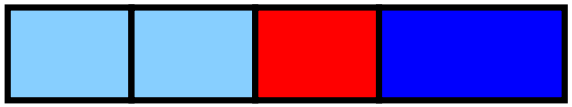
  estimate $\hat{R}_{\theta_m}(D^{tr})$ using cross-validation
- select $\theta^*_m = \arg\min_{\theta_m} \hat{R}_{\theta_m}(D^{tr})$
- retrain $f^*(D^{tr}) = \arg\min_{f \in \mathcal{F}_{\theta^*_m}} \hat{R}(f, D^{tr})$
- estimate $R(f) = \frac{1}{te} \sum_{z_i \in D^{te}} L(z_i, f^*(D^{tr}))$

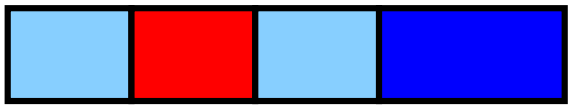
Double Cross-validation

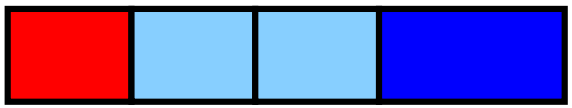
- Select a family of functions with hyper-parameter θ
- Divide your training set D_n into k distinct and equal parts $D^1, \cdots, D^k$
- For each part D^j
  - select the best model $f^*(\bar{D}^j)$ by cross-validation on $\bar{D}^j$
  - let $\hat{R}(f^*(\bar{D}^j), D^j) = \frac{1}{|D^j|} \sum_{z_i \in D^j} L(z_i, f^*(\bar{D}^j))$
- estimate $R(f) = \frac{1}{k} \sum_j \hat{R}(f^*(\bar{D}^j), D^j)$
- Note: this process only gives you an estimate of the risk, but not a model. If you need the model as well, you have to perform a separate model selection process!

Double Cross-validation

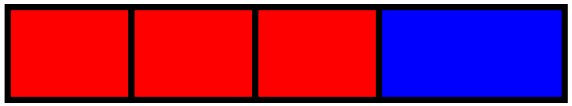
the whole dataset is
cut into 3 parts







the first 2 parts are cut into 3 parts
then perform a 3–fold cross–valid to
select the best hyper–parameter



the best hyper–parameter is used to
retrain on the 2 original parts and test
on the other one

... and do the same for each part to estimate risk