PLS

Objectives

PLS is a **prediction** model of Y from X.

- PLS proposes a set of factors for X and Y that can be used to decompose the data tables.
- These factors should explain as much as possible of the link (covariance) between X and Y.

They are called **latent factors**.

Conditions of use

- Prediction of Y from linear combinations of the explanatory variables X.
- The number of explanatory variables (*p*) can be large compared with the number of observations (*n*).
- The explanatory variables X may be highly collinear.

PLS regression is often used in chemometrics, medical data analysis or OMIC.

Advantages

- **Multivariate method**: we can predict several explanatory variables (matrix for Y) or just one (vector for Y).
- **Missing data** are handled in the PLS regression algorithm and do not prevent analysis.
- The explanatory variables can be **correlated, even strongly**.
- Can deal with a **large number of explanatory variables**, even be larger than the number of individuals.

Limits

- The **choice of the number of latent factors** needs to be determined, usually by cross-validation.
- The complexity introduced by the construction of latent factors.
- The assumption that the relationships between the variables to be explained and the explanatory variables are **linear**.

Examples

Wine tasting (PLS regression, H. Abdi)

Data matrix Y of variables to be explained

Wine	Hedonic	Goes_meat	Goes_dessert
1	14	7	8
2	10	7	6
3	8	5	5
4	2	4	7
5	6	2	4

Data matrix X of explanatory variables

Wine	Price	Sugar	Alcohol	Acidity
1	7	7	13	7
2	4	3	14	7
3	10	5	12	5
4	16	7	11	3
5	13	3	10	3

Soil organic carbon content (Statistics with R, P-A Cornillon et. al)

Extract of the data matrix (1 variable to be explained CO, explanatory variables X of wavelengths

CO	X400	X401	X402	X403	X404	X405	X406	X407	X408
1.140	0.076	0.077	0.078	0.083	0.083	0.082	0.081	0.082	0.084
1.780	0.081	0.082	0.082	0.087	0.090	0.089	0.083	0.089	0.093
1.200	0.081	0.083	0.084	0.082	0.084	0.087	0.088	0.091	0.093
1.440	0.095	0.094	0.096	0.104	0.104	0.102	0.103	0.102	0.103
1.150	0.098	0.094	0.089	0.096	0.102	0.104	0.098	0.099	0.100
1.220	0.087	0.092	0.095	0.097	0.095	0.093	0.100	0.100	0.098

PLS Algorithm(s)

The data are normalised (centred and reduced for each variable in the Y table and each variable in the X table).

Principle

We are looking for both a vector formed by a linear combination of the columns of X and a vector formed by a linear combination of the columns of Y :

$$t = \sum_{l=1}^{p} X_{.l} w_l, \ u = \sum_{k=1}^{m} Y_{.k} c_k$$

where the weight vectors $w = (w_1,...,w_p)'$ and $c = (c_1,...,c_p)'$ are normalized.

$$\sum_{l=1}^{p}w_{l}^{2} = 1, \ \sum_{k=1}^{m}c_{k}^{2} =1$$

and such that the covariance $\left(= \sum_{i=1}^{n} t_i u_i \right)$ is maximum.

Once this first factor has been constructed, its information is subtracted from the X and Y data matrices and the process is repeated for the subsequent factors.

NIPALS Algorithm

Steps for calculating the first factor

- Initialisation: $E = X^{normed}$, $F = Y^{normed}$, the values of the vector $u = (u_1,...,u_n)'$ are arbitrary.
- Steps
    - Computation of the weight vector $w = (w_1,...,w_p)$= calculate the linear combination of the rows of E as follows $\sum_{i=1}^{n} E_{i.} u_i$, normalise the result.
    - Computation of the score vector $t = (t_1,...,t_n)$= calculate the linear combination of the columns of E as follows $\sum_{j=1}^{p} E_{.j} w_j$, normalise the result.
    - Computation of the weight vector $c = (c_1,...,c_m)$= calculate the linear combination of the rows of F as follows $\sum_{i=1}^{n} F_{i.} t_i$, normalise the result.
    - Computation of the score vector $u = (u_1,...,u_n)$= calculate the linear combination of the columns of F as follows $\sum_{j=1}^{m} F_{.j} c_j$, normalise the result.
- Iterate until the vectors converge.

Steps for calculating the second factor

- Subtracting information from the first factor
    - Compute the coefficient of slope $b = \sum_{i=1}^{n} t_i u_i$ of the regression of u on t. (Note that this coefficient is equivalent to a covariance).
    - Replace each column of E with the residuals from the regression of each variable in E (in column) as a function of t. Let $E = (I - t\, t^T)\, E$
    - Similarly, subtract the information of the first factor from F. Let $F = (I - b\, t\, t^T)\, F$.
- Run the iterative algorithm with the new values of E and F (normalised) to find the second factor.

Algorithm using Matrix Decomposition

- First factor: We show that the components *t*, *u* and the weight vectors *w*, *c* are obtained as the first eigenvector of the following matrices $E\,E^TF\,F^T$, $F\,F^TE\,E^T$, $E^TF\,F^TE$, $F^TE\,E^TF$.
- Second factor: as with the nipals algorithm, we subtract the information from the first factor using the residuals.

Computation of the Loadings for X

- For each vector/factor *t*, we calculate the projection of *E* onto *t*. This produces loadings which are used to quantify the information in the initial table *X* represented/explained by each factor.
- In Abdi's article

X Loadings

```
##         [,1]    [,2]    [,3]
## [1,] -1.8706 -0.6845 -0.1796
## [2,]  0.0468 -1.9977  0.0829
## [3,]  1.9547  0.0283 -0.4224
## [4,]  1.9874  0.0556  0.2170
```

Variances

```
## [1] 11.272  4.463  0.265
```

Percentage of variance for each component

```
## [1] 0.7045 0.2790 0.0165
```

Computation of the Scores for each *t* vector

In Abdi's article, T= X scores

```
##        [,1]   [,2]   [,3]
## [1,]  0.454 -0.466  0.572
## [2,]  0.540  0.494 -0.463
## [3,]  0.000  0.000  0.000
## [4,] -0.430 -0.533 -0.530
## [5,] -0.563  0.505  0.422
```

Reconstitution of the *X* matrix from *t* vectors (scores) and associated loadings

X scores %*% t(X loadings)

```
##    Price Sugar Alcohol Acidity
## 1 -0.632     1   0.632       1
## 2 -1.265    -1   1.265       1
## 3  0.000     0   0.000       0
## 4  1.265     1  -0.632      -1
## 5  0.632    -1  -1.265      -1
```

Loadings and Scores of each factor *u* for Y

- For each factor *u*, we compute the projection of *F* onto *u*. This produces loadings which are used to quantify the information in the starting table *Y* represented/explained by each factor.

Y loadings

```
##
## Loadings:
##              Comp 1 Comp 2 Comp 3
## Hedonic       7.542 -0.410  4.791
## Goes_meat     3.996 -1.312 -0.558
## Goes_dessert  1.356 -2.615 -0.271
##
##                Comp 1 Comp 2 Comp 3
## SS loadings      74.7   8.73  23.33
## Proportion Var   24.9   2.91   7.78
## Cumulative Var   24.9  27.80  35.58
```

Yscores

```
##   Comp 1 Comp 2  Comp 3
## 1  55.96  -5.44  13.063
## 2  23.08   3.47 -10.827
## 3  -1.36   2.61   0.271
## 4 -47.89  -4.58 -12.512
## 5 -29.78   3.93  10.005
## attr(,"class")
## [1] "scores"
```

Prediction of Y

- Instead of studying the multivariate regression of Y on X , the pls expresses Y as a function of the factors t :
  - A number K of factors t is selected to reconstruct Y.
  - Each factor t is multiplied by its variance (and denoted t^*), to return to the original units rather than standardised.
  - The c weights are used to reconstruct Y from the t^*factors (see nipals algorithm).

$\hat{Y} = T_{(K)}{}^* C_{(K)}{}'$

- The loadings of X are used to express the $T_{(K)}{}^*$ matrix of factors as a function of the matrix of initial data X.
- Finally, we identify the matrix of coefficients $B_{p\,l\,s}$ which allows us to predict Y as a function of X: $\hat{Y} = X\,B_{pls}$.

PlS Coefficients :

```
# Using K=2 factors
```

```
##         Hedonic Goes_meat Goes_dessert
## Price    -1.250    -0.565    -0.000131
## Sugar     0.325     0.687     1.254032
## Alcohol   1.207     0.703     0.364195
## Acidity   1.426     0.825     0.418149
```

```
# Using K=3 factors
```

```
##         Hedonic Goes_meat Goes_dessert
## Price     -4.74    -0.158        0.198
## Sugar      1.50     0.550        1.187
## Alcohol   -6.32     1.581        0.791
## Acidity    5.50     0.350        0.187
```

Y Prediction

```
# Raw Data
```

```
##   Hedonic Goes_meat Goes_dessert
## 1      14         7            8
## 2      10         7            6
## 3       8         5            5
## 4       2         4            7
## 5       6         2            4
```

```
# Using K=1 factors
```

```
##   Hedonic Goes_meat Goes_dessert
## 1   11.09      6.64         6.56
## 2   12.39      7.33         6.79
## 3    8.00      5.00         6.00
## 4    4.36      3.07         5.35
## 5    4.16      2.96         5.31
```

```
# Using K=2 factors
```

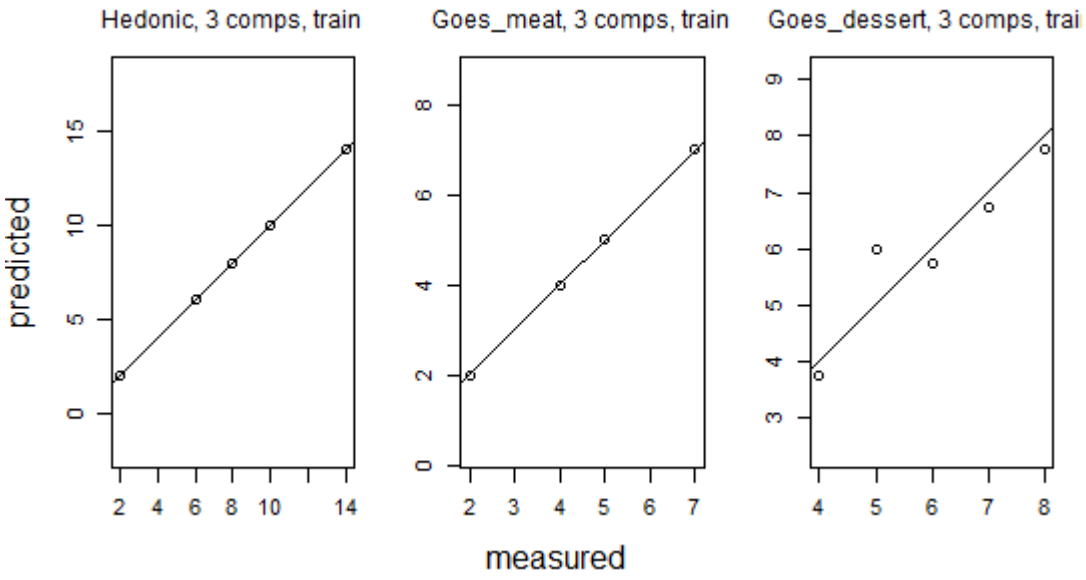
```
##   Hedonic Goes_meat Goes_dessert
## 1   11.30      7.31         7.90
## 2   12.21      6.74         5.62
## 3    8.00      5.00         6.00
## 4    4.55      3.70         6.61
## 5    3.93      2.24         3.87
```

```
# Using K=3 factors
```

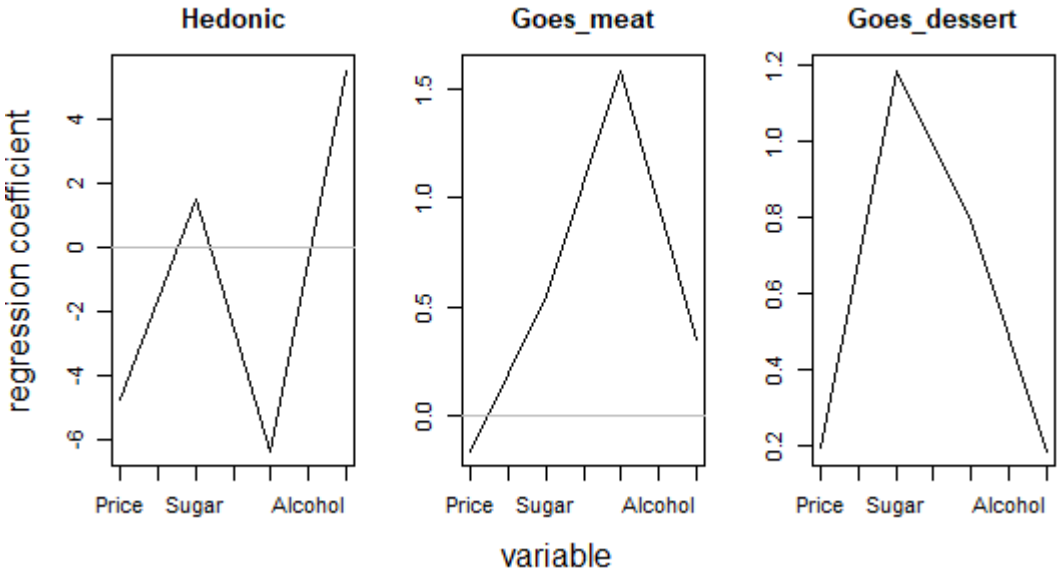
```
##   Hedonic Goes_meat Goes_dessert
## 1      14         7         7.75
## 2      10         7         5.75
## 3       8         5         6.00
## 4       2         4         6.75
## 5       6         2         3.75
```

Graphical representation of results

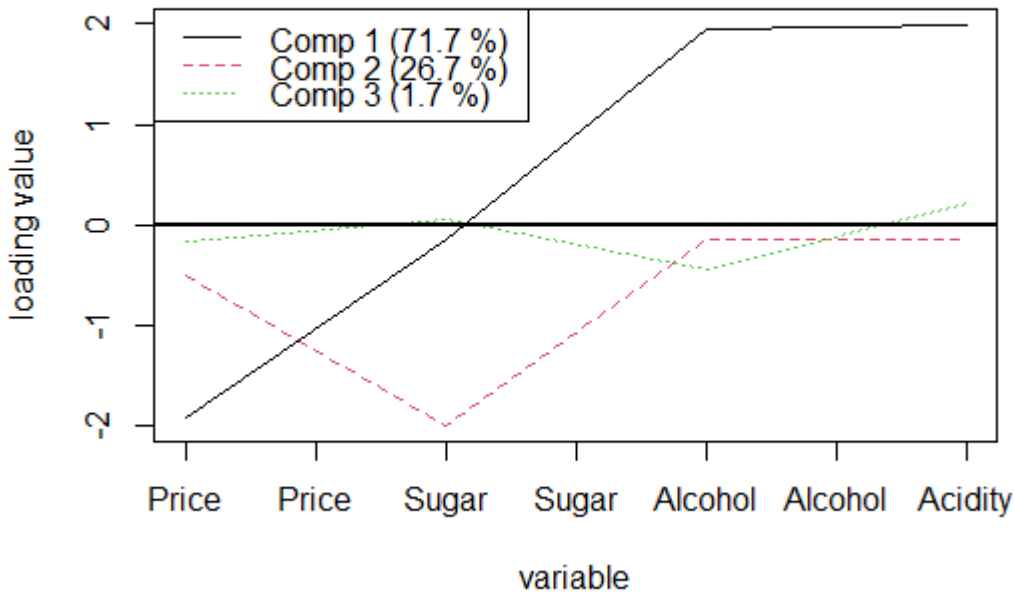
- Predictions based on observations



- PLS regression coefficients



- Loadings associated with the matrix X



Extensions

- Discriminant PLS: consists of a combination of PLS regression and discriminant analysis.
- Non-linear PLS: prior transformations of the data (X) using splines, for example. Use of non-linear combinations to construct PLS components
- Sparse PLS for parsimonious data (package *spls* from *R*)
- PLS logistic regression and generalised PLS regression
- Bibliography for this presentation
    - PLS regression, Tenenhaus Michel (book)
    - Statistics with R, Cornillon P-A *et al* (book)
    - Partial Least Squares (PLS) Regression, Abdi Hervé (article)
    - How the PLS model is calculated