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
...
X: Independent variables (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 Squares

From univariate to Multiple Linear Regression (MLR)
Least squares regression
\[ y = b_0 + b_1 x_1 + \varepsilon \]
\[ b_0 : \text{intercept} \]
\[ b_1 : \text{slope} \]

MLR: Multiple Linear Regression
Least squares regression
\[ y = b_0 + b_1 x_1 + \varepsilon \]
\[ b_0 : \text{intercept} \]
\[ b_1 : \text{slope} \]

Multiple Linear Regression
\[ y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_p x_p + \varepsilon \]
\[ Y = \hat{Y} + E \]
maximizes \( \rho(y, \hat{y}) \)
**MLR: Multiple Linear Regression**

\[ y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_p x_p + \epsilon \]

Disadvantages:
- Uncorrelated X-variables required
- \( n \geq p + 1 \)

Fits a plane through a line!!

**Dimension reduction**
- **Variable Selection**
- Latent variables (PCR, PLS)

**Disadvantages:**
- Uncorrelated X-variables required
- \( n \geq p + 1 \)

**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**

1. **Dimension reduction:**
   - Use scores (projections) on latent variables that explain maximal variance in X

2. **Step 0:** Mean center X
3. **Step 1:** Perform PCA:
   - $X = TP^T \Rightarrow X' = (TP)'$
4. **Step 2:** Perform MLR
   - $Y = TA$
   - $A = (T^T)^{-1}T^TY$
5. **Step 3:** Calculate B
   - $Y = (TP)'B$
   - MLR on reconstructed $X' = (TP)'$
   - $A = P^TB$
   - $B = PA$
   - Calculate $b_0$'s:
     - $b_0 = y - \hat{y}$

**Optimal number of PC’s**

Calculate Crossvalidation RMSE for different # PC’s

**PLS: Partial Least Squares Regression**

**Phase 1:** Calculate new independent variables (T)

1. **Sequential Algorithm:** Latent variables and their scores are calculated sequentially
2. **Step 0:** Mean center X
3. **Step 1:** Calculate $w$
   - Calculate $LV = w$, that maximizes Covariance ($X,Y$): SVD on $X'Y'$
   - $w_i = 1^{st}$ col. of $W$
**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 = \( w_1 \) that maximizes Covariance (X,Y): SVD on X'Y

\[
(X'Y)_m = W_mD_mZ_m' \quad w_1 = 1^{st} \text{ col of } W
\]

- **Step 2**: Calculate \( t_1 \), scores (projections) of X on \( w_1 \)

\[
t_n = Xnpw_p1
\]

**Phase 1**: Calculate new independent variables (T)

**Phase 2**:

**Phase 3**: Optimal number of LV’s

Calculate Crossvalidation RMSE for different # LV’s

\[
RMSECV_i = \frac{\sum (\hat{y}_i - y_i)^2}{n}
\]

**MLR, PCR, PLS:**

<table>
<thead>
<tr>
<th>Set A</th>
<th>Set B</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>1.01</td>
<td>-1.89</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>-0.99</td>
<td>10.33</td>
</tr>
<tr>
<td>3.23</td>
<td>3.25</td>
<td>2.19</td>
</tr>
<tr>
<td>5.49</td>
<td>5.55</td>
<td>-8.09</td>
</tr>
<tr>
<td>0.23</td>
<td>0.21</td>
<td>11.29</td>
</tr>
<tr>
<td>2.87</td>
<td>2.91</td>
<td>1.60</td>
</tr>
<tr>
<td>3.67</td>
<td>3.76</td>
<td>1.62</td>
</tr>
</tbody>
</table>

Set A: 1.60, 1.62, 1.60, 1.62

Set B: 1.60, 1.62, 1.60, 1.62

**Common measure for prediction error**

**Root Mean Square Error (RMSE)**

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

\( \hat{y}_i \): prediction for sample \( i \)

\( y_i \): true value of sample \( i \)

\( n \): number of samples

**VALIDATION**

**Estimating prediction error**

Basic Principle:

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

Full data set

Training set

Build model:

\[ y = b_0 + b_1 x \]

RMSEP

Test set

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 entire procedure on the test and validation sets

Cross-validation

• Most simple case: Leave-One-Out (=LOO, segment=1 sample). Normally 10-20% out (=LNO).
• 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

[Diagram showing training and test sets]

Cross-validation: an example

• Split data into training set and test set

[Diagram showing training and test sets]

Cross-validation: an example

• Build a model on the training set

[Diagram showing model on training set]

Cross-validation: an example

• Check prediction of $y''$

[Diagram showing model prediction]

• Save $\hat{y}$

Cross-validation: an example

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

[Diagram showing training and test sets]

Cross-validation: an example

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

[Diagram showing training and test sets]
Cross-validation: an example

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

Cross-validation: a warning

• Data: 13 x 5 = 65 NIR spectra (1102 wavelengths)
  – 13 samples: different composition of NaOH, NaOCl, and Na₂CO₃
  – 5 temperatures: each sample measured at 5 temperatures

<table>
<thead>
<tr>
<th>Composition</th>
<th>NaOH (wt%)</th>
<th>NaOCl (wt%)</th>
<th>Na₂CO₃ (wt%)</th>
<th>Temperature (°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>18.99</td>
<td>0</td>
<td>0</td>
<td>15 21 27 34 40</td>
</tr>
<tr>
<td>2</td>
<td>9.15</td>
<td>9.99</td>
<td>0.15</td>
<td>15 21 27 34 40</td>
</tr>
<tr>
<td>3</td>
<td>15.01</td>
<td>0</td>
<td>4.01</td>
<td>15 21 27 34 40</td>
</tr>
<tr>
<td>4</td>
<td>9.34</td>
<td>5.96</td>
<td>3.97</td>
<td>15 21 27 34 40</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>13</td>
<td>16.02</td>
<td>2.01</td>
<td>1.00</td>
<td>15 21 27 34 40</td>
</tr>
</tbody>
</table>

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

Cross-validation: a warning

• The data
  
  Leave SAMPLE out

12/9/2013
**Selection of number of LV's**

Trough Validation:
- Choose number of LV's that results in model with lowest prediction error
- Testset to assess final model cannot be used
- Divide training set
- Crossvalidation

**Validation**

1) determine #LV's : with test set
2) Build model : $\hat{y}$
- $\text{RMSEP}$

**Double Cross Validation**

1) determine #LV's : CV Innerloop
2) Build model : CV Outer loop

**Remark:** Use whole data set

**Double cross-validation**

- The data
- $X$, $y$
- $\text{RMSEP}$

**Double cross-validation**

- Split data into training set and validation set

**Double cross-validation**

- Split data into training set and validation set
- $\text{RMSEP}$
- Used later to assess model performance!
Double cross-validation

- Apply cross-validation on the rest: Split training set into (new) training set and test set

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

• Build models for 1 to N LVs on the training set and predict \(y''\)

Lowest RMSECV

Double cross-validation

- Check prediction of \(y''\)

• Save \(\hat{y}\)
Cross-validation: an example

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

\[ \text{RMSEP} = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{n}} \]

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.

RMSECV vs. No of LVs

Regression coefficients
**Why Pre-Processing?**

**Data Artefacts**
- Baseline correction
- Alignment
- Scatter correction
- Noise removal
- Scaling, Normalisation
- Transformation
- …

**Other**
- Missing values
- Outliers

---

**Pre-Processing Methods**

4914 combinations: all reasonable

<table>
<thead>
<tr>
<th>Step 1: Normalisation</th>
<th>Step 2: Detrending</th>
<th>Step 3: Baseline correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>No baseline correction</td>
<td>No detrending</td>
<td>No baseline removal</td>
</tr>
<tr>
<td>Dilatation</td>
<td>Detrending</td>
<td></td>
</tr>
<tr>
<td>Spectral index</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logarithmic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Detrending</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Normalisation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spectral index</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logarithmic</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Supervised pre-processing methods:
- OLS
- No detrending
- Mean centreing
- Averaging
- Detrending
- Polynomial
- Mean centreing
- Averaging
- Log transformation
- Log transformation

---

**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