Partial Least Squares

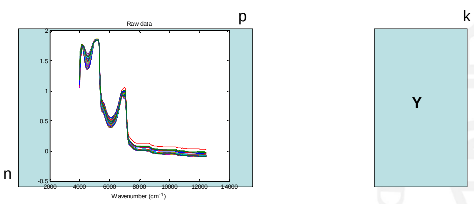
A tutorial

Lutgarde Buydens

Partial least Squares

- Multivariate regression
  - Multiple Linear Regression (MLR)
  - Principal Component Regression (PCR)
  - Partial Least Squares (PLS)
- Validation
- Preprocessing

Multivariate Regression



Rows: Cases, observations, …

Collums: Variables, Classes, tags

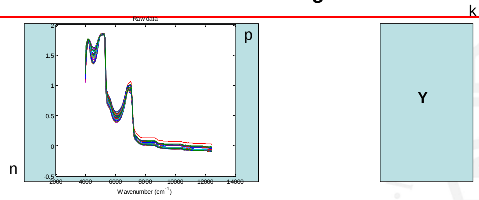
Analytical observations of different samples
Experimental runs
Persons
….

P: Spectral variables
Analytical measurements

K: Class information
Concentration,..

X: Independent variabels (will be always available)
Y: Dependent variables (to be predicted later from X)

Multivariate Regression



Rows: Cases, observations …

Collums: Variables, Classes, tags

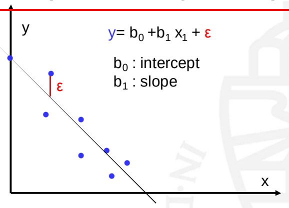
X: Independent variabels (will be always available)
Y: Dependent variables (to be predicted later from X)

Y = f(X) : Predict Y from X

MLR: Multiple Linear Regression
PCR: Principal Component Regression
PLS: Partial Least Sqaures

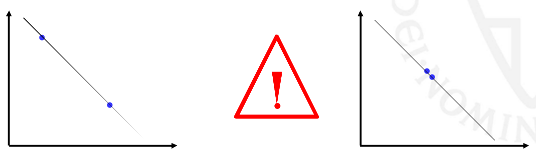
From univariate to Multiple Linear Regression (MLR)

Least squares regression



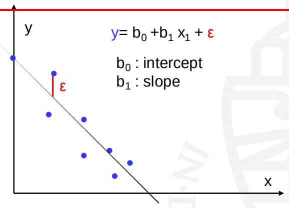
$y = b_0 + b_1 x_1 + \varepsilon$

b_0 : intercept
b_1 : slope



MLR: Multiple Linear Regression

Least squares regression



$y = b_0 + b_1 x_1 + \varepsilon$

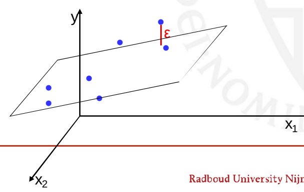
b_0 : intercept
b_1 : slope

Multiple Linear Regression

$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots b_p x_p + \varepsilon$

$$\mathbf{Y} = \hat{\mathbf{Y}} + \mathbf{E}$$

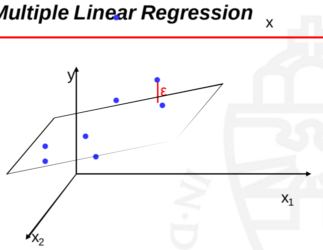
maximizes $\mathbf{r}(\mathbf{y}, \hat{\mathbf{y}})$



MLR: Multiple Linear Regression

$y = b_0 + b_1 x_1 + b_2 x_2 + \dots b_p x_p + \varepsilon$

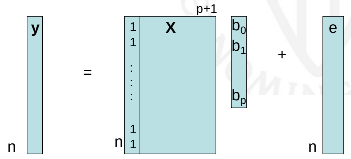
$$\mathbf{Y} = \hat{\mathbf{Y}} + \mathbf{E}$$



$y_{n1} = X_{np} b_{p1} + e_{n1}$

$Y_{nk} = X_{np} B_{pk} + E_{nk}$

$b = (X^T X)^{-1} X^T y$

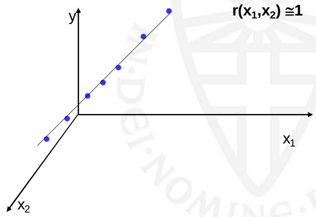


MLR: Multiple Linear Regression

•*Disadavantages:* $(X^T X)^{-1}$

• Uncorrelated X-variables required

• n ≥ p +1

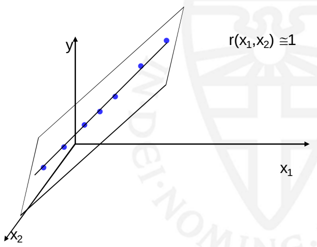


MLR: Multiple Linear Regression

Disadavantages: $(X^T X)^{-1}$

• Uncorrelated X-variables required

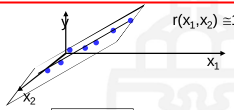
Fits a plane through a line !!



MLR: Multiple Linear Regression

Disadavantages: $(X^T X)^{-1}$

• Uncorrelated X-variables required



Set A x_1	Set A x_2	Set B x_1	Set B x_2	y
-1.01	-0.99	-1.01	-0.99	-1.89
3.23	3.25	3.23	3.25	10.33
5.49	5.55	5.49	5.55	19.09
0.23	0.21	0.23	0.23	2.19
-2.87	-2.91	-2.87	-2.91	-8.09
3.67	3.76	3.67	3.76	11.29

$y = b_1 x_1 + b_2 x_2 + \varepsilon$

	b_1	b_2	b_1	b_2
MLR	10.3	-6.92	2.96	0.28

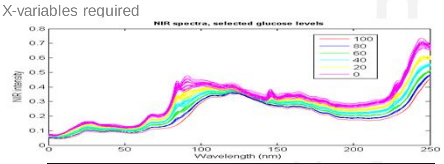
R^2 =0.98 R^2 =0.98

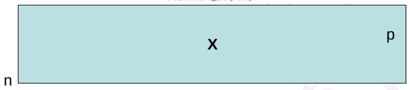
MLR: Multiple Linear Regression

Disadavantages: $(X^T X)^{-1}$

• Uncorrelated X-variables required

• n ≥ p +1





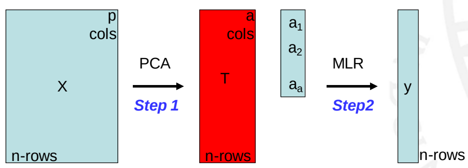
→ **Dimension reduction** • **Variable Selection**

• **Latent variables (PCR, PLS)**

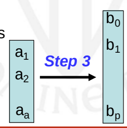
PCR: Principal Component Regression

Step 1: Perform PCA on the original **X**

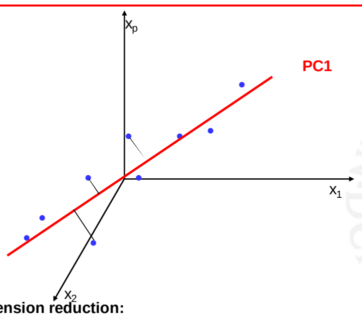
Step 2 : Use the orthogonal PC-scores as independent variables in a MLR model



Step 3: Calculate **b-coefficients** from the **a-coefficients**



PCR: Principal Component Regression



Dimension reduction:

Use scores (projections) on latent variables that explain *maximal variance in X*

IMM — Radboud University Nijmegen

PCR: Principal Component Regression

Step 0 : Meancenter X

Step 1: Perform PCA: **X = TP^T ⇒ X* = (TP^T)***

Step 2: Perform MLR **Y=TA**

A = (T^{T}T)$^{-1}$T^{T}Y

Step 3 : Calculate B **Y = X* B**

Y = (T(P^T) B MLR on reconstructed **X*= (TP^T)***

A = P^{T}B

B = (PP^T)$^{-1}$PA

B = PA

Calculate b_0's $\mathbf{b}_0 = \bar{\mathbf{y}} - \bar{\hat{\mathbf{y}}}$

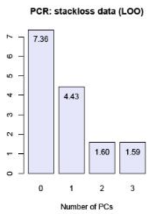
IMM — Radboud University Nijmegen

PCR: Principal Component Regression

Optimal number of PC's

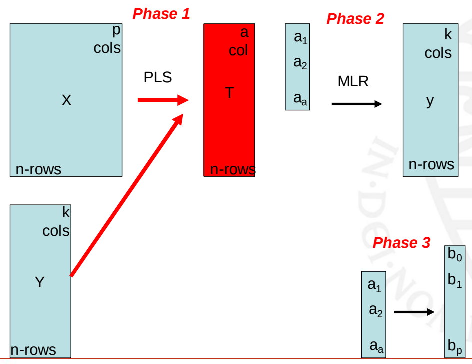
Calculate Crossvalidation RMSE for different # PC's

$$RMSECV = \sqrt{\sum (\hat{y}_i - y_i)^2 / n}$$



IMM — Radboud University Nijmegen

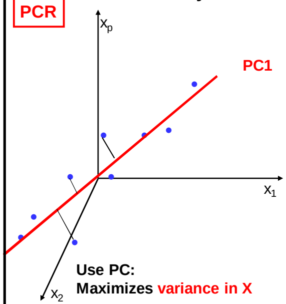
PLS: Partial Least Squares Regression



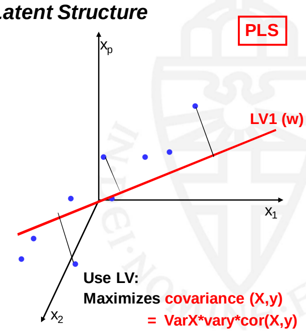
IMM — Radboud University Nijmegen

PLS: Partial Least Squares Regression

Projection to Latent Structure



Use PC:
Maximizes variance in X



Use LV:
Maximizes covariance (X,y)
= VarX*vary*cor(X,y)

IMM — Radboud University Nijmegen

PLS: Partial Least Squares Regression

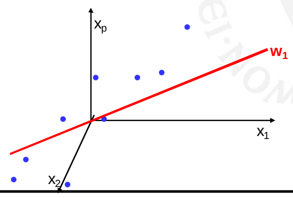
Phase 1 : Calculate new independent variables (T)

Sequential Algorithm: Latent variables and their scores are calculated sequentially

- **Step 0: Mean center X**
- **Step 1: Calculate w**

Calculate LV1= w_1 that maximizes Covariance (**X,Y**) : SVD on **X^{T}Y**

$(X^TY)_{pk} = W_{pa}D_{aa}Z^T_{ak}$ **w_1 = 1st col. of W**



IMM — Radboud University Nijmegen

PLS: Partial Least Squares Regression

Phase 1 : Calculate new independent variables (T)

Sequential Algorithm: Latent variables and their scores are calculated sequentially

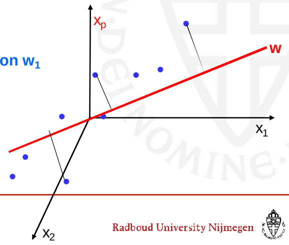
- *Step 1*: Calculate LV1= $\mathbf{w}_1$ that maximizes Covariance (**X**,**Y**) : SVD on $\mathbf{X}^T\mathbf{Y}$

$(X^TY)_{pk} = W_{pa}D_{aa}Z^T_{ak}$ $\quad w_1 = 1^{st}$ col. of W

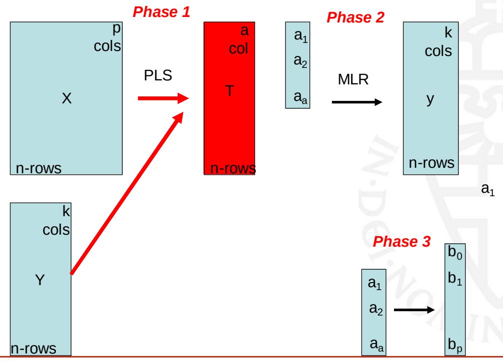
•**Step 2:**

Calculate t_1, scores (projections) of X on w_1

$t_{n1} = X_{np}w_{p1}$



PLS: Partial Least Squares Regression

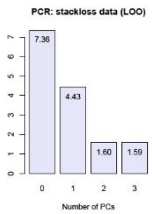


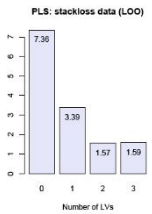
PLS: Partial Least Squares Regression

Optimal number of LV's

Calculate Crossvalidation RMSE for different # LV's

$$RMSECV = \sqrt{\sum (\hat{y}_i - y_i)^2 / n}$$





MLR, PCR, PLS:

Set A x_1	Set A x_2	Set B x_1	Set B x_2	y
-1.01	-0.99	-1.01	-0.99	-1.89
3.23	3.25	3.23	3.25	10.33
5.49	5.55	5.49	5.55	19.09
0.23	0.21	0.23	0.23	2.19
-2.87	-2.91	-2.87	-2.91	-8.09
3.67	3.76	3.67	3.76	11.29

$y = b_1 x_1 + b_2 x_2 + \varepsilon$

	b_1	b_2	b_1	b_2
MLR	10.3	-6.92	2.96	0.28
PCR	1.60	1.62	1.60	1.62
PLS	1.60	1.62	1.60	1.62

VALIDATION

Estimating prediction error.

Basic Principle:

test how well your model works with **new** data,

it has not seen yet!

Common measure for prediction error

Root Mean Square Error:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

$\hat{y}_i$: prediction for sample i
y_i: true value of sample i
n: number of samples

A Biased Approach

Prediction error of the samples the model was built on

Error is biased!

Samples also used to build the model

→ model is biased towards accurate prediction of these specific samples

Validation: Basic Principle

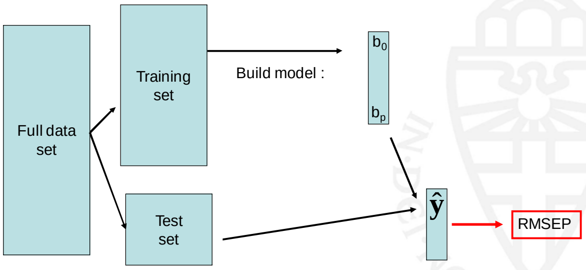
Basic Principle:

test how well your model works with **new** data, it has not seen yet!

Split data in **training** and **test** set.

Several ways:
One large test set
Leave one out and repeat: LOO
Leave *n* objects out and repeat: LNO
. . .
Apply entire model procedure on the test set

Validation

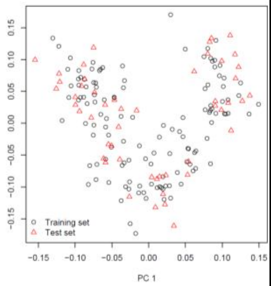


Remark: for final model use whole data set.

Training and test sets

Split in **training** and **test** set.

- Test set should be representative of training set
- **Random** choice is often the best
- Check for extremely unlucky divisions
- Apply whole procedure on the test and validation sets



Cross-validation

segment 1	segment 1	segment 1	segment 1	segment 1
segment 2	**segment 2**	segment 2	segment 2	segment 2
segment 3	segment 3	**segment 3**	segment 3	segment 3
segment 4	segment 4	segment 4	**segment 4**	segment 4
segment 5	segment 5	segment 5	segment 5	**segment 5**

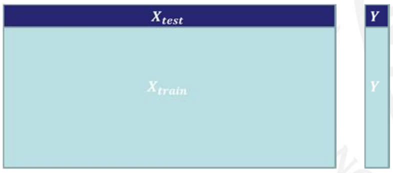
- Most simple case: Leave-One-Out (=LOO, segment=1 sample). Normally 10-20% out (=L*n*O).
- Remark: for final model use whole data set.

Cross-validation: an example

- The data

Cross-validation: an example

- Split data into *training set* and *validation set*



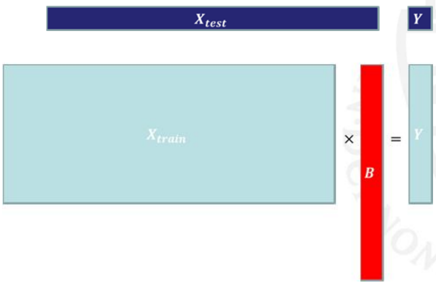
Cross-validation: an example

- Split data into *training set* and *test set*

Radboud University Nijmegen

Cross-validation: an example

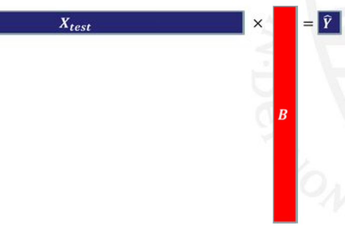
- Build a model on the training set



Radboud University Nijmegen

Cross-validation: an example

- Check prediction of Y'

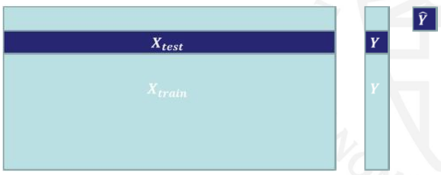


- Save $\hat{Y}$

Radboud University Nijmegen

Cross-validation: an example

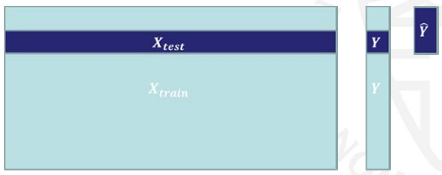
- Split data again into *training set* and *valid. set*
  - Until all samples have been in the validation set once
  - Common: Leave-One-Out (LOO)



Radboud University Nijmegen

Cross-validation: an example

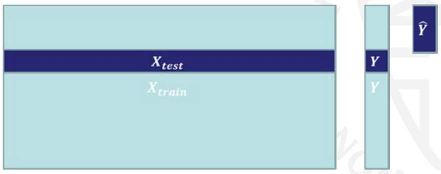
- Split data again into *training set* and *valid. set*
  - Until all samples have been in the validation set once
  - Common: Leave-One-Out (LOO)



Radboud University Nijmegen

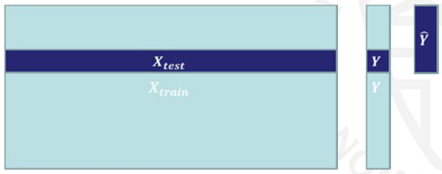
Cross-validation: an example

- Split data again into *training set* and *valid. set*
  - Until all samples have been in the validation set once
  - Common: Leave-One-Out (LOO)



Cross-validation: an example

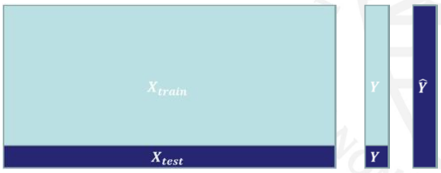
- Split data again into *training set* and *valid. set*
  - Until all samples have been in the validation set once
  - Common: Leave-One-Out (LOO)



Radboud University Nijmegen

Cross-validation: an example

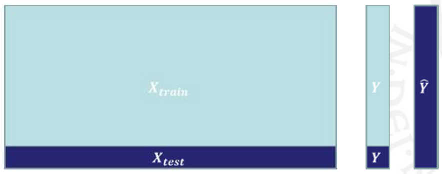
- Split data again into *training set* and *valid. set*
  - Until all samples have been in the validation set once
  - Common: Leave-One-Out (LOO)



Radboud University Nijmegen

Cross-validation: an example

- Split data again into *training set* and *valid. set*
  - Until all samples have been in the validation set once
  - Common: Leave-One-Out (LOO)



$$RMSECV = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

Radboud University Nijmegen

Cross-validation: a warning

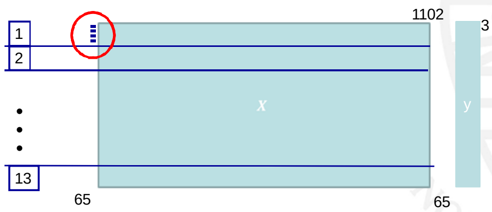
- Data: 13 x 5 = 65 NIR spectra (1102 wavelengths)
  - 13 samples: different composition of NaOH, NaOCl and Na_2CO_3
  - 5 temperatures: each sample measured at 5 temperatures

Composition	NaOH (wt%)	NaOCl (wt%)	Na_2CO_3 (wt%)	Temperature (° C)				
1	18.99	0	0	15	21	27	34	40
2	9.15	9.99	0.15	15	21	27	34	40
3	15.01	0	4.01	15	21	27	34	40
4	9.34	5.96	3.97	15	21	27	34	40
…	…	…	…	…				
13	16.02	2.01	1.00	15	21	27	34	40

Radboud University Nijmegen

Cross-validation: a warning

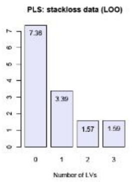
- The data



Leave SAMPLE out

Radboud University Nijmegen

Selection of number of LV's



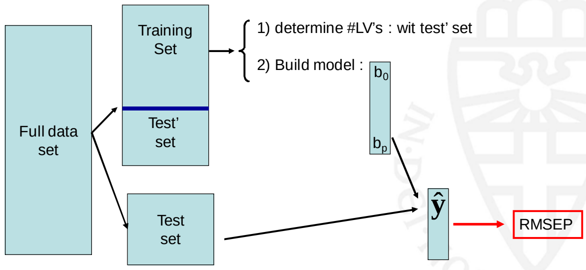
Trough Validation:

Choose number of LV's that results in model with lowest prediction errror

Testset to assess final model cannot be used !

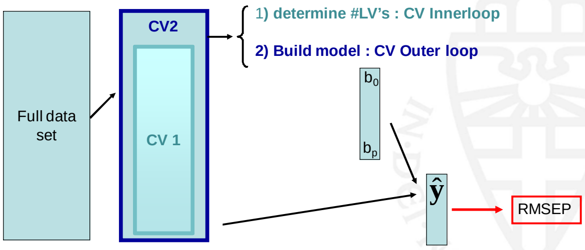
Divide trainingset
Crossvalidation

Validation



Remark: for final model use whole data set.

Double Cross Validation



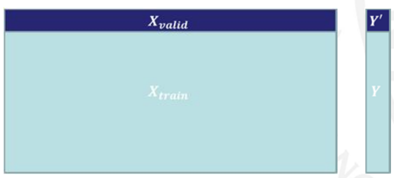
Remark: for final model use whole data set Skip.

Double cross-validation

- The data

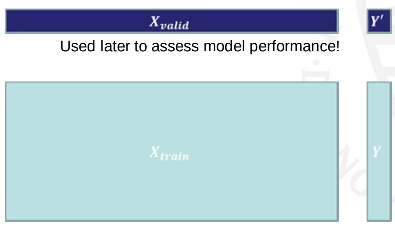
Double cross-validation

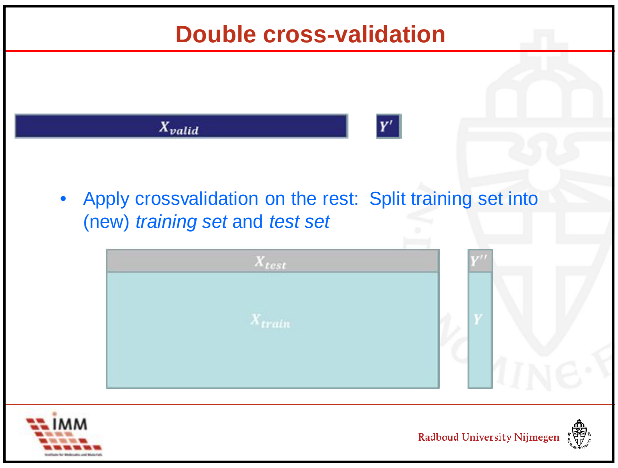
- Split data into *training set* and *validation set*

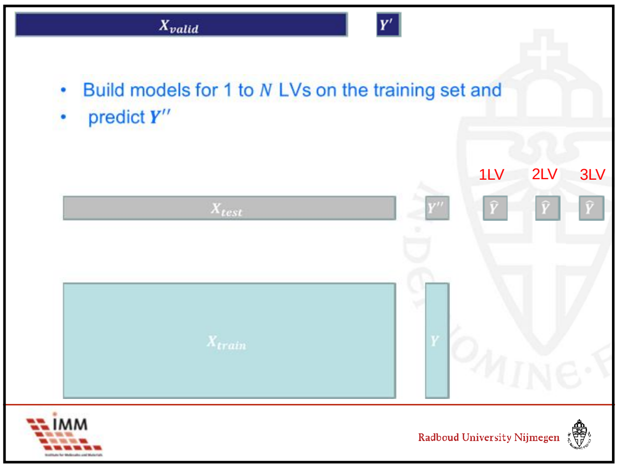


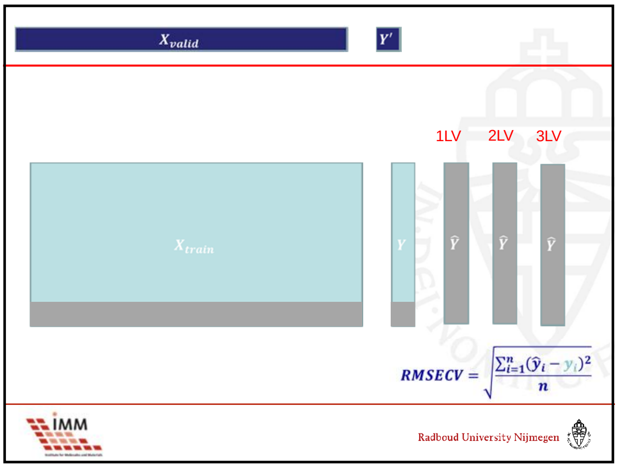
Double cross-validation

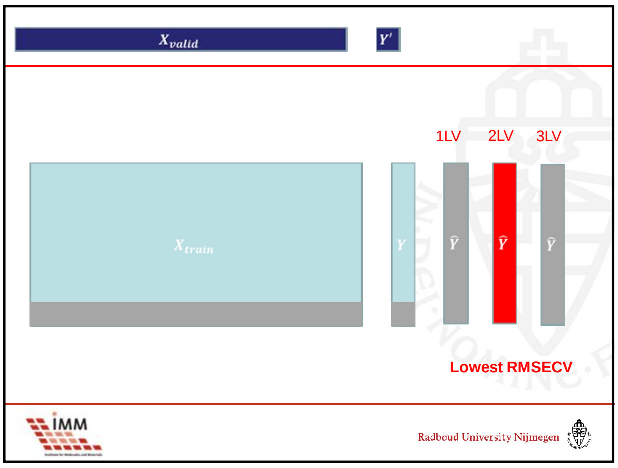
- Split data into *training set* and *validation set*

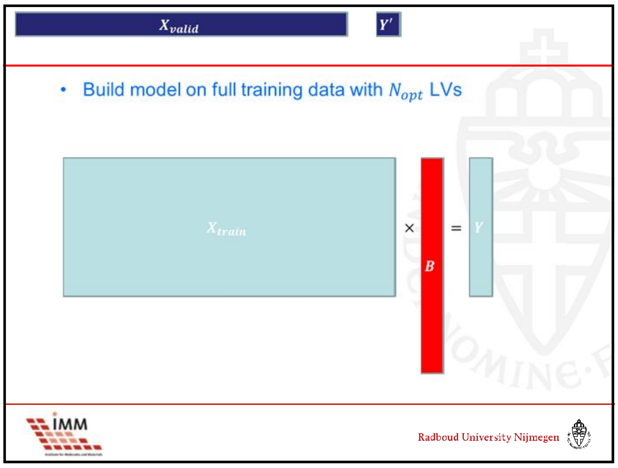


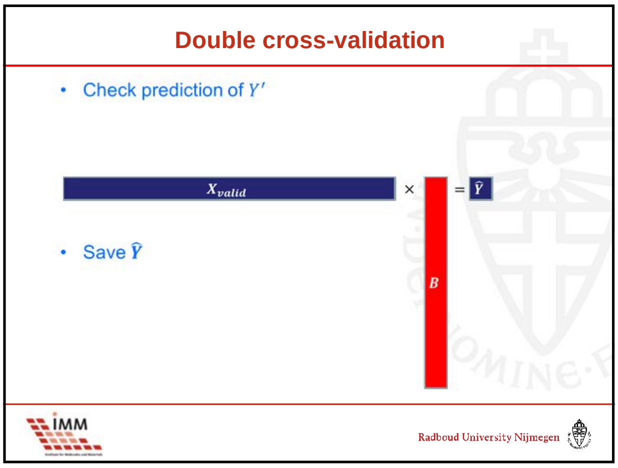

Double cross-validation
X_{valid}
Y'
Apply crossvalidation on the rest: Split training set into (new) *training set* and *test set*
X_{test}
Y''
X_{train}
Y
IMM
Radboud University Nijmegen


X_{valid}
Y'
Build models for 1 to N LVs on the training set and
predict Y''
1LV
2LV
3LV
X_{test}
Y''
$\hat{Y}$
$\hat{Y}$
$\hat{Y}$
X_{train}
Y
IMM
Radboud University Nijmegen


X_{valid}
Y'
1LV
2LV
3LV
X_{train}
Y
$\hat{Y}$
$\hat{Y}$
$\hat{Y}$
$RMSECV = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$
IMM
Radboud University Nijmegen

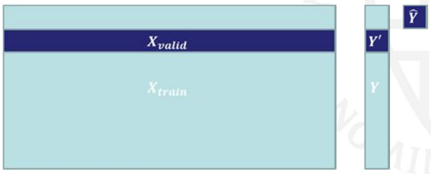

X_{valid}
Y'
1LV
2LV
3LV
X_{train}
Y
$\hat{Y}$
$\hat{Y}$
$\hat{Y}$
Lowest RMSECV
IMM
Radboud University Nijmegen


X_{valid}
Y'
Build model on full training data with N_{opt} LVs
X_{train}
×
B
=
Y
IMM
Radboud University Nijmegen


Double cross-validation
Check prediction of Y'
X_{valid}
×
B
=
$\hat{Y}$
Save $\hat{Y}$
IMM
Radboud University Nijmegen

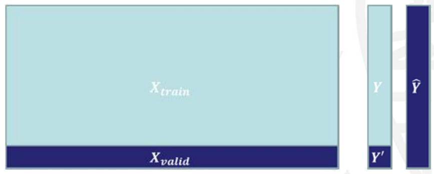
Cross-validation: an example

- Repeat procedure
  - Until all samples have been in the validation set once



Cross-validation: an example

- Repeat procedure
  - Until all samples have been in the validation set once



$$RMSEP = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

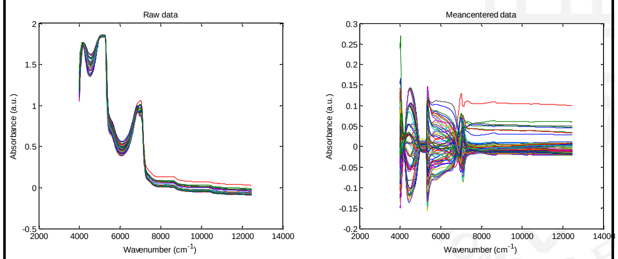
Double cross-validation

- In this way:
  - The number of LVs is determined by using samples not used to build the model with
  - The prediction error is also determined using samples the model has not seen before

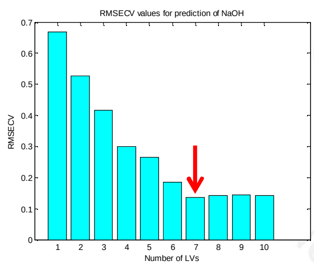
Remark: for final model use whole data set.

PLS: an example

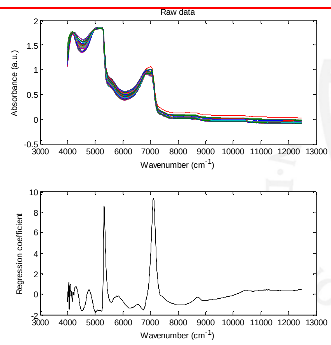
Raw + meancentered data



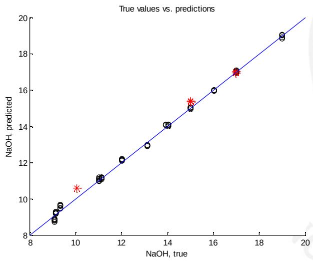
RMSECV vs. No of LVs



Regression coeffficients



True vs. predicted

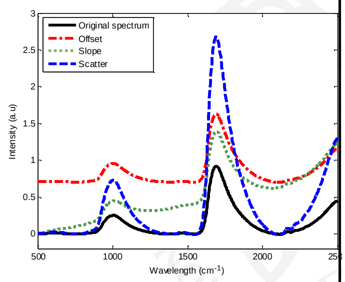


Radboud University Nijmegen

Why Pre-Processing ?

Data Artefacts

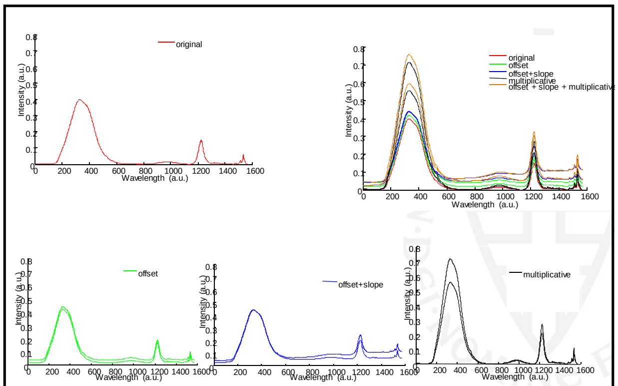
- Baseline correction
- Alignment
- Scatter correction
- Noise removal
- Scaling, Normalisation
- Transformation
- .....



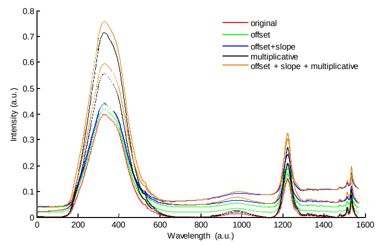
Other

- Missing values
- Outliers

Radboud University Nijmegen



Radboud University Nijmegen



Radboud University Nijmegen

Pre-Processing Methods

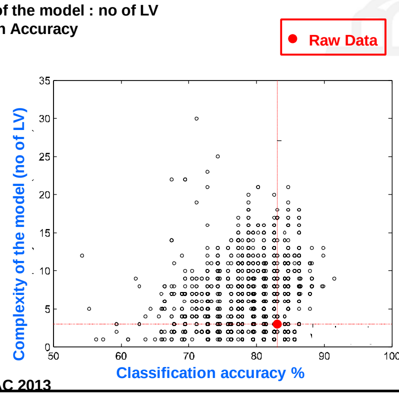
4914 combinations: all reasonable

STEP 1: (7x) BASELINE	STEP 2: (10x) SCATTER	STEP 3: (10x) NOISE	STEP 4: (7x) SCALING & TRANSFORMATION
No baseline correction	No scatter correction	No noise removal	Meancentering
(3x) Detrending polynomial order (2-3-4)	**(4x)** scaling: Mean Median Max L2 norm	**(9x)** S-G smoothing (window: 5-9-11 pt) (order: 2-3-4)	Autoscaling
			Range scaling
(2x) Derivatisation (1st – 2nd)	SNV		Pareto scaling
	(3x) RNV (15, 25, 35)%		Poisson scaling
AsLS	MSC		Level scaling
			Log transformation
Supervised pre-processing methods			
	OSC DOSC	No noise removal	Meancentering Autoscaling Range scaling Pareto scaling Poisson scaling Level scaling Log scaling

Radboud University Nijmegen

Pre-Processing Results

- Complexity of the model : no of LV
- Classification Accuracy



J. Engel et al. TrAC 2013

SOFTWARE

- PLS Toolbox (Eigenvector Inc.)
  - www.eigenvector.com
  - For use in MATLAB (or standalone!)
- XLSTAT-PLS (XLSTAT)
  - www.xlstat.com
  - For use in Microsoft Excel
- Package `pls` for R
  - Free software
  - http://cran.r-project.org

IMM

Radboud University Nijmegen