

Resampling techniques for statistical modeling

Gianluca Bontempi

Département d'Informatique

Boulevard de Triomphe - CP 212

<http://www.ulb.ac.be/di>

Beyond the empirical error

- Empirical error tend to be a *too optimistic* estimate of the true prediction error.
- The reason is that we are using the same data to assess the model as were used to fit it, using parameter estimates that are fine-tuned to our particular data set.
- In other words the **test** sample is the same as the original sample, sometimes called also the **training** sample.
- Estimates of prediction error obtained in this way are called **apparent** error estimates.

The linear case: MISE error

Let us compute now the expected prediction error of a linear model trained on D_N when this is used to predict for the same training inputs X a set of outputs y_{ts} distributed according to the same linear law but **independent** of the training output y . We call this quantity **Mean Integrated Squared Error**

$$\begin{aligned} \text{MISE} &= E_{\mathbf{D}_N, \mathbf{y}_{ts}} [(\mathbf{y}_{ts} - X\hat{\beta})^T (\mathbf{y}_{ts} - X\hat{\beta})] = \\ &= E_{\mathbf{D}_N, \mathbf{y}_{ts}} [(\mathbf{y}_{ts} - X\beta + X\beta - X\hat{\beta})^T (\mathbf{y}_{ts} - X\beta + X\beta - X\hat{\beta})] = \\ &= N\sigma_w^2 + E_{\mathbf{D}_N} [(X\beta - X\hat{\beta})^T (X\beta - X\hat{\beta})] \quad (1) \end{aligned}$$

Since

$$\begin{aligned} X\beta - X\hat{\beta} &= X\beta - X(X^T X)^{-1} X^T y = \\ &= X\beta - X(X^T X)^{-1} X^T (X\beta + w) = -X(X^T X)^{-1} X^T w \quad (2) \end{aligned}$$

we have

$$\begin{aligned} N\sigma_{\mathbf{w}}^2 + E_{\mathbf{D}_N}[(X\beta - X\hat{\beta})^2] &= N\sigma_{\mathbf{w}}^2 + E_{\mathbf{D}_N}[\mathbf{w}^T X (X^T X)^{-1} X^T X (X^T X)^{-1} X \mathbf{w}] \\ &= N\sigma_{\mathbf{w}}^2 + E_{\mathbf{D}_N}[\text{tr}(\mathbf{w}^T \mathbf{w})] = \sigma_{\mathbf{w}}^2 (N + p) \end{aligned} \quad (3)$$

Then, we obtain that the **residual sum of squares** $\widehat{\text{SSE}}_{\text{emp}}$ returns a **biased estimate of MISE**, that is

$$E_{\mathbf{D}_N}[\widehat{\text{SSE}}_{\text{emp}}] = E_{\mathbf{D}_N}[\mathbf{e}^T \mathbf{e}] \neq \text{MISE} \quad (4)$$

As a consequence, if we replace the residual sum of squares with

$$\mathbf{e}^T \mathbf{e} + 2\sigma_{\mathbf{w}}^2 p \quad (5)$$

we obtain an unbiased estimator.

Nevertheless, this estimator requires an estimate of the noise variance.

The PSE and the FPE

Given an a priori estimate $\hat{\sigma}_{\mathbf{w}}^2$ we have the **Predicted Square Error (PSE)** criterion

$$PSE = \widehat{\text{SSE}}_{\text{emp}}(\hat{\beta}) + 2\hat{\sigma}_{\mathbf{w}}^2 p$$

where $p = n + 1$ is the total number of model parameters.

Taking as estimate of $\sigma_{\mathbf{w}}^2$

$$\hat{\sigma}_{\mathbf{w}}^2 = \frac{1}{N - p} \widehat{\text{SSE}}_{\text{emp}}(\hat{\beta})$$

we have the **Final Prediction Error (FPE)**

$$FPE = \frac{1 + p/N}{1 - p/N} \widehat{\text{SSE}}_{\text{emp}}(\hat{\beta})$$

Note that in order to have an estimate of the MSE error, it is sufficient to divide the above statistics by N .

The AIC criterion

- The Akaike Information Criterion (AIC) is commonly used to assess and select among a set of models on the basis of the likelihood.
- AIC was introduced by Akaike in 1973.
- He shows that the maximum log likelihood is a biased estimator of the mean expected log likelihood.
- In other terms the maximum log likelihood has a general tendency to overestimate the true value of the mean expected log likelihood. This tendency is more prominent for models with a larger number of free parameters.
- In other terms, if we choose the model with the largest maximum log likelihood, a model with an unnecessarily large number of free parameters is likely to be chosen.

The AIC criterion (II)

- Consider a training set D_N generated according to the parametric probability distribution $p_{\mathbf{z}}(z, \theta)$ where \mathbf{z} is a continuous random variable and $\theta \in \Theta \subset \mathbb{R}^n$.
- We assume that there are no constraints on the parameters, i.e., the number of free parameters in the model is n .
- The maximum likelihood approach consists in returning an estimate of θ on the basis of the training set D_N .
- The estimate is

$$\hat{\theta}_{\text{ml}} = \arg \max_{\theta \in \Theta} l_N(\theta)$$

where $l_N(\theta)$ is the empirical log-likelihood

$$l_N(\theta) = \frac{1}{N} \sum_{i=1}^N \ln g(z_i, \theta)$$

The AIC criterion (III)

- We define with $l(\hat{\theta}_{\text{ml}})$ the expected log-likelihood of the distribution $p_{\mathbf{z}}(\cdot, \hat{\theta}_{\text{ml}})$

$$l(\hat{\theta}_{\text{ml}}) = \int_{\mathcal{Z}} p(z) \ln p_{\mathbf{z}}(z, \hat{\theta}_{\text{ml}}) dz$$

- The quantity $l(\hat{\theta}_{\text{ml}})$ is negative and represents the accuracy of $p(z, \hat{\theta}_{\text{ml}})$ as estimator of $p_{\mathbf{z}}(z)$.
- Therefore, the larger the value of $l(\hat{\theta}_{\text{ml}})$ the better is the approximation of $p(z)$ returned by $p(z, \hat{\theta}_{\text{ml}})$.
- The maximum likelihood analogous of the mean integrated squared error is the *mean expected log-likelihood*

$$L_N = E_{\mathbf{D}_N} [l(\hat{\theta}_{\text{ml}})] = \int_{\mathcal{Z}^N} l(\hat{\theta}_{\text{ml}}(D_N)) dP^N(D_N)$$

that is the average over all possible realizations of a dataset of size N .

The AIC criterion (IV)

- The Akaike's criterion is based on the following assumption: *there exists a value $\theta^* \in \Theta$ so that the probability model $g(z, \theta^*)$ is equal to the underlying distribution $P(z)$.*
- Under this assumption he shows that the quantity

$$\text{AIC} = l_N(\hat{\theta}_{\text{ml}}) - \frac{n}{N} \quad (6)$$

is an unbiased estimate of the mean expected log-likelihood.

- A model which minimizes $-\text{AIC}$ is considered to be the most appropriate model.
- When there are several models whose values of maximum likelihood are about the same level, we should choose the one with the smallest number of free parameters.
- In this sense AIC realizes the so called *principle of parsimony*.

The non-linear case

- The above results hold only for the linear case.
- Note that linear case means that the **phenomenon underlying the data is linear**, not simply that the model is linear.
- In other terms, so far we assumed that the bias of the model was null. The corrective terms of FPE and PSE aimed at correcting the wrong estimation of the variance returned by the empirical error.
- How to measure MSE in a reliable way in the non-linear case starting from a finite dataset?
- The simplest approach is to extend the linear measures to the nonlinear case.

The C_p statistic

- Another estimate of the prediction error is the C_p **statistic**

$$C_p = \sum_{i=1}^N \frac{(y_i - \hat{y}_i)^2}{N} + \frac{2p\hat{\sigma}^2}{N}$$

where $\hat{\sigma}^2$ is an estimate of the residual variance.

- A reasonable choice for $\hat{\sigma}^2$ is $\sum_{i=1}^N \frac{(y_i - \hat{y}_i)^2}{N-p}$.
- Note that this statistic is the nonlinear version of the PSE criterion.
- It was derived for linear smoothers, a class of approximators which includes also linear models.
- An alternative is represented by validation methods.

Validation techniques

The most common techniques to return an estimate \hat{MSE} are

Testing: a *testing sequence* independent of D_N and distributed according to the same probability distribution is used to assess the quality. In practice, unfortunately, an additional set of input/output observations is rarely available.

Holdout: The *holdout* method, sometimes called test sample estimation, partitions the data D_N into two mutually exclusive subsets, the training set $D_{N_{tr}}$ and the holdout or test set $D_{N_{ts}}$ where $N = N_{tr} + N_{ts}$.

K -fold Cross-validation: the set D_N is randomly divided into K mutually exclusive test partitions of approximately equal size. The cases not found in each test partition are independently used for selecting the hypothesis which will be tested on the partition itself. The average error over all the k partitions is the cross-validated error rate.

The K -fold cross-validation

This is the algorithm in detail:

1. split the dataset D_N into k roughly equal-sized parts.
2. For the k th part $k = 1, \dots, K$, fit the model to the other $K - 1$ parts of the data, and calculate the prediction error of the fitted model when predicting the k -th part of the data.
3. Do the above for $k = 1, \dots, K$ and combine the K estimates of prediction error.

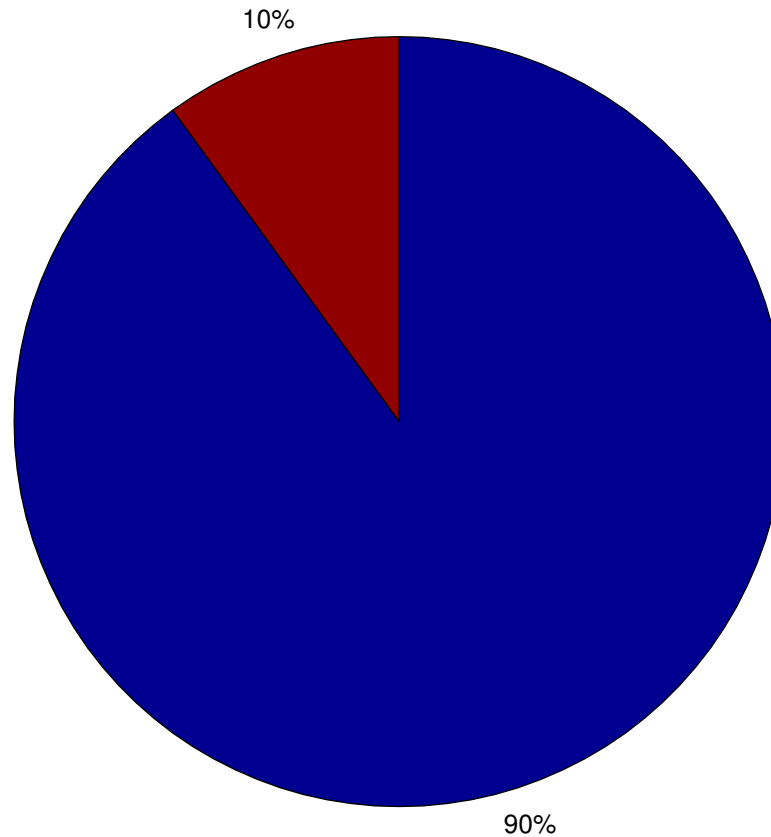
Let $k(i)$ be the part of D_N containing the i th sample. Then the cross-validation estimate of the MSE prediction error is

$$\widehat{\text{MSE}}_{\text{cv}} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i^{-k(i)})^2$$

where $\hat{y}_i^{-k(i)}$ denotes the fitted value for the i th observation returned by the model estimated with the $k(i)$ th part of the data removed.

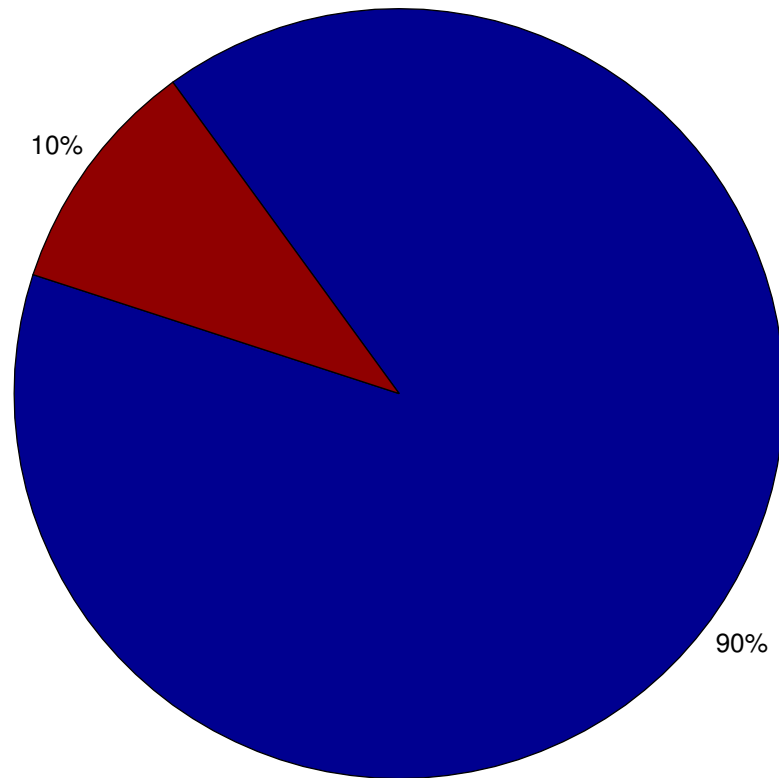
10-fold cross-validation

$K = 10$: at each iteration 90% of data are used for training and the remaining 10% for the test.



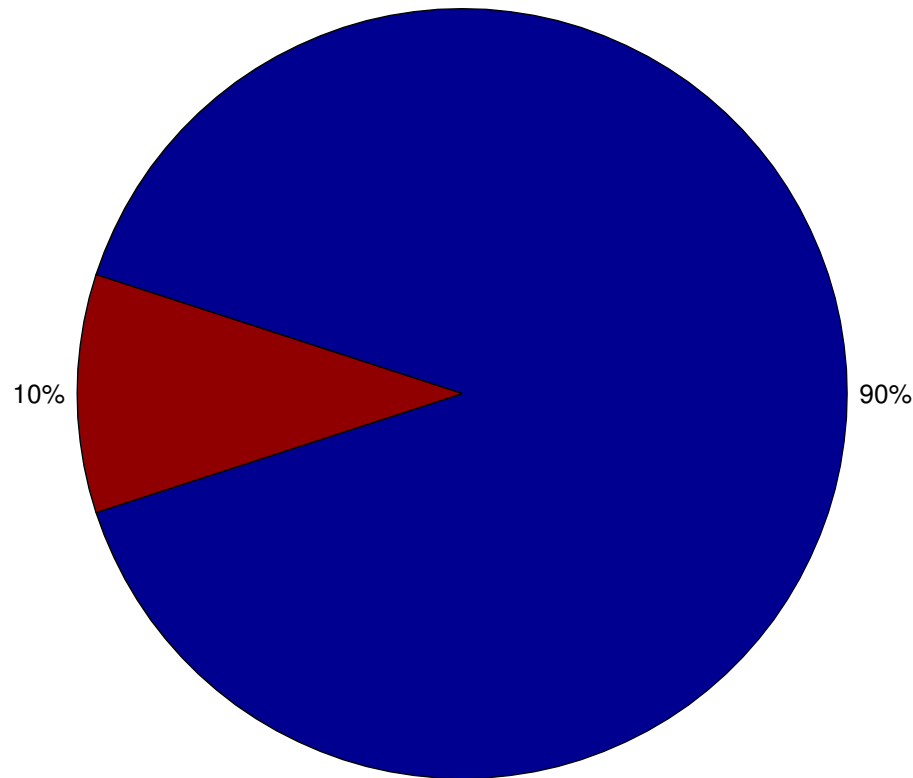
10-fold cross-validation

$K = 10$: at each iteration 90% of data are used for training and the remaining 10% for the test.



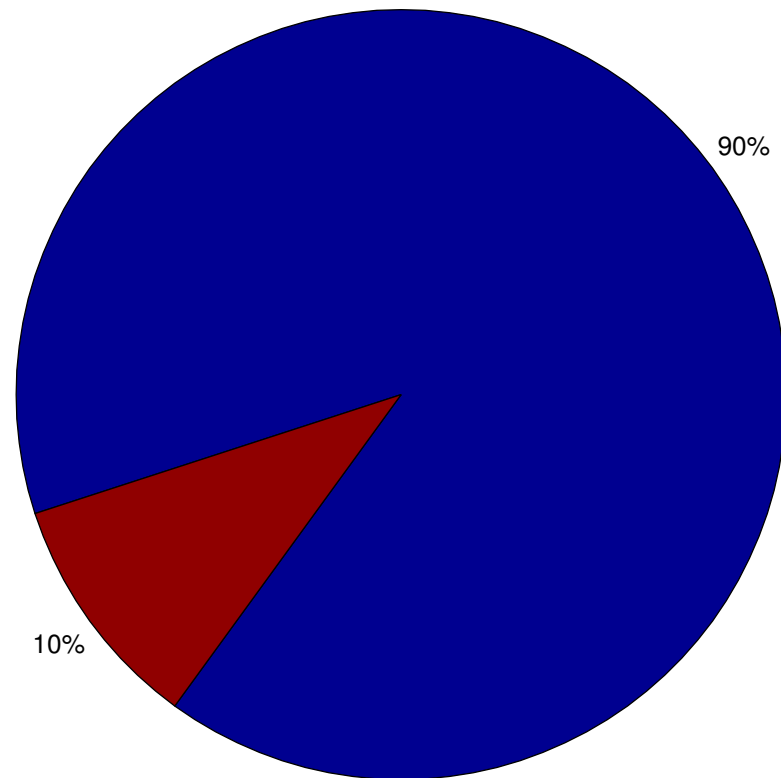
10-fold cross-validation

$K = 10$: at each iteration 90% of data are used for training and the remaining 10% for the test.



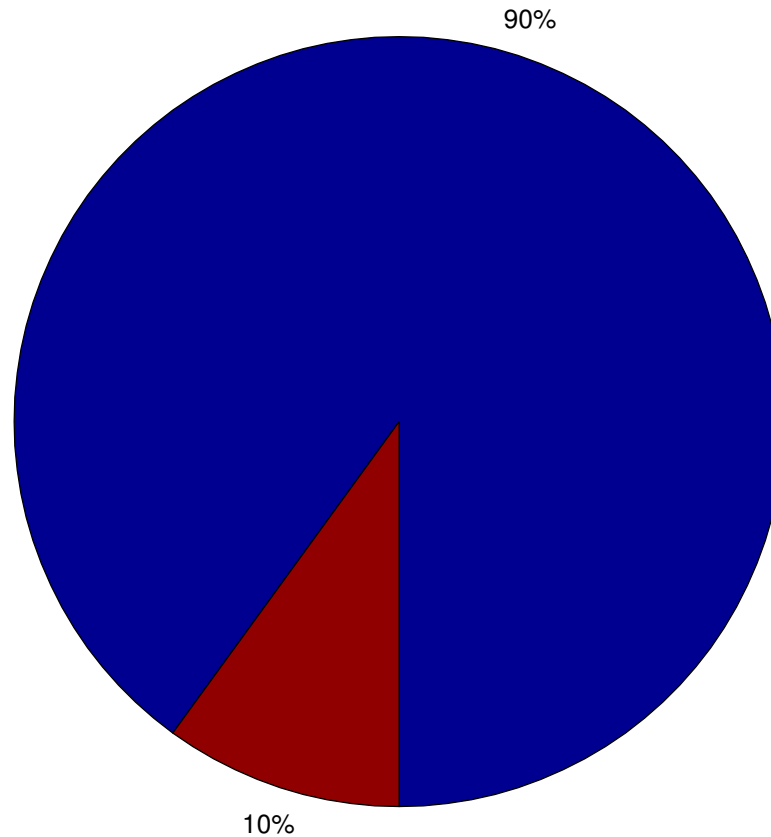
10-fold cross-validation

$K = 10$: at each iteration 90% of data are used for training and the remaining 10% for the test.



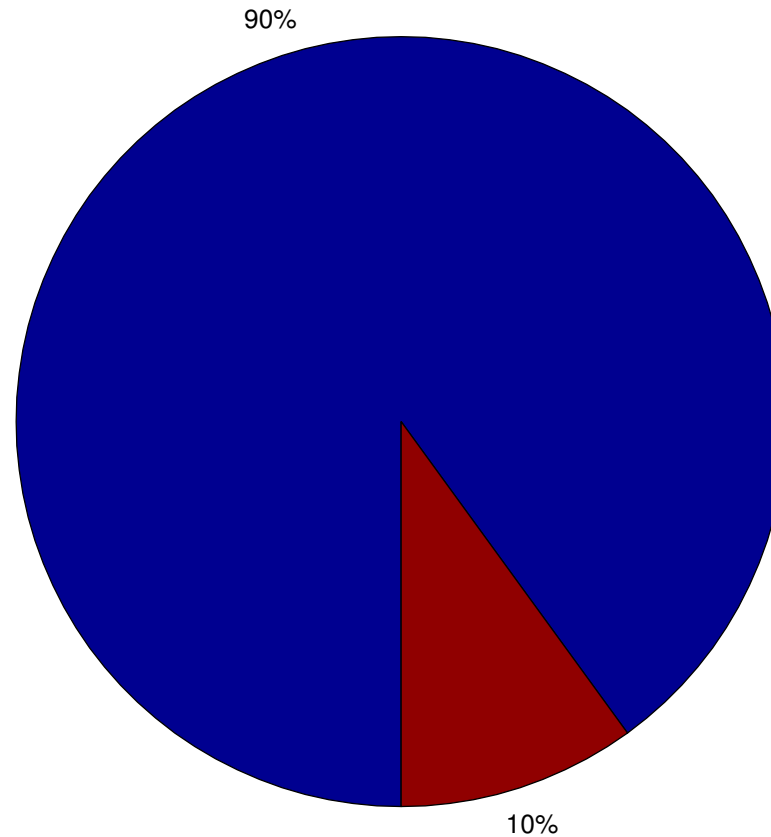
10-fold cross-validation

$K = 10$: at each iteration 90% of data are used for training and the remaining 10% for the test.



10-fold cross-validation

$K = 10$: at each iteration 90% of data are used for training and the remaining 10% for the test.



Leave-one-out cross validation

- The cross-validation algorithm where $K = N$ is also called the **leave-one-out** algorithm.
- This means that for each i th sample, $i = 1, \dots, N$,
 1. we carry out the parametric identification, leaving that observation out of the training set,
 2. we compute the predicted value for the i th observation, denoted by \hat{y}_i^{-i}

The corresponding estimate of the MSE prediction error is

$$\widehat{\text{MSE}}_{\text{loo}} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i^{-i})^2$$

R TP: An overfitting example

- Consider a dataset $D_N = \{x_i, y_i\}$, $i = 1, \dots, N$ where $N = 50$ and

$$\mathbf{x} \in \mathcal{N} \left([0, 0, 0], \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right)$$

is a 3-dimensional vector.

- Suppose that y is linked to \mathbf{x} by the input/output relation

$$y = x_1^2 + 4 \log(|x_2|) + 5x_3$$

where x_i is the i th component of the vector x .

- Consider as non-linear model a single-hidden-layer neural network (implemented by the R package `nnet`) with $s = 15$ hidden neurons.
- The number of neurons is an index of the complexity of the model.

- We want to estimate the prediction accuracy on a new i.i.d dataset of $N_{ts} = 50$ samples.
- Let us train the neural network on the whole training set. The empirical prediction MSE error is

$$\widehat{\text{MSE}}_{\text{emp}} = \frac{1}{N} \sum_{i=1}^N (y_i - h(x_i, \alpha_N))^2 = 1.6 * 10^{-6}$$

where α_N is obtained by the parametric identification step.

- However, if we test $h(\cdot, \alpha_N)$ on the test set we obtain

$$\widehat{\text{MSE}}_{ts} = \frac{1}{N_{ts}} \sum_{i=1}^{N_{ts}} (y_i - h(x_i, \alpha_N))^2 = 5.53$$

- This neural network is **seriously overfitting** the dataset.
- The empirical error is a **very bad** estimate of the MSE.

- We perform a K -fold cross-validation in order to have a better estimate of MSE.
- We put $K = 10$.
- Cross-validation implemented in the `cv.r` R file.
- The $K = 10$ cross-validated estimate of MSE is

$$\widehat{\text{MSE}}_{cv} = 7.00$$

This figure is a much more reliable estimation of the prediction accuracy.

- The leave-one-out estimate $K = N = 50$ is

$$\widehat{\text{MSE}}_{loo} = 6.60$$

- The cross-validated estimate could be used to select a better number of hidden neurons.

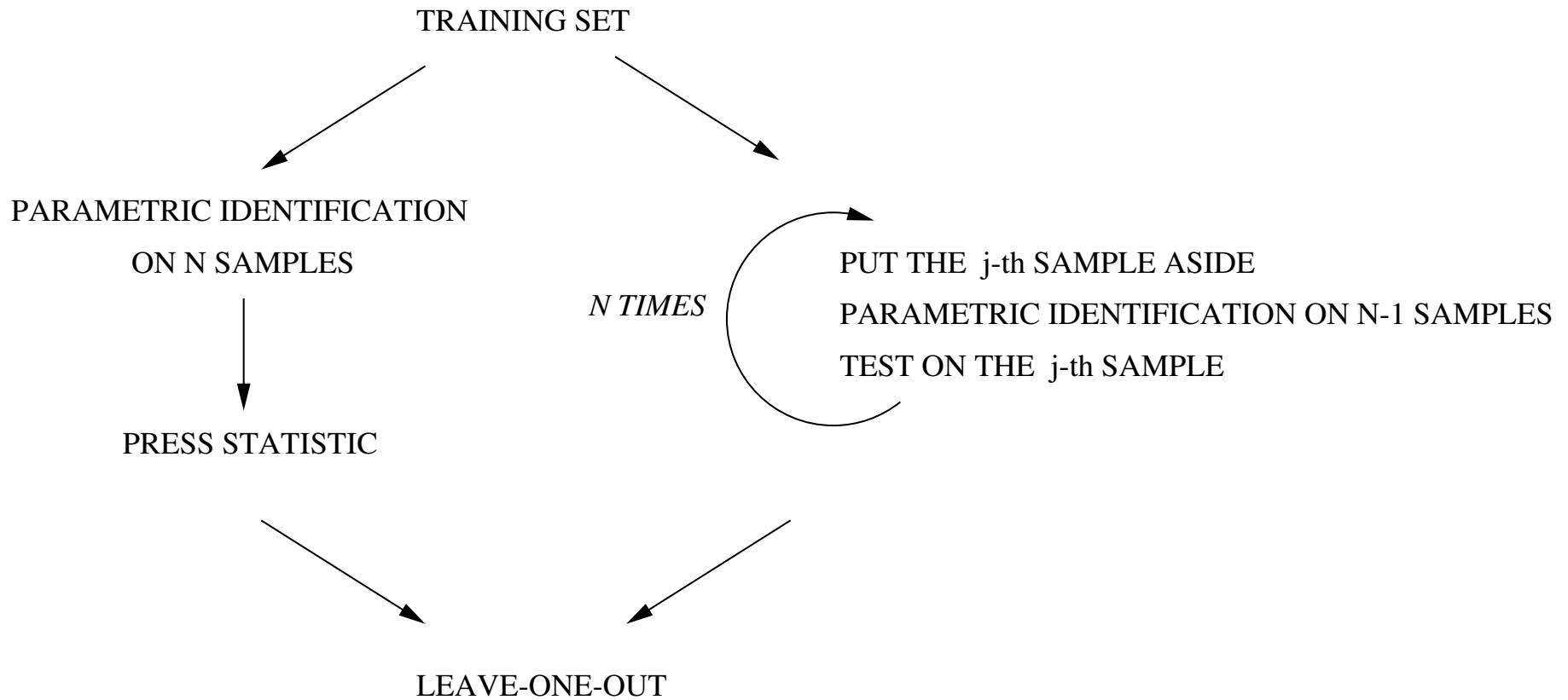
Model selection

- Model selection concerns the final choice of the model structure in the set that has been proposed by model generation and assessed by model validation.
- In real problems, this choice is typically a subjective issue and is often the result of a compromise between different factors, like the quantitative measures, the personal experience of the designer and the effort required to implement a particular model in practice.
- Here we will consider only quantitative criteria. A possible approach is the *winner-takes-all* approach where a set of S different architectures featuring different complexity are assessed and among them the best one is selected.

The cost of cross-validation

- Cross-validation requires the repetition of the parametric identification procedure for K times. In the leave-one-out case $K = N$.
- The procedure is highly time consuming for a generic non-linear approximator. Note that, for instance, in the case of a neural network, each parametric identification requires an expensive non-linear optimization procedure.
- In other words, the cost of leave-one-out for a generic approximator is $N * C$ where C is the cost of the identification.
- A very special case is represented by linear models. In this case the cost of a leave-one-out is comparable to C . The parametric identification procedure has as a by-product the leave-one-out residuals.

Leave-one-out for linear models



The PRESS statistic

- The PRESS (Prediction Sum of Squares) statistic is a simple formula which returns the leave-one-out (l-o-o) as a by-product of the parametric identification of $\hat{\beta}$
- The leave-one-out residual is

$$e_i^{\text{loo}} = y_i - \hat{y}_i^{-i} = y_i - x_i^T \hat{\beta}^{-i}$$

The PRESS procedure is made of the following steps

1. we use the whole training set to estimate the linear regression coefficients $\hat{\beta}$. This procedure is performed only once on the N samples and returns as by product the Hat matrix

$$H = X(X^T X)^{-1} X^T$$

2. we compute the residual vector e , whose i^{th} term is $e_i = y_i - x_i^T \hat{\beta}$,

3. we use the PRESS statistic to compute e_i^{loo} as

$$e_i^{\text{loo}} = \frac{e_i}{1 - H_{ii}}$$

where H_{ii} is the i^{th} diagonal term of the matrix H .

4. The resulting leave-one-out estimate of MSE is

$$\widehat{\text{MSE}}_{\text{loo}} = \frac{1}{N} \sum_{i=1}^N \left(e_i^{\text{loo}} \right)^2$$

Note that this is not an approximation but simply a faster way of computing the leave-one-out residual e_j^{loo} .

Cross-validation and other estimates

- Why use cross-validation if simpler estimates are available?
- The main reason is that these simple estimates are extrapolation of analogous figures for linear models. This extrapolation is doubtful for nonlinear data.
- The effective number of parameters p is not so easy to be determined. Theoretical criticism (Vapnik) vs. the simplest approach (count).
- The simple criteria requires the estimate of the variance $\hat{\sigma}^2$.
- The power of cross-validation comes from the reduced number of assumptions and its applicability to complex situations.

Bootstrap estimates of prediction error (1)

- The simplest bootstrap approach generates B bootstrap samples $D_{(b)}$, estimates the model $h(\cdot, \alpha_{(b)})$ on each of them, and then applies each fitted model to the original sample D_N to give B estimates

$$\widehat{\text{MSE}}^{(b)} = \sum_{i=1}^N (y_i - h(x_i, \alpha_{(b)}))^2$$

of the prediction error.

- The overall bootstrap estimate of prediction error is the average of these B estimates.

$$\widehat{\text{MSE}}_{\text{bs}} = \frac{1}{B} \sum_{b=1}^B \widehat{\text{MSE}}^{(b)} = \frac{1}{B} \sum_{b=1}^B \frac{1}{N} \sum_{i=1}^N (y_i - h(x_i, \alpha_{(b)}))^2$$

Bootstrap estimates of prediction error (2)

- This simple bootstrap approach turns out not to work very well. A second way to employ the bootstrap paradigm is to estimate the bias (or optimism) of the empirical risk.

$$\text{Bias}^{\widehat{\text{MSE}}_{\text{emp}}} = \text{MSE} - E_{D_N}[\widehat{\text{MSE}}_{\text{emp}}]$$

- The bootstrap estimate of this quantity is obtained by generating B bootstrap samples $D_{(b)}$ estimating the model $h(\cdot, \alpha_{(b)})$ for each of them and calculating the difference between the MSE on D_N and the empirical risk on $D_{(b)}$.

$$\text{Bias}_{\text{bs}}^{\widehat{\text{MSE}}_{\text{emp}}} = \frac{1}{B} \sum_{b=1}^B \widehat{\text{MSE}}^{(b)} - \frac{1}{B} \sum_{b=1}^B \widehat{\text{MSE}}_{\text{emp}}^{(b)}$$

- The final estimate is then

$$\widehat{\text{MSE}}_{\text{bs2}} = \widehat{\text{MSE}}_{\text{emp}} + \text{Bias}_{\text{bs}}^{\widehat{\text{MSE}}_{\text{emp}}}$$

R TP: bootstrap prediction error

- We apply the bootstrap procedure to the same regression problem assessed before by cross-validation.
- We apply the simple bootstrap procedure with $B = 10$.
- We obtain $\widehat{\text{MSE}}_{\text{bs}} = 4.066$.
- R-file `booterr.r`

R TP: bootstrap prediction error (2)

- Here we apply the bootstrap procedure to estimate the optimism of the empirical MSE.
- We put $B = 10$. These are the experimental results
- We denote by $\widehat{\text{MSE}}_{\text{emp}}^{(b)}$ the empirical MSE of the neural network trained on $D_{(b)}$. In other words, training and test set are in this case the same $D_{(b)}$.
- We denote by $\widehat{\text{MSE}}^{(b)}$ the MSE of the neural network trained on $D_{(b)}$ and tested on D_N .
- We compute the bootstrap estimate $\text{Bias}_{\text{bs}}^{\widehat{\text{MSE}}_{\text{emp}}}$ of the optimism.

- The final estimate is then

$$\widehat{\text{MSE}}_{\text{bs2}} = \widehat{\text{MSE}}_{\text{emp}} + \text{Bias}_{\text{bs}}^{\widehat{\text{MSE}}_{\text{emp}}}$$

In our case we have

b	$\widehat{\text{MSE}}_{\text{emp}}^{(b)}$	$\widehat{\text{MSE}}^{(b)}$
1	9.78e-07	4.00
2	1.71e-06	3.69
3	9.13e-07	3.66
4	7.25e-07	7.03
5	1.31e-06	3.19
6	1.45e-06	1.33
7	1.79e-06	4.62
8	1.10e-06	6.94
9	1.33e-06	2.20
10	1.95e-06	3.94

and then $\widehat{\text{MSE}}_{\text{bs}} = 1.62 * 10^{-6} + 4.066357 = 4.066358$.

Other bootstrap estimates

- Let us consider the simple bootstrap estimator $\widehat{\text{MSE}}_{\text{bs}}$. This is obtained by calculating the prediction error of $\alpha_{(b)}$ for each elements of D_N .
- One problem with this estimate is that we have points belonging to the training set $D_{(b)}$ and test set D_N . In particular it can be shown that the percentage of points belonging to both is 63.2%.
- In order to remedy to this problem, an idea could be to consider as test samples only the ones that do not belong to $D_{(b)}$.

Other bootstrap estimates (II)

- We will define by $\widehat{\text{MSE}}^0$ the MSE computed only on the samples that do not belong to $D_{(b)}$.

$$\widehat{\text{MSE}}^0 = \frac{1}{N} \sum_{i=1}^N \frac{1}{B_i} \sum_{b \in C_i} (y_i - h(x_i, \alpha_{(b)}))^2$$

where C_i is the set of indices of the bootstrap samples $D_{(b)}$ that do not contain i and B_i is the number of such bootstrap samples $D_{(b)}$.

The .632 estimator

- However, it can be shown that the samples used to compute $\widehat{\text{MSE}}^0$ are particularly hard test cases (too far from the training set) and that consequently $\widehat{\text{MSE}}^0$ is a pessimistic estimate of MSE.
- On the other side, the test cases used in $\widehat{\text{MSE}}_{\text{emp}}$ are too easy (too close to the test set) and consequently $\widehat{\text{MSE}}_{\text{emp}}$ is an optimistic estimate of MSE.
- A more reliable estimator is provided by the weighted average of the two quantities

$$\widehat{\text{MSE}}^{.632} = 0.368 * \widehat{\text{MSE}}_{\text{emp}} + 0.632 * \widehat{\text{MSE}}^0$$

Some final consideration

- Empirical error is a strong biased estimate of the prediction error and has order $O(1/N)$.
- It can be shown (Efron, 1983) that leave-one-out cross validation reduces the bias of the estimate from $O(1/N)$ to $O(1/N^2)$. However, it can have high variability particularly for N small.
- The simplest bootstrap $\widehat{\text{MSE}}_{\text{bs}}$ has large downward bias (too optimistic).
- In general, bootstrap methods has smaller variability but tends to have larger bias.
- Bootstrap methods are more informative if we want more than the simple estimate of the prediction error.
- Cross-validation behaves more like the bootstrap if the cost function is smooth (like the quadratic case).