Re-sampling methods in the structural identification context

Olivier Caelen

Université Libre de Bruxelles; Bruxelles, Belgium

Machine Learning Group

Table of Contents

- The estimation of the generalisation error.
- Some re-sampling methods.
- The model selection problem.

Estimation of the accuracy of a model

The estimation of the accuracy of a model induced by supervised learning is important for :

- Choosing the best model from a given set (model selection).
- Choosing some models from a given set to combine them.
- Estimate the accuracy of the final model.

Machine Learning Group

Some notations (1)

- $\mathcal{X}$ is the *input space* and $\mathcal{X} \subset \mathbb{R}^n$.
- $\mathcal{Y}$ is the *output space* and $\mathcal{Y} \subset \mathbb{R}$.
- $\mathbf{x}$ is a random input vector of dimension n.
- $\mathbf{y}$ is a random output variable.
- $\mathbf{y} = f(\mathbf{x}) + \epsilon$, where ϵ is a random variable.
- $z = \langle x, y \rangle$ is a realisation of $\mathbf{z} = \langle \mathbf{x}, \mathbf{y} \rangle$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.

Machine Learning Group

Some notations (2)

- $D_N = (\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \ldots, \langle x_i, y_i \rangle, \ldots, \langle x_N, y_N \rangle)$ is a training set, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$.
- $z_i = \langle x_i, y_i \rangle$ is a training sample.
- N is the number of training samples in D_N.

Machine Learning Group

The complexity of a model (1)

$\Psi_1, \Psi_2, \ldots, \Psi_v, \ldots$ are different classes of model, some examples:

- $\Psi_1 \Rightarrow$ Feedforward Neural Network.
- $\Psi_2 \Rightarrow$ Lazy-Learning.
- $\Psi_3 \Rightarrow$ Regression Tree.
- $\Psi_4 \Rightarrow$ Support Vector Machine.
- $\ldots$

Ψ_v is the set of all the *structures of model* of the *class model* v.

Machine Learning Group

The complexity of a model (2)

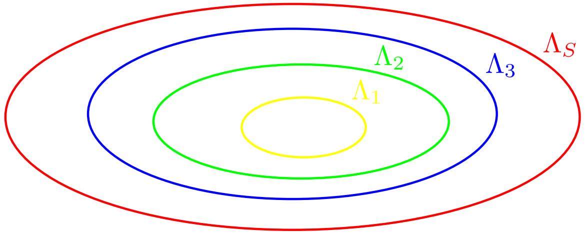
Λ_s is the set of all the *structures* with complexity s.

$$\Lambda_1 \subseteq \ldots \subseteq \Lambda_s \subseteq \ldots \subseteq \Lambda_S$$

and

$$\Lambda_S = \bigcup_{s=1}^{S} \Lambda_s$$

A figure with the different sets:



Machine Learning Group

The complexity of a model (3)

Γ_{vs} is the set of all models of class v and complexity s

$$\Gamma_{vs} = \Psi_v \cap \Lambda_s$$

It follows :

$$\Gamma_{v1} \subseteq \ldots \subseteq \Gamma_{vs} \subseteq \ldots \subseteq \Gamma_{vS}$$

Machine Learning Group

Estimation function

- $\hat{h}(x,\alpha)$ is an estimation of $f(x)$.
- $\alpha \in \Gamma_{vs}$ and α is the vector of parameters of $\hat{h}$.
- $R_{emp}^{N}(\alpha)$ is the empirical risk function of α on D_N.
- $\alpha_{N} = \arg\min_{\alpha\in\Gamma_{vs}} R_{emp}^{N}(\alpha)$
- $\hat{h}(x,\alpha_{N})$ is the best estimator of $f(x)$, build on D_N in Γ_{vs}.

Machine Learning Group

the generalisation error

- MSE is the generalisation error of $\hat{h}(x, \alpha_N)$:

$$MSE(\alpha_N) = \int_{\mathcal{X},\mathcal{Y},\mathcal{Z}^N} L(z, \alpha_N) dP^N(D_N)\, dP(y|x) dP(x)$$

- Where, $L(z, \alpha_N)$ is the prediction error of $\hat{h}(x, \alpha_N)$.
- $L(z, \alpha_N)$ is called *the cost function*.

Machine Learning Group

Some examples of cost functions

- In regression :

$$L(z, \alpha_N) = \left(y - \hat{h}\,(x, \alpha_N)\right)^2$$

or

$$L(z, \alpha_N) = \left|y - \hat{h}\,(x, \alpha_N)\right|$$

- In classification :

$$L(z, \alpha_N) = \begin{cases} = 1 & \textbf{if}(y = \hat{h}\,(x, \alpha_N)) \\ = 0 & \textbf{if}(y \neq \hat{h}\,(x, \alpha_N)) \end{cases}$$

Machine Learning Group

The choose of the cost function [3]

Supposed $N \to \infty$:

- **if** $L(z, \alpha_N) = (\ldots)^2$ **then** the best model, who minimizes MSE, is $E_{\mathbf{y},\mathbf{x}}[y|x]$
- **if** $L(z, \alpha_N) = |\ldots|$ **then** the best model, who minimizes MSE, is $median_{\mathbf{y},\mathbf{x}}[y|x]$

Machine Learning Group

re-sampling methods

There are different methods to estimate the MSE by re-sampling:

- Empirical risk function.
- Holdout.
- Monte-Carlo cross-validation.
- V fold cross-validation.
- Leave-one-out cross-validation.
- Bootstrap.
- Bootstrap 632.

Empirical risk function

$$\widehat{MSE}_{emp}(\alpha_N) = R^N_{emp}(\alpha_N)$$

As complexity increases, $R^N_{emp}(\alpha_N)$ decreases.
$\rightarrow$ this estimator privileges too complex models.
$\rightarrow$ this estimator privileges models who make overfitting!

$R^N_{emp}(\alpha_N)$ is a bad estimator of the $MSE(\alpha_N)$.

Machine Learning Group

Holdout

- The training sample D_N is randomly split into two parts:

$$D_{N^{val}} = D_N / D_{N^{tr}}$$

- $D_{N^{tr}}$ is used to find $\alpha_{N^{tr}}$.

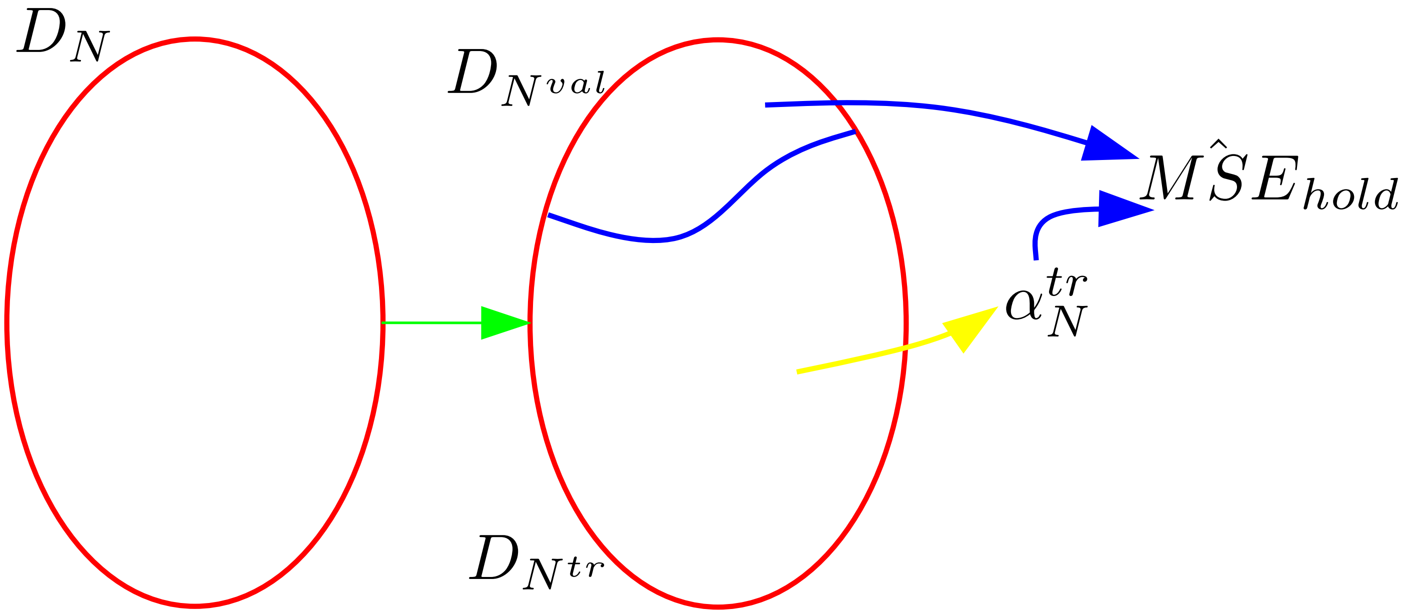
$$\alpha_{N}^{tr} = \arg\min_{\alpha \in \Gamma_{vs}} R_{emp}^{N^{tr}}(\alpha)$$

- $D_{N^{val}}$ is used to estimate the accuracy of $\alpha_{N^{tr}}$.

$$\widehat{MSE}_{hold}\left(\alpha_N\right) = \frac{\sum_{i=1}^{N^{val}} L\left(z_i, \alpha_{N^{tr}}\right)}{N^{val}}$$

Holdout

The holdout method splits the data into two mutually exclusive subsets called the *training set* and the *validation set*:



Notations for the cross-validation (1)

In each cross-validation method, we create a set D^* such as :

$$D^* = \left\{ \left\langle D_{N_1^{tr}}, D_{N_1^{val}} \right\rangle, \ldots, \left\langle D_{N_k^{tr}}, D_{N_k^{val}} \right\rangle, \ldots \left\langle D_{N_K^{tr}}, D_{N_K^{val}} \right\rangle \right\}$$

You can see that each D_N is split into two parts:

$$D_{N_k^{val}} = D_N / D_{N_k^{tr}}$$

Machine Learning Group

Notations for the cross-validation (2)

For each couple of D^* we compute :

$$\alpha_{N_k^{tr}} = \arg\min_{\alpha \in \Gamma_{vs}} R_{emp}\left(\alpha, D_{N_k^{tr}}\right)$$

And,

$$\widehat{MSE}_{cv}^{k}\left(\alpha_N\right) = \frac{\sum_{i \in N_k^{val}} L\left(z_i, \alpha_{N_k^{tr}}\right)}{N_k^{val}}$$

Machine Learning Group

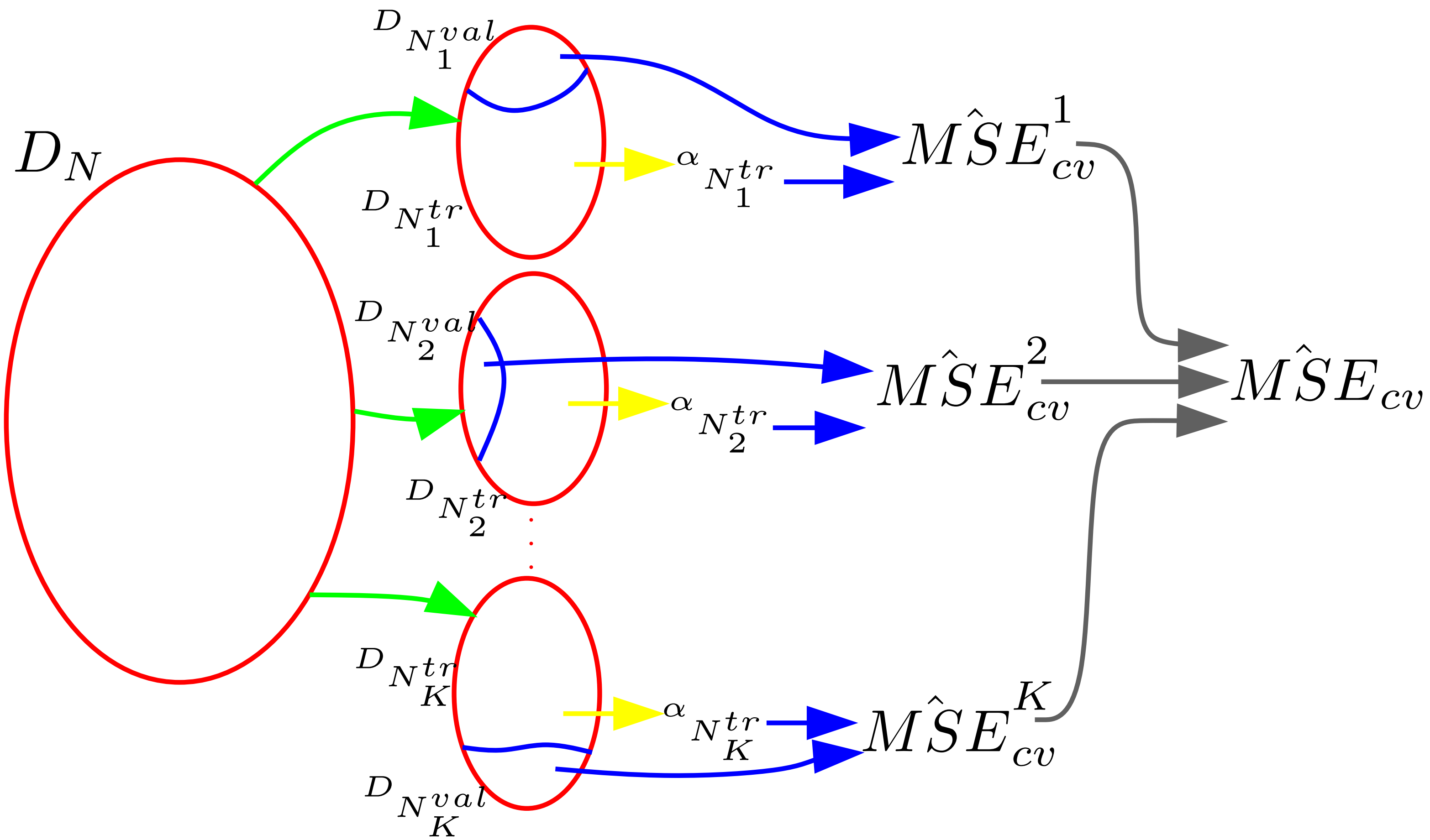
Notations for the cross-validation (3)

And finally:

$$\widehat{MSE}_{cv}(\alpha_N) = \frac{\sum_{k=1}^{K} \widehat{MSE}_{cv}^{k}(\alpha_N)}{K}$$

Each cross-validation method differs in the way of creating D^*.

Notations for the cross-validation (4)



Monte-Carlo cross-validation

- To create a couple $\left\langle D_{N_k^{tr}}, D_{N_k^{val}} \right\rangle$, the Monte-Carlo cross-validation selects randomly N^{tr} samples in D_N. These samples are put in $D_{N_k^{tr}}$.
  and finally: $D_{N_k^{val}} = D_N / D_{N_k^{tr}}$
- This procedure is repeated K times.
- All the couples $\left\langle D_{N_k^{tr}}, D_{N_k^{val}} \right\rangle$ in D^* are iid.

Machine Learning Group

V fold cross-validation(1)

V fold cross-validation is a *Monte-Carlo cross-validation* with two supplementary conditions :

- A condition on the values of N_k^{tr} and N_k^{val} :

$$\begin{cases} N_k^{val} = \lfloor N/V \rfloor & k \in [1, V-1] \\ N_V^{val} = N - (V-1)\lfloor N/V \rfloor \end{cases}$$

$$N_k^{tr} = N - N_k^{val} \quad k \in [1, V]$$

V fold cross-validation(2)

V fold cross-validation is a *Monte-Carlo cross-validation* with two supplementary conditions :

- some conditions on the position of each z_i :
  - Each z_i appears exactly one time in each $\left\langle D^k_{N^{tr}}, D^k_{N^{val}} \right\rangle$ couple of D^*.
  - Each z_i appears exactly $V-1$ times, in all the $D^k_{N^{tr}}$ sets
  - Each z_i appears exactly 1 time, in all the $D^k_{N^{val}}$ sets

V fold cross-validation(3)

- In [5], a good value for V is ten.
- *Leave-one-out cross-validation* is an extreme case of the *V fold cross-validation*, where $V = N$.

cross-validation property(1) [3]

IF

$$\widehat{MSE} \textit{ is estimate by cross validation.}$$

$$\hat{s} = \arg\min_{s} \widehat{MSE}\left(\alpha_N^s\right)$$

$$\tilde{s} = \arg\min_{s} MSE\left(\alpha_N^s\right)$$

$$\#val(N) \xrightarrow[N\to\infty]{} \infty$$

THEN

$$PL = MSE(\hat{s}) - MSE(\tilde{s}) \xrightarrow[N\to\infty]{} 0$$

Machine Learning Group

cross-validation property(2) [3]

- The convergence rate of PL to zero is in

$$O\left(\frac{log(S)}{\sqrt{\#val(N)}}\right)$$

- if $L(z, \alpha_N) = (\ldots)^2$ then the convergence rate of PL to zero is in

$$O\left(\frac{log(S)}{\#val(N)}\right)$$

Machine Learning Group

Leave-One-Out C-V failed

I will give an example where the leave-one-out cross-vaildation failed [5]:

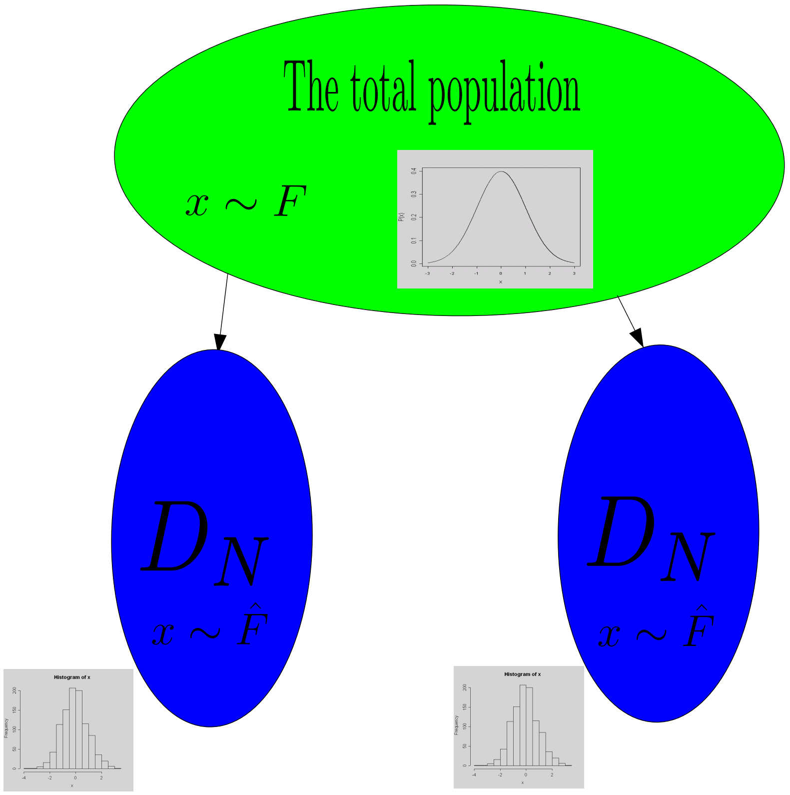
- **if** $v = the\ majority\ class\ model$
- and **if** D_N has only three classes with the same proportion (e.g. $3 * 50 = 150\ samples$)

Then the real accuracy is about 33% but $\widehat{MSE}_{loo} = 0\%$!

Machine Learning Group

Bootstrap(1)[4]

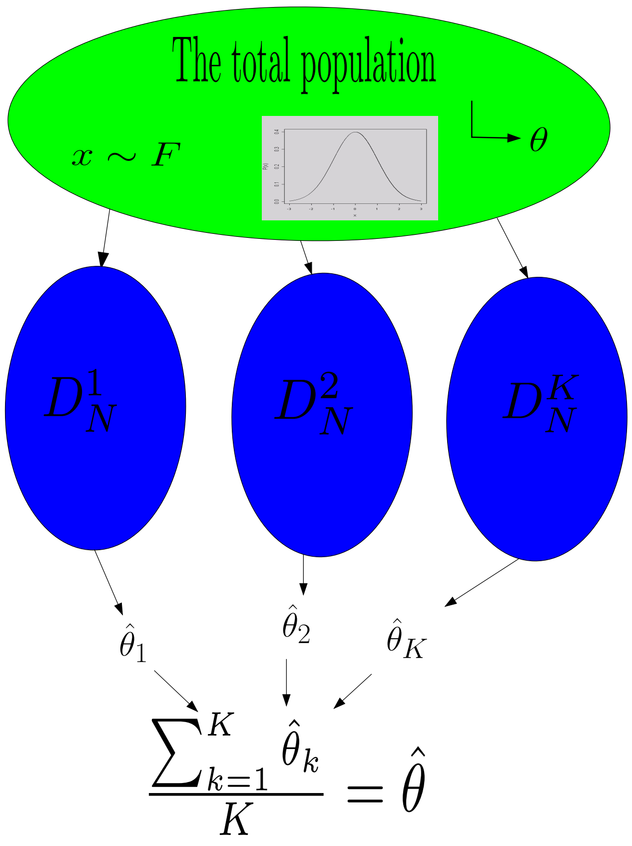
The distribution of the samples in D_N is an estimation of the distribution of the samples in the real population:



Machine Learning Group

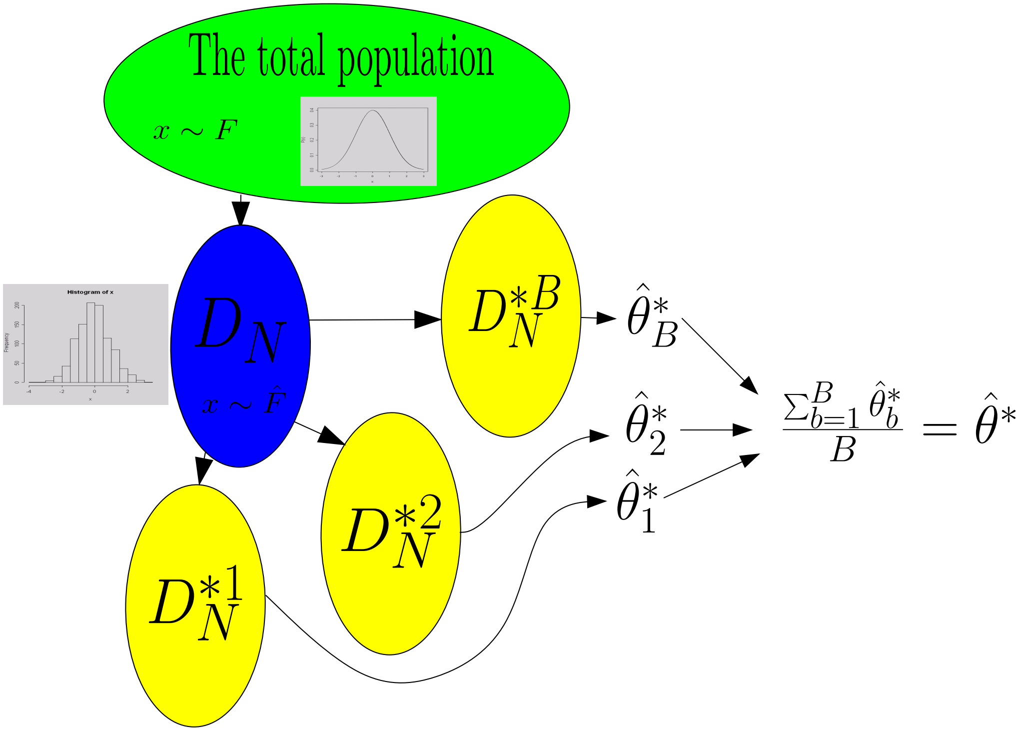
Bootstrap(2)

Estimation of a parameter θ with the total population:



Bootstrap(3)

Estimation of a parameter θ with the bootstrap:

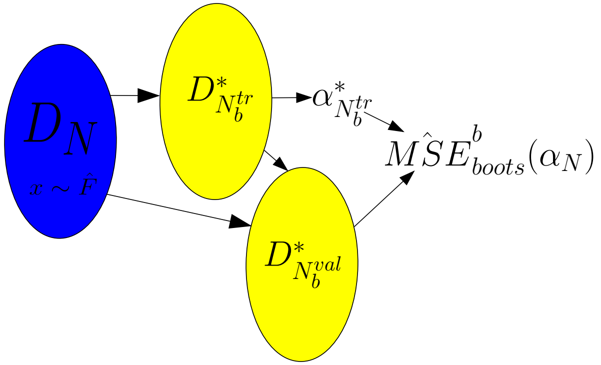


D_N^{*b} is created by sampling D_N with replacement.

Bootstrap(4)

One step in the estimation of the MSE with the Bootstrap method.

$$\widehat{MSE}^{b}_{boot}(\alpha_N) = \frac{\sum_{i \in N_b^{val}} L\left(z_i, \alpha^*_{N_b^{tr}}\right)}{N^{val}}$$



- $D^*_{N_b^{tr}}$ is created by sampling D_N with replacement.
- $D^*_{N_b^{val}} = D_N / D^*_{N_b^{tr}}$

Machine Learning Group

Bootstrap(5)

The bootstrap estimation $\widehat{MSE}_{boot}$ is the average of $\widehat{MSE}^{b}_{boot}$ with $b \in [1, B]$

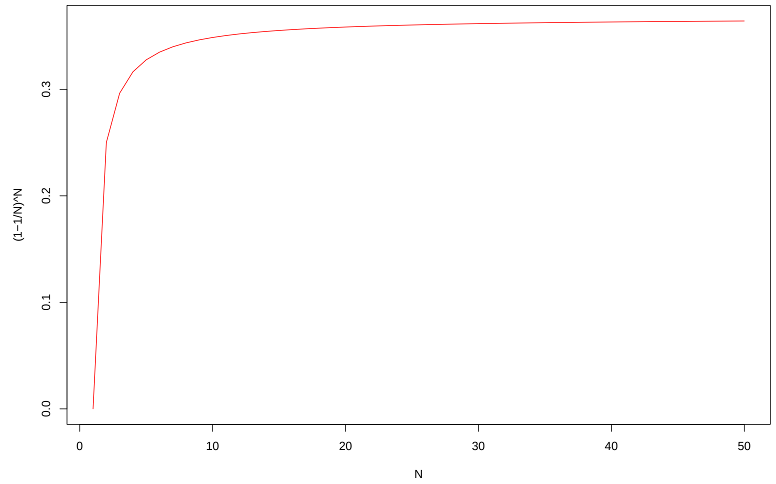
$$\widehat{MSE}_{boot}(\alpha_N) = \frac{\sum_{b=1}^{B} \widehat{MSE}^{b}_{boot}(\alpha_N)}{B}$$

Machine Learning Group

Bootstrap(6)

For each bootstrap sample, about $1/3$ of the cases is not in $D^*_{N_b^{tr}}$:

$$\left(1 - \frac{1}{N}\right)^N \xrightarrow[N\to\infty]{} e^{-1} \approx .368$$



About 36.8% of the samples in $D^*_{N_b^{tr}}$ are copies!

Machine Learning Group

Bootstrap 632

- the Bootstrap is a pessimistic estimator :
$E_{\mathbf{D_N}}\left[\widehat{MSE}_{boot}\right] - MSE > 0$.
- the Empirical risk function is a optimistic estimator :
$E_{\mathbf{D_N}}\left[\widehat{MSE}_{emp}\right] - MSE < 0$.
- the Bootstrap 632 tries to reduce the bias of the classical Bootstrap:
$\widehat{MSE}_{boot632}(\alpha_N) = \frac{\sum_{b=1}^{B}\left[0.632*\widehat{MSE}^{b}_{boot}(\alpha_N)+0.368*R^{N}_{emp}(\alpha_N)\right]}{B}$

Machine Learning Group

Bootstrap 632

A example where the Bootstrap632 fails [5] :

- **If** we have a perfect memorizer classifier (e.g. a one nearest neighbour classifier)
- and **if** the dataset is completely random, say with 2 classes.

Then

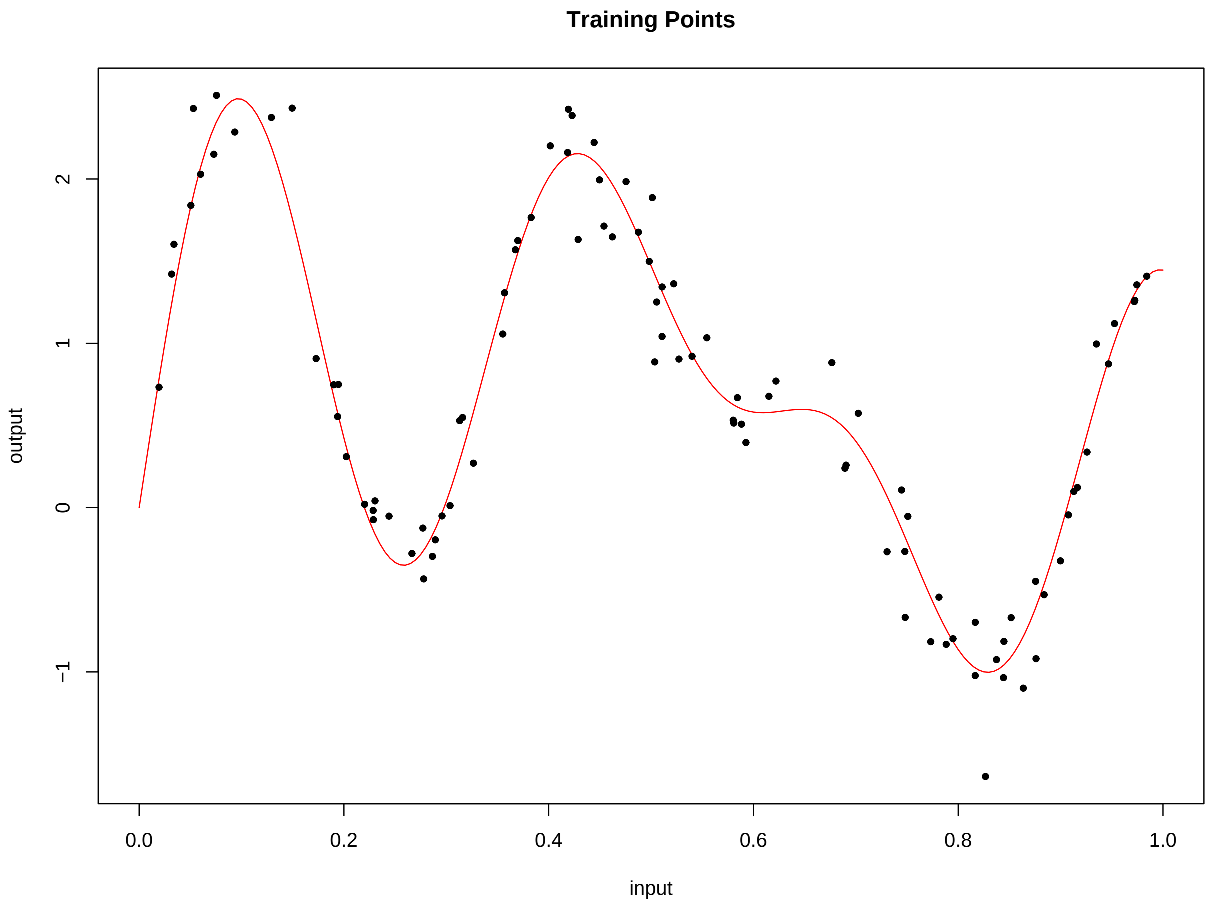
$$\widehat{MSE}_{boot632} \approx \frac{\sum_{b=1}^{B} 0.632 * 0.5 + 0.368 * 1}{B} = 68.4\%$$

and the real MSE is 50%!

Machine Learning Group

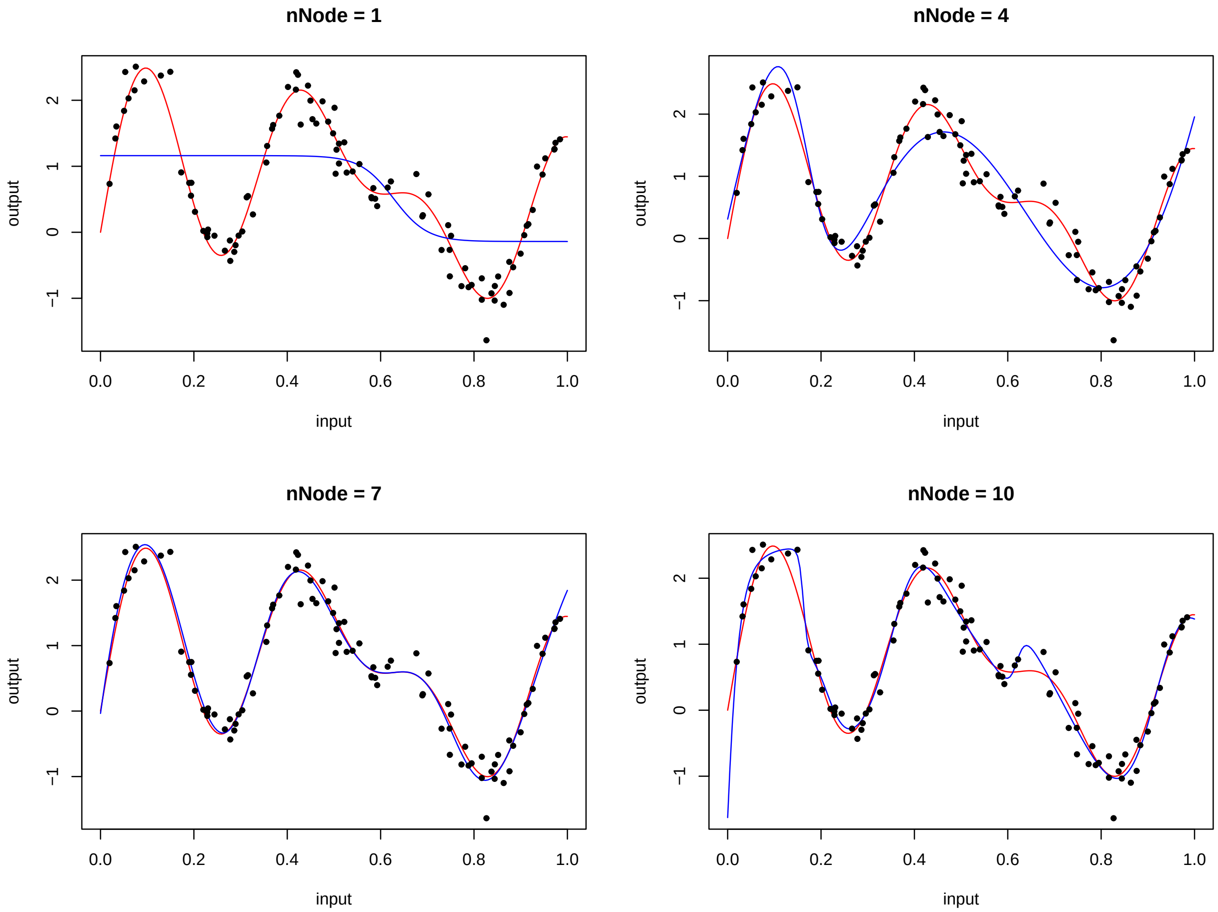
Example of model selection(1)

$$n = 1 \; , \; N = 100 \; , \; \mathcal{X} = [0, 1]$$



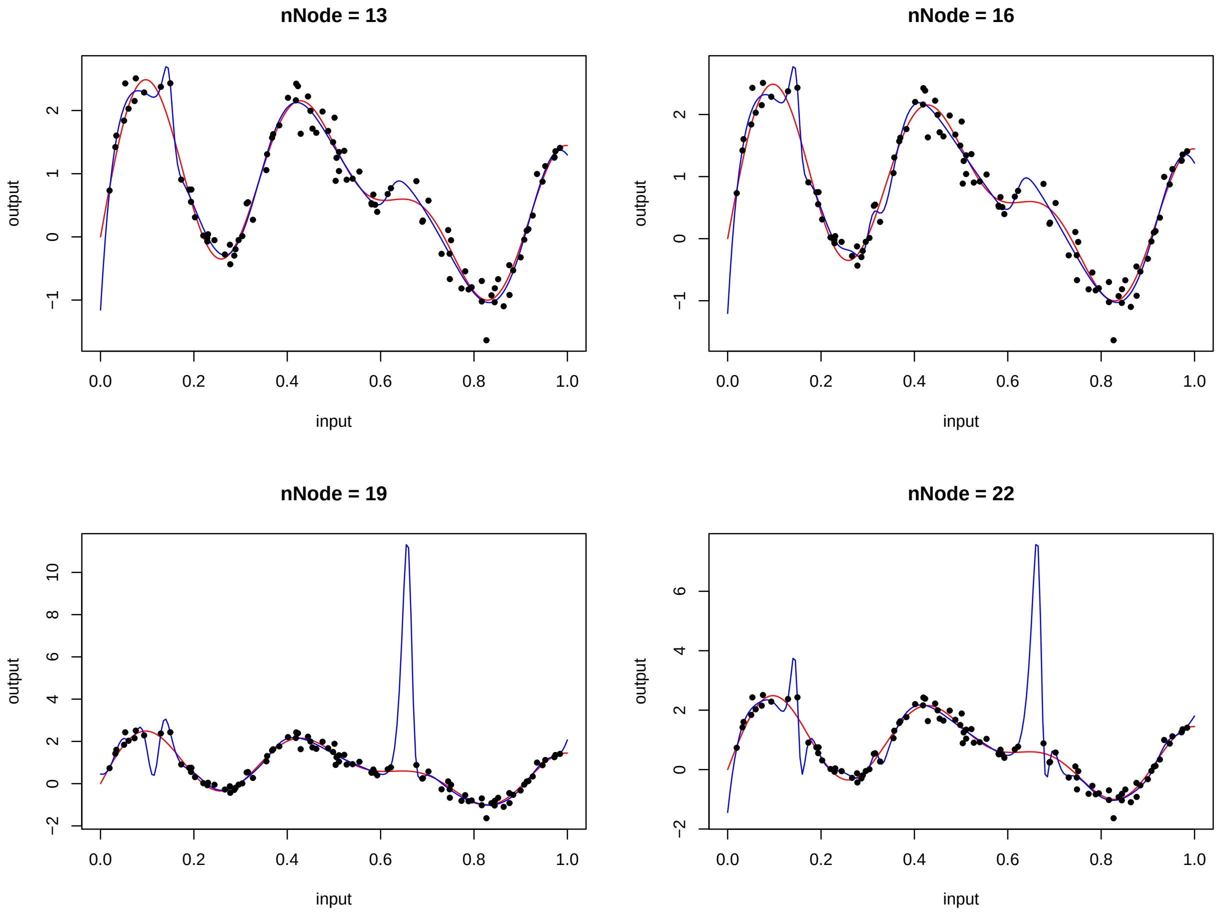
Example of model selection(1)

$v =$ *neural network* and $s = \{1, 4, 7, 10\}$



Example of model selection(2)

$v =$ *neural network* and $s = \{13, 16, 19, 22\}$

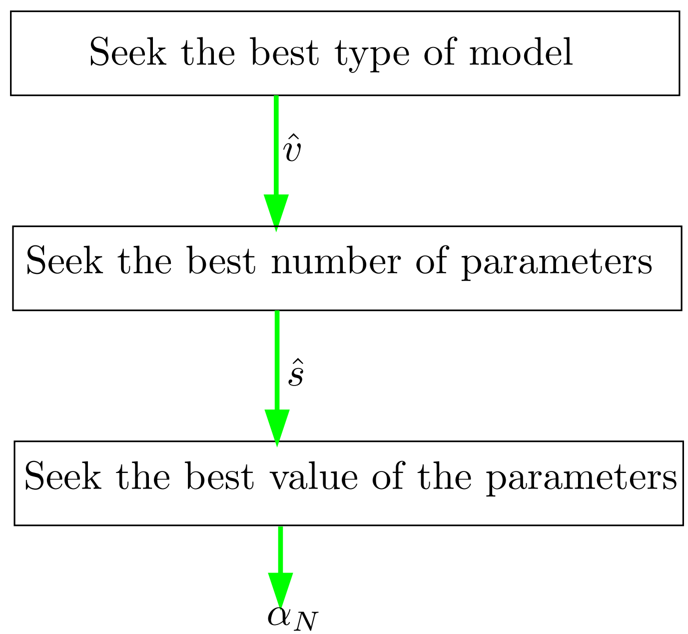


Machine Learning Group

The goal of model selection

The goal of *model selection* is to find the best v and s, which minimise the generalisation error of $\hat{h}(x, \alpha_N)$.

$$\langle \hat{v}, \hat{s} \rangle = \arg\min \widehat{MSE}^{\langle v,s \rangle}$$



Machine Learning Group

MSE and NMSE

- $\widehat{MSE}$ = estimation of the **M**ean **S**quared **E**rror
- $N\hat{M}SE$ = estimation of the **N**ormalized **M**ean **S**quared **E**rror

$$N\hat{M}SE = \frac{\widehat{MSE}}{v\hat{a}r(Y)}$$

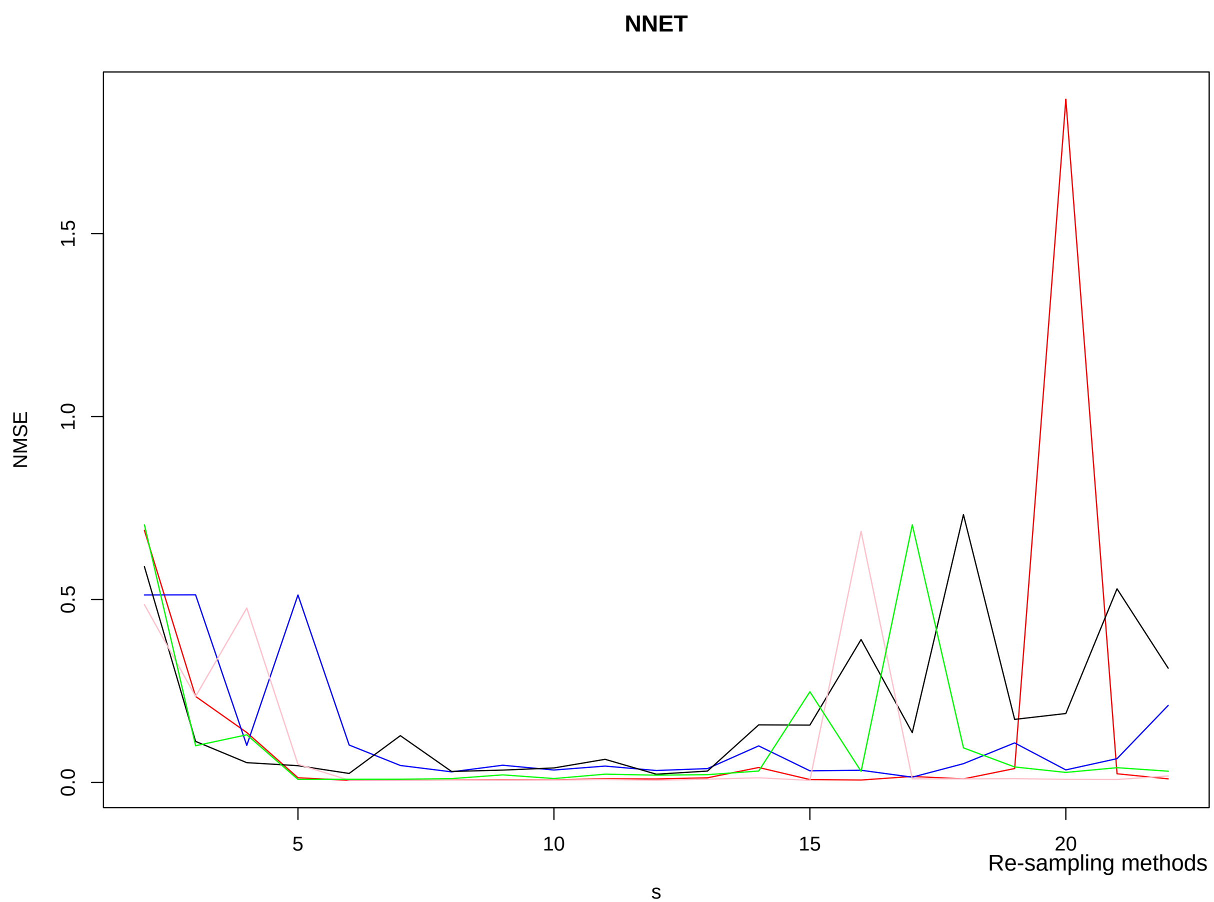
Interpretation of the NMSE

$$\begin{aligned} var(Y) &= \frac{\sum_{i=1}^{N} (y_i - \hat{\mu})^2}{N} \\ &= Learning\ Error\ of\ \hat{\mu} \\ &= R_{emp}(\hat{\mu}) \\ &\Downarrow \\ N\hat{M}SE &= \frac{\widehat{MSE}\left(\hat{h}\right)}{R_{emp}(\hat{\mu})} \end{aligned}$$

if $N\hat{M}SE > 1$, **then** $\hat{h}$ is "bad".

NMSE of a neural network

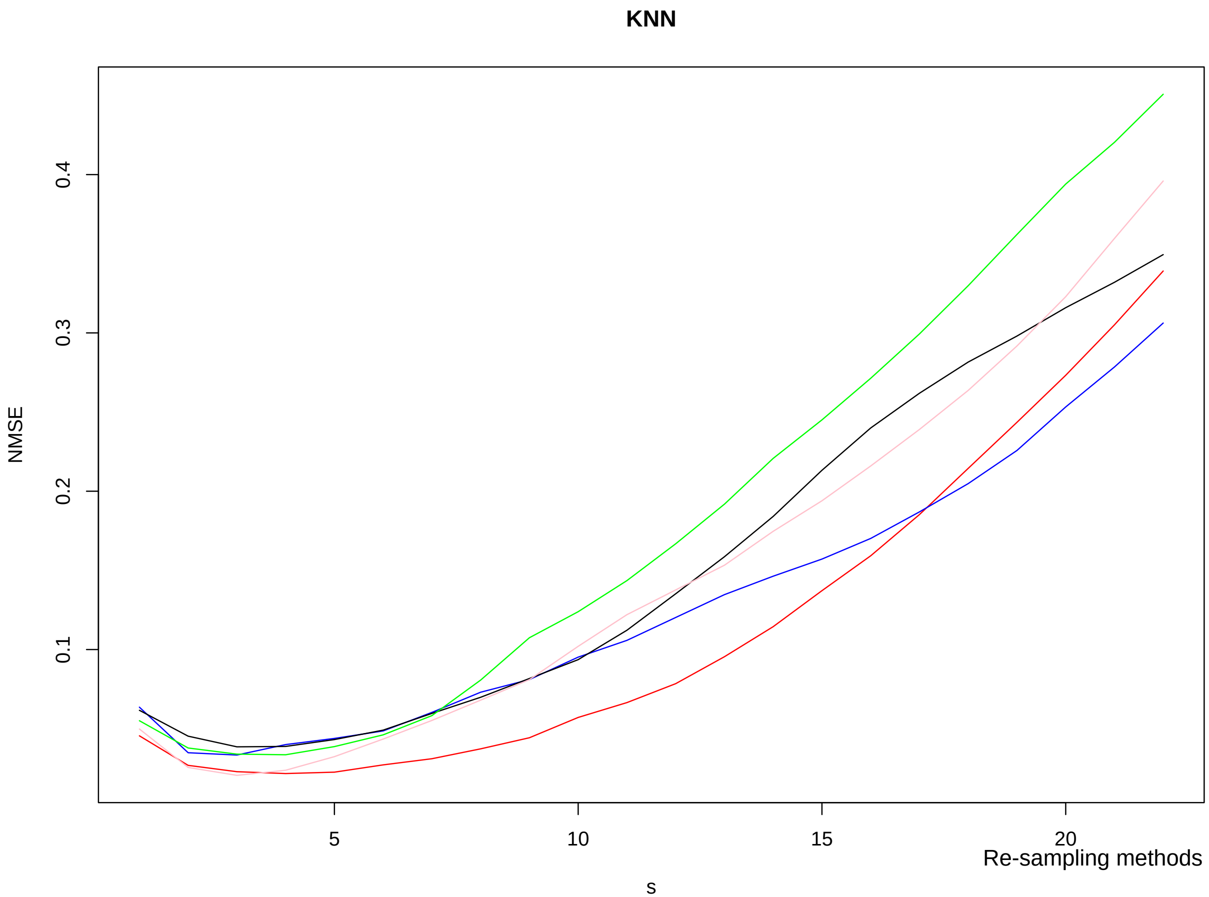
$N_{ts} = 10001$, $N_{tr} = 100$
On each run, I change only the $D_{N^{tr}}$.



Machine Learning Group

NMSE of a K-Nearest-Neighbour

$N_{ts} = 10001$, $N_{tr} = 100$
On each run, I change only the $D_{N^{tr}}$.



Stability of a class model v

- "*An inducer is **stable** for a given dataset and a set of perturbation, if it induces classifiers that make the same prediction when it is given the perturbed datasets*" [5]
- For [2], the *neural network* is an *unstable* class of model and the *KNN* is a *stable* class of model.
- And for [2, 5], the re-sampling methods for model selection do not work well with unstable classes.

Machine Learning Group

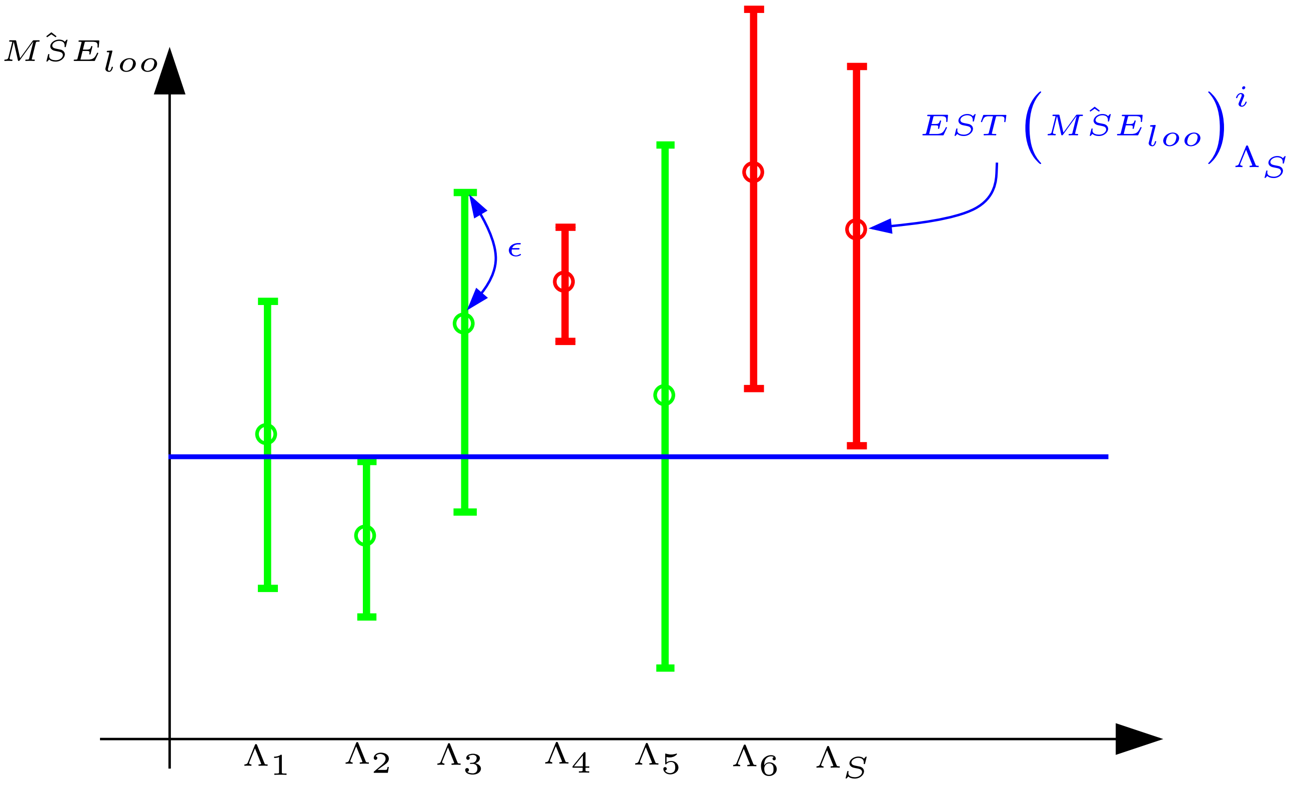
Speed-up the model selection

The racing method [6] can be used to speed up the model selection process:

- A set of models are tested in parallel by loo-cv.
- If a model is appreciably bad then it is withdrawn from the set.
- $\Longrightarrow$ The computation power is used for the estimation of the accuracy of the other modes.

Machine Learning Group

One step in the loo-cv process



What is the value of the bound ϵ?

the ϵ in the racing

There exist different methods for estimate ϵ :

- Hoeffding's Bound [6].
- Bayesian's Bound [6].
- F-Race [1]

Machine Learning Group

THANK YOU

for your attention!

You can find the slides of this presentation at this address:

`http://www.ulb.ac.be/di/mlg/seminars.html`

Machine Learning Group

References

[1] M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp. A racing algorithm for configuring metaheuristics. In W. B. Langdon, editor, *GECCO 2002*, pages 11–18. Morgan Kaufmann, 2002.

[2] L. Breiman. Heuritics of instability and stabilization in model selection. *The Annals of Statistics*, 24(6), December 1996.

[3] S. Dudoit and M.J. van der Laan. Asymptotics of cross-validation risk estimation in model selection and performance assessment. Technical Report 126, U.C. Berkeley, February 2003. http://www.bepress.com/ucbbiostat/paper126.

[4] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall, New York, 1993.

[5] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of IJCAI-95*, 1995. available at http://robotics.stanford.edu/users/ronnyk/ronnyk-bib.html.

[6] Oden Maron and Andrew W. Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11(1-5):193–225, 1997.