Repeated Double Cross Validation for Estimation of Prediction Errors

Kurt Varmuza [1] and Peter Filzmoser [2]

Vienna University of Technology

[1] **Laboratory for Chemometrics, Institute of Chemical Engineering, Vienna University of Technology**
Getreidemarkt 9/166,
A-1060 Vienna, Austria
kvarmuza@email.tuwien.ac.at,
www.lcm.tuwien.ac.at

[2] **Institute of Statistics and Probability Theory, Vienna University of Technology**
Wiedner Hauptstrasse 8-10,
A-1040 Vienna, Austria
P.Filzmoser@tuwien.ac.at,
www.statistik.tuwien.ac.at/public/filz

Poster Presentation: 4th International Symposium on Computer Applications and Chemometrics in Analytical Chemistry - SCAC 2008
September 1–5, 2008, Balatonalmádi, Hungary

Introduction

The performance of multivariate regression models

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_m x_m$$

obtained from a data set $\boldsymbol{X}(n \times m)$ and $\boldsymbol{y}(n \times 1)$ can be estimated from

- a reasonable large number (z) of prediction errors (residuals) $\hat{y}_i - y_i$ ($i = 1 \ldots z$),
- obtained from objects not used in model development and model optimization (test sets).

For data sets with a rather small number of objects, a single random split into a calibration set and a test set may give very misleading results.

Much better approaches are

- **repeated double cross validation (RDCV)** (used in this contribution), or
- bootstrap.

RDCV is used here

- to estimate the optimum complexity of linear regression models (number of PLS components),
- to estimate the prediction errors to be expected for new objects - using models that are derived from the considered data set.

Repeated double cross validation (RDCV)

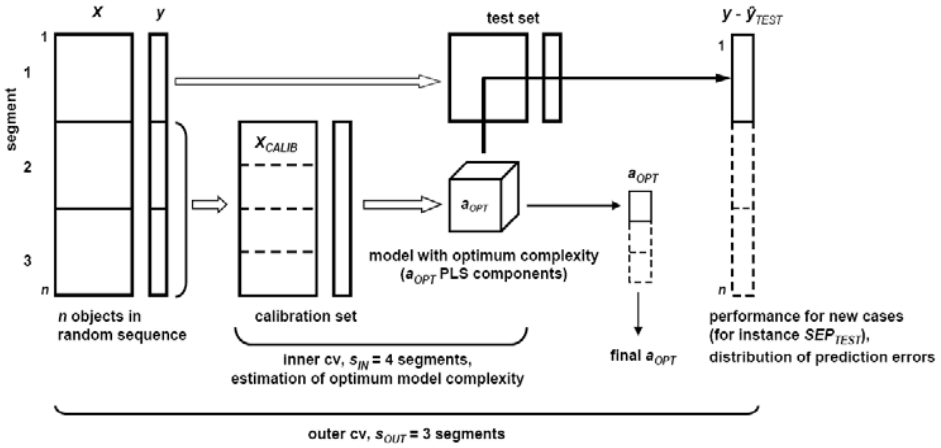
RDCV applies cross validation in three nested loops:

In an outer loop the available n objects are randomly split into a test set and a calibration set.

In an inner loop cross validation is applied to the calibration set to find the optimum number of PLS components, a_{OPT}. A model with a_{OPT} components is then calculated from all data of the calibration set and is applied to the test set giving **test-set-predicted** values $\hat{y}$ for the current test set.

After completing the outer loop, for each of the n objects a test-set-predicted value $\hat{y}$ is available.

This procedure is repeated k times, giving $z = k.n$ values $\hat{y}_i$ and z residuals $\hat{y}_i - y_i$. Each object has been used k times in a test set.



Repeated double cross validation (RDCV)

FOR *rep* = 1 TO *k* (number of repetitions) — repetition loop

- ❍ Split all *n* objects randomly into s_{OUT} segments (typ. 3-7)
- ❍ **FOR** *loop_out* = 1 TO s_{OUT} — outer loop
  - ☐ **test set** = segment with number *loop_out* (n_T objects)
    calibration set = other s_{OUT} - 1 segments (n_C objects)
  - ☐ Split calibration set into s_{IN} segments (typ. 3-7)
  - ☐ **FOR** *loop_in* = 1 TO s_{IN} — inner loop
    - ■ **validation set** = segment with number *loop_in*
      training set = other s_{IN} - 1 segments
    - ■ Make PLS models from the training set, with a = 1, 2, ..., a_{MAX} components
    - ■ Apply the PLS models to the validation set: giving $\hat{y}_{CV}$ for the segment *loop_in*, for a = 1, 2, ...

    NEXT *loop_in*
  - ☐ Estimate optimum number of components from $\hat{y}_{CV,j}$ (j = 1 ... n_C), giving a_{OPT}(*loop_out*) for this outer loop
  - ☐ Make a PLS model for the whole calibration set using a_{OPT}(*loop_out*) components
  - ☐ Apply the model to the test set: giving test-set-predicted $\hat{y}_{TEST}$ for n_T test set objects

  NEXT *loop_out*
- ❍ After completing the outer loop: we have one test-set-predicted $\hat{y}_{TEST}$ for each of the *n* objects

NEXT *rep*

RDCV is freely available by the function **mvr_dcv** in the new package **chemometrics** for the R programming system [1, 2].

RDCV yields

- $w = k \,.\, s_{OUT}$ values for the optimum no. of components
- $z = k.n$ test-set-predicted values $\hat{y}_i$ ($i = 1 \ldots z$)

k	number of repetitions
n	number of objects
s_{OUT}	number of segments in outer loop

A **final optimum number of PLS components**, a_{OPT}, can be estimated from the RDCV results e.g. as follows.

- a_{OPT} is the number of components most often obtained in the $k \,.\, s_{OUT}$ estimations (see application).
- Depending on the shape of the frequency distribution, more than one value for a_{OPT} should be considered.

A final model is calculated from all *n* objects using the final optimum number of components.

The **prediction performance** of the final model can be estimated from the RDCV results as follows.

- SEP_{TEST} is the standard deviation of all z residuals; *SEP* is often called "standard error of prediction".
- Results from the k repetitions give k values $SEP_{TEST}(rep)$, characterizing the variability of SEP_{TEST} .
- The distribution of all z residuals gives a good picture of the prediction errors to be expected for new cases.

  E.g. the quantiles at 0.025 and 0.975 define a 95% tolerance interval; for a (usually) normal distribution of the residuals it is approximately given by $\pm$ 2 SEP_{TEST}

QSPR example

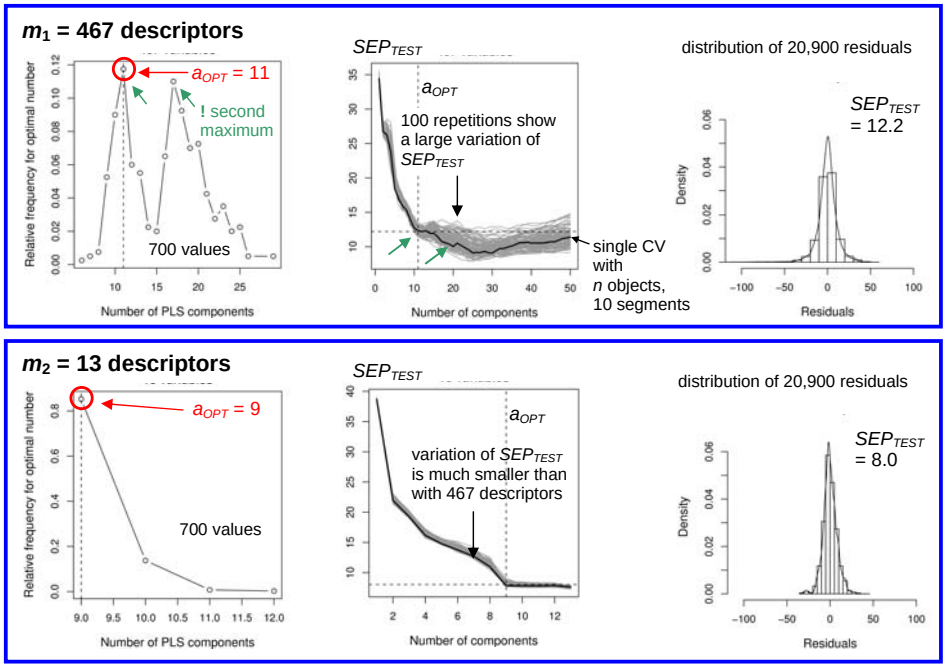
n = **209 polycyclic aromatic compounds**, 3D, all H-atoms; *Corina* [3]

y **gas-chromatographic retention indices**, *Lee* indices [4]

X m_1 = 467 molecular descriptors; *Dragon* [5]

m_2 = 13 descriptors selected by a genetic algorithm; *MobyDigs* [6]

RDCV: 7 and 4 segments in outer and inner loop, resp.; k = 100 repetitions



A single cross validation may give very misleading results. Repeated double cross validation (or bootstrap) is recommended.

References

[1] Varmuza K, Filzmoser P, Introduction to multivariate statistical analysis in chemometrics, CRC Press, Boca Raton, FL, USA, in print (2009).

[2] Our R function "mvr_dcv" uses a PLS package, described by Mevik B.H. and Wehrens R., J. Stat. Software 18 (2007) issue 2, 1-24.

[3] Corina software, Molecular Networks GmbH Computerchemie, www.mol-net.de, Erlangen, Germany (2004).

[4] Lee M.L., et al., Anal. Chem. 51 (1979) 768-773.

[5] Dragon software, 5.0, Talete srl, www.talete.mi.it, Milan, Italy (2004).

[6] MobyDigs software, 1.0. Talete srl, www.talete.mi.it, Milan, Italy (2004)