The Unscrambler Appendices: Method References

This document details the ***algorithms*** used in The Unscrambler, as well as some ***statistical measures*** and ***formulas***.

The last section provides some ***bibliographical references*** for the methods implemented in The Unscrambler.

Contents

Notations Used in The Unscrambler **3**

Model Equations and Algorithms **4**
MLR Equation and Algorithm 4
MLR Model Equation 4
MLR Algorithm 5
Other Methods Based On MLR 5
PCA Equation and Algorithm 5
PCA Model Equation 5
PCA Algorithm 5
Stop criterion in PCA 6
PCR Equation and Algorithm 7
PCR Model Equation 7
PCR Algorithm 7
PLS Equation and Algorithms 8
PLS Model Equation 8
PLS1 Algorithm 8
PLS2 Algorithm 10
Stop criterion in PLS2 11
N-PLS Equation and Algorithm 11
N-PLS Model Equation 11
tri-PLS Algorithms 12

Data Centering, Interactions and Squares **15**
Data Centering 15
Interactions And Squares 16
Generating Interactions And Squares From Raw Data 16
How To Make Predictions With Interactions And Squares 16
Interactions And Squares in Analysis of Effects 17

Computation of Main Results **17**
Residuals, Variances and RMSE Computations 18
Degrees of Freedom 18
Calculation of Residuals 19
Individual Residual Variance Calculations 19
Total Residual Variance Calculations 20
Explained Variance Calculations 20
RMSEC and RMSEP Formula 21

SEP and Bias 21
Studentized Residuals 22
Weighting of individual segments in Cross Validation 22
Two-Variable Statistics Computations 22
Regression Statistics 22
Correlation Coefficient 23
RMSED and SED 23
Descriptive Statistics Computations 23
Standard Deviation 23
Histogram Statistics 23
Percentiles 24
Effects Computations 25
Significance Testing Computations 25
Standard Error of the B-coefficients 25
t-values 25
F-ratios 25
p-values 26
Multiple Comparisons 26
Comparison with a Scale-Independent Distribution (COSCIND) 26
Higher Order Interaction Effects (HOIE) 26
Leverage Calculations 27
High Leverage and Outlier Detection 28
Warning Limits and Outlier Warnings 29
Hotelling T2 Computations 30
Deviation in Prediction 31
Classification Statistics 31
Sample to Model Distance 32
Model Distance 32
Discrimination Power 33
Modeling Power 33
Class Membership Limits 33

Computation of Transformations 34
Smoothing Methods 34
Normalization Equations 34
Spectroscopic Transformation Equations 35
Multiplicative Scatter Correction Equations 36
Added Noise Equations 36
Differentiation Algorithm 36

Mixture and D-Optimal Designs 37
Shape of the Mixture Region 37
Notations 37
Simplex Region 37
Upper/Lower Bound Consistency 38
Computation of Candidate Points for a D-Optimal Design 38
D-Optimal Selection of Design Points 38

Bibliographical References 39
About Statistics and Multivariate Data Analysis 39
About Martens' Uncertainty Test 41
About Three-way Data and Tri-PLS 41
About Experimental Design 42
About Numerical Algorithms 42

Index 42

Notations Used in The Unscrambler

The tables below list the general notation and the nomenclature used throughout this manual.

General notation

Symbol	Description
v	Scalar variable
$\mathbf{v} = [v_1\ v_2\ \dots\ v_N]$	(1:N) Vector, line or column, with N elements
v_i	The ith element in the vector v
$\boldsymbol{v}^T$	Vector v transposed
	(1:I, 1:J) Matrix with I rows and J columns
$m_{ij} = \boldsymbol{M}(i, j)$	The ith element in the jth column of M. As a general rule, the parenthesis notation is used in connection with "named" matrices (eg. DescMean), whereas the index notation is used where needed for readability (eg. xik instead of xRaw(i, k)).
$\mathbf{M}_{(j)} = \mathbf{M}(\bullet, j)$	The jth column vector in M
$\mathbf{M}^T$	The matrix M transposed
$\mathbf{T}_k(i, j)$	The ith element in the jth column of the kth slice of a 3-dimensional matrix T

Nomenclature

Symbol	Description
a, *A*	Principal component (PC) number and No. of PCs
b_0, $\mathbf{b_0}$	Intercept (single, vector)
b, **b**, **B**	Regression coefficients (estimated) ***
C	0 if model is un-centered, 1 if model is centered
d	Number of degrees of freedom
E_a	X-residuals for a model using (*a*) PCs
f, F_a	Y-residuals for a model using (*a*) PCs
h, **H**	Leverages of samples (single, matrix)
i, *I*	Sample number and No. of samples
j, *J*	Y-variable number and No. of Y-variables
k, *K*	X-variable number and No. of X-variables
N	Number of elements
p, **P**	X-Loadings (vector, matrix)
q, **Q**	Y-Loadings (vector, matrix)
ß	B-coefficient (exact)
t, **T**	Scores (vector, matrix)
u, **U**	Preliminary scores (vector, matrix)
w, **W**	X-Loading weights (vector, matrix)
$\bar{x}$, $\bar{\mathbf{x}}$	Mean value in x or X
x, **x**, **X**	x-values (single, vector, matrix)

Symbol	Description
$\bar{y}$, $\bar{\mathbf{y}}$	Mean value in y or Y
$\hat{y}$	Predicted value of *y*
y, **y**, **Y**	y-values (single, vector, matrix)

Model Equations and Algorithms

The equations used in the different analysis methods are shown here.

The Unscrambler contains the algorithms for MLR, PCA, PCR, and PLS.

For further reading about algorithms and computations in general we refer you to the texts by Golub and Press (see section The D-optimal selection of a design point is based on the algorithm DOPT (MILLER and NGUYEN, 1994).

The FORTRAN algorithm DOPT is used to D-optimally select a number of design points from a set of candidate points. The design points are chosen so that the determinant of X'X is maximized (X is the design point matrix).

DOPT:

1. Start with a default set of design points, or else generate a random set.
2. Calculate X'X for the selected points.
3. As long as there is improvement do:
   a. Find the selected point and unselected candidate point which will improve the X'X determinant the most.
   b. Exchange the points, and calculate new X'X.

Bibliographical References).

MLR Equation and Algorithm

MLR is used as common basis for three analysis methods used in The Unscrambler:

- Multiple Linear Regression;
- Analysis of Effects;
- Response Surface Analysis.

MLR Model Equation

The general model equation, which relates a response variable to several predictors by means of regression coefficients, has the following shape:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_K x_K + f$$

or, in matrix notation : y = **X** b + f.

MLR Algorithm

The SVD (Singular Value Decomposition) algorithm is the most widely used algorithm to compute the estimated regression coefficients for MLR.

Other Methods Based On MLR

Analysis of Effects and Response Surface are based on MLR computations, and algorithms will not be shown explicitly for these methods here.

PCA Equation and Algorithm

The algorithms used in The Unscrambler for PCA, PCR and PLS are described in "Multivariate calibration" by Martens & Næs (Wiley 1991, ISBN 0471930474). The following descriptions of the PCA, PCR and PLS algorithms are reproduced from the book, with the permission of John Wiley & Sons Ltd.

PCA Model Equation

The general form of the PCA model is:

$$X = T \cdot P^T + E$$

Usually the PCA model is centered, which gives:

$$X = 1 \cdot x_{mean} + T_{(A)} \cdot P_{(A)}^T + E_{(A)}$$

Another way to put this is:

$$x_{ik} = x_{mean,k} + \sum_{a=1}^{A} t_{ia} p_{ka} + e_{ik(A)}$$

PCA Algorithm

The PCA computations implemented in The Unscrambler are based on the NIPALS algorithm.

The NIPALS algorithm for PCA

The algorithm extracts one factor at a time. Each factor is obtained iteratively by repeated regressions of **X** on scores $\hat{t}$ to obtain improved $\hat{p}$ and of **X** on these $\hat{p}$ to obtain improved $\hat{t}$. The algorithm proceeds as follows:

Pre-scale the X-variables to ensure comparable noise-levels. Then center the X-variables, e.g. by subtracting the calibration means $\bar{x}'$, forming $\mathbf{X}_0$. Then for factors a = 1, 2, ..., A compute $\hat{t}_a$ and $\hat{p}_a$ from $\mathbf{X}_{a-1}$:.

Start:

Select start values. e.g. $\hat{t}_a$ = the column in $\mathbf{X}_{a-1}$ that has the highest remaining sum of squares.

Repeat points i) to v) until convergence.

i) Improve estimate of loading vector $\hat{p}_a$ for this factor by projecting the matrix $\mathbf{X}_{a-1}$ on $\hat{t}_a$, i.e.

$$\hat{p}'_a = (\hat{t}'_a \hat{t}_a)^{-1} \hat{t}'_a X_{a-1}$$

ii) Scale length of $\hat{p}_a$ to 1.0 to avoid scaling ambiguity:

$$\hat{p}_a = \hat{p}_a (\hat{p}'_a \hat{p}_a)^{-0.5}$$

iii)Improve estimate of score $\hat{t}_a$ for this factor by projecting the matrix $\mathbf{X}_{a-1}$ on $\hat{p}_a$:

$$\hat{t}_a = X_{a-1} \hat{p}_a (\hat{p}'_a \hat{p}_a)^{-1}$$

iv)Improve estimate of the eigenvalue $\hat{\tau}_a$:

$$\hat{\tau}_a = \hat{t}'_a \hat{t}_a$$

v) Check convergence: If $\hat{\tau}_a$ minus $\hat{\tau}_a$ in the previous iteration is smaller than a certain small pre-specified constant, e.g. 0.0001 times $\hat{\tau}_a$, the method has converged for this factor. If not, go to step i).

Subtract the effect of this factor:

$$X_a = X_{a-1} - \hat{t}_a \hat{p}'_a$$

and go to *Start* for the next factor

Stop criterion in PCA

Some users of Unscrambler versions prior to 7.5 have reported that the stop criterion has been too loose in some situations compared to e.g. Matlab results. As a result, in The Unscrambler 7.5 and later versions the stop criterion has been changed to $\|t_{old}-t\| < 1.e-12$, giving more strict orthogonality in scores and loadings. The maximum number of iterations has been changed as well, from 50 to 100.

PCR Equation and Algorithm

PCR is based on PCA and MLR.

PCR Model Equation

The general form of the PCR model is:

$X = T \cdot P^T + E$ and $y = T \cdot b + f$

where the decomposition of the X matrix is computed using PCA and the b-coefficients are computed using MLR.

PCR Algorithm

PCR is performed as a two step operation:

4. First X is decomposed by PCA, see PCA Algorithm.
5. Then the principal components regression is obtained by regressing y on the $\hat{t}$'s.

A Principal Component Regression of J different Y-variables on K X-variables is equivalent to J separate principal component regressions on the same K X-variables (one for each Y-variable). Thus we here only give attention to the case of one single Y-variable, y.

The principal component regression is obtained by regressing **y** on the $\hat{t}$'s obtained from the PCA of X. the regression coefficients $\hat{b}$ for each y can according to equation (1) be written

(1) $\hat{b} = \hat{P}\hat{q}$

where X-loadings

$\hat{P} = \{\hat{p}_{ka}, k = 1,2,\ldots,K \text{ and } a = 1,2,\ldots,A\}$

represent the PCA loadings of the A factors employed, and Y-loadings

$\hat{q} = (\hat{q}_1,\ldots,\hat{q}_A)'$

are found the usual way by least squares regression of y on $\hat{T}$ from the model y=Tq+f.

Since the scores in $\hat{T}$ are uncorrelated, this solution is equal to

$\hat{q} = (\text{diag}(1/\hat{\tau}_a))^{-1}\hat{T}'y$

Inserting this in (1) and replacing $\hat{T}$ by $X\hat{P}$, the PCR estimate of b can be written as

$\hat{b} = \hat{P}(\text{diag}(1/\hat{\tau}_a))\hat{P}'X'y$

which is frequently used as definition of the PCR (Gunst and Mason, 1979).

When the number of factors A equals K, the PCR gives the same $\hat{b}$ as the MLR. But the X-variables are often intercorrelated and somewhat noisy, and then the optimal A is less than K:

In such cases MLR would imply division by eigenvalues $\hat{\tau}_a$ close to zero, which makes the MLR estimate of b unstable. In contrast PCR attains a stabilized estimation of b by dropping such unreliable eigenvalues.

PLS Equation and Algorithms

PLS decomposes X and Y simultaneously. It is found in two versions:

- PLS2 is the most general and handles several Y-variables together;
- PLS1 is a simplification of the PLS algorithm made possible in the case of only one Y-variable.

PLS Model Equation

The general form of the PLS model is:

$$X = T \cdot P^T + E \quad \text{and} \quad Y = T \cdot B + F$$

PLS1 Algorithm

Orthogonalized PLSR algorithm for one Y-variable: PLS1.

Calibration:

C 1 The scaled input variables **X** and y are first centered, e.g.

$$X_0 = X - 1\bar{x}' \quad \text{and} \quad y_0 = y - 1\bar{y}$$

Choose $A_{\max}$ to be higher than the number of phenomena expected in **X**.

For each factor $a = 1,..., A_{\max}$ perform steps C 2.1 - C 2.5:

C 2.1 Use the variability remaining in y to find the loading weights $\mathbf{w}_a$, using LS and the local 'model'

$$X_{a-1} = y_{a-1} w'_a + E$$

and scale the vector to length 1. The solution is

$$\hat{w}_a = c X'_{a-1} y_{a-1}$$

where c is the scaling factor that makes the length of the final $\hat{w}_a$ equal to 1, i.e.

$$c = (y'_{a-1} X_{a-1} X'_{a-1} y_{a-1})^{-0.5}$$

C 2.2 Estimate the scores $\hat{t}_a$ using the local 'model'

$$X_{a-1} = t_a \hat{w}'_a + E$$

C 2.4 Estimate the chemical loading q_a using the local 'model'

$$y_{a-1} = \hat{t}_a q_a + f$$

which gives the solution

$$\hat{q}_a = y'_{a-1} \hat{t}_a / \hat{t}'_a \hat{t}_a$$

C 2.5 Create new **X** and **y** residuals by subtracting the estimated effect of this factor:

$$\hat{E} = X_{a-1} - \hat{t}_a \hat{p}'_a$$

$$\hat{f} = y_{a-1} - \hat{t}_a \hat{q}_a$$

Compute various summary statistics on these residuals after a factors, summarizing $\hat{e}_{ik}$ over objects *i* and variables *k*, and summarizing $\hat{f}_i$ over *i* objects (see Chapters 4 and 5).

Replace the former X_{a-1} and y_{a-1} by the new residuals $\hat{E}$ and $\hat{f}$ and increase a by 1, i.e. set

$$X_a = \hat{E}$$

$$y_a = \hat{f}$$

$$a = a + 1$$

C 3 Determine *A*, the number of valid PLS factors to retain in the calibration model.

C 4 Compute $\hat{b}_0$ and $\hat{b}$ for *A* PLS factors, to be used in the predictor $\hat{y} = 1\hat{b}_0 + X\hat{b}$

(optional, see P4 below)

$$\hat{b} = \hat{W}(\hat{P}'\hat{W})^{-1}\hat{q}$$

$$\hat{b}_0 = \bar{y} - \bar{x}'\hat{b}$$

Prediction:

Full prediction

For each new prediction object *i* = 1,2,... perform steps P1 to P3, or alternatively, step P4.

P1 Scale input data x_i like for the calibration variables. Then compute

$$x'_{i,0} = x'_i - \bar{x}'$$

where $\bar{x}$ is the center for the calibration objects.

For each factor $a = 1 \ldots A$ perform steps P 2.1 - P 2.2.

P 2.1 Find $\hat{t}_{i,a}$ according to the formula in C 2.2 i.e.

$$\hat{t}_{i,a} = x'_{i,a-1}\hat{w}_a$$

P 2.2 Compute new residual $x_{i,a} = x_{i,a-1} - \hat{t}_{ia}\hat{p}'_a$

If $a < A$, increase a by 1 and go to P 2.1. If $a = A$, go to P 3.

P 3 Predict y_i by $\hat{y}_i = \bar{y} + \sum_{a=1}^{A} \hat{t}_{ia}\hat{q}_a$

Compute outlier statistics on $\mathbf{x}_{iA}$ and $\hat{t}_i$ (Chapters 4 and 5).

Short prediction

P 4 Alternatively to steps P 1 - P 3, find $\hat{y}$ by using $\hat{b}_0$ and $\hat{b}$ in C 4, i.e.

$$\hat{y}_i = \hat{b}_0 + x'_i\hat{b}$$

Note that P and Q are not normalized. T and W are normalized to 1 and orthogonal.

PLS2 Algorithm

Simultaneous PLSR calibration for several *Y*-variables: PLS2.

If we replace vectors **y**, **f**, and **q** in the PLS1 algorithm by matrices **Y**(dim *I*J*), **F**(dim *I*J*) and **Q**(dim *J*A*), the calibration in PLS2 is almost the same as for the orthogonalized PLS1.

The exceptions are that y_{a-1} in C 2.1 is replaced by a temporary Y-score for this factor, $\hat{u}_a$ and that two extra steps are needed between C 2.4 and C 2.5:

C 2.1 Use the temporary *Y*-factor $\hat{u}_a$ that summarizes the remaining variability in **Y**, to find the loading-weights $\hat{w}_a$ by LS, using the local 'model'

$$X_{a-1} = \hat{u}_a w'_a + E$$

and scale the vector to length 1. The LS solution is

$$\hat{w}_a = cX'_{a-1}\hat{u}_a$$

where c is the scaling factor that makes the length of the final $\hat{w}_a$ equal to 1, i.e.

$$c = (\hat{u}'_a X_{a-1} X'_{a-1} \hat{u}_a)^{-.5}$$

The first time this step is encountered, $\hat{u}_a$ has been given some start values, e.g. the column in Y_{a-1} with the largest sum of squares.

The following two extra stages are then needed between C 2.4 and C 2.5:

C 2.4b

Test whether convergence has occurred, by e.g. checking that the elements have no longer changed meaningfully since the last iteration.

C 2.4c

If convergence is not reached, then estimate temporary factor scores u_a using the 'model'

$$Y_{a-1} = u_a \hat{q}'_a + F$$

giving the LS solution

$$\hat{u}_a = Y_{a-1} \hat{q}_a (\hat{q}'_a \hat{q}_a)^{-1}$$

and go to C 2.1.

If convergence has been reached, then go to step 2.5.

The expression for $\hat{B}$ is the same in this PLS2 algorithm as in the PLS1 algorithm, i.e.

$$\hat{B} = \hat{W}(\hat{P}'\hat{W})^{-1}\hat{Q}'$$

and

$$b'_0 = \bar{y}' - \bar{x}'\hat{B}$$

Stop criterion in PLS2

Some users of Unscrambler versions prior to 7.5 have reported that the stop criterion has been too loose in some situations compared to e.g. Matlab results. As a result, in The Unscrambler 7.5 and later versions the stop criterion has been changed to $\|t_{old}-t\| < 1.e-12$, giving more strict orthogonality in scores and loadings. The maximum number of iterations has been changed as well, from 50 to 100.

N-PLS Equation and Algorithm

N-PLS or tri-PLS is the new method that allows you to build a model where a matrix of responses (Y) is expressed as a function of a 3-way array of predictors.

N-PLS Model Equation

The general form of the N-PLS model is:

$$X = T \cdot (W^{(2)} \circ W^{(1)})^T + E \text{ and } Y = T \cdot B + F$$

where X is unfolded to a matrix and $\circ$ is the Khatri-Rao product (columnwise Kronecker product).

tri-PLS Algorithms

The tri-PLSR algorithm for one or several Y-variables using three-way X (tri-PLS1 and tri-PLS2 regression) is given hereafter.

Note: Step numbering has been adjusted so as to be consistent with the numbering in the PLS1 and PLS2 algorithms.

For tri-PLS1, the matrix Y has only one column. Below the three-way X(dim I*K*L) has been rearranged to a matrix X with I rows and KL columns (unfolded/matricized).

Calibration:

C 1 The scaled input variables **X** and **Y** are first centered, i.e.

$$X_0 = X - 1\bar{x}' \text{ and } Y_0 = Y - 1\bar{y}$$

Choose A_{max} to be higher than the number of phenomena expected in X.

For each factor a = 1,..., A_{max} perform steps C 2.1 - C 2.5:

C 2.1 Use the variability remaining in Y to find the loading weights w_a, using LS and the local 'model' below. For each component there is a weight vector for the first variable mode, $\mathbf{w}^{(1)}$(dim *K*A*) and one for the second variable mode $w^{(2)}$(dim *L*A*)

$$X_0 = u_a (w_a^{(2)} \otimes w_a^{(1)})' + E$$

Scale the weight vectors to length 1. This model is a trilinear model similar to the two-way bilinear analogue. The solution is obtained from a one-component PCA model of the matrix Z(dim *K*L*) which is the 'inner' product of **X** and **u**. Hence the element *k,l* of **Z** is the inner product of **u** and the column in **X** with variable mode 1 index *k* and variable mode 2 index *l*. The normalized score vector of the one-component PCA model of **Z** is equal to $w_a^{(1)}$ and the loading vector to $w_a^{(2)}$. From the two weight vectors, a combined weight vector w_a applicable for the rearranged X data, is defined as

$$w_a = w_a^{(2)} \otimes w_a^{(1)}) .$$

This is the Kronecker tensor product which is a larger matrix formed from all possible products of elements of $w_a^{(1)}$ with those of $w_a^{(2)}$.

The first time this step is encountered, $\hat{u}_a$ has been given some start values, e.g. the column in Y_{a-1} with the largest sum of squares.

C 2.2 Calculate the scores $\hat{t}_a$ using the local 'model'

$$X_{a-1} = t_a \hat{w}'_a + E$$

which has the solution

$$\hat{t}_a = X_0 w_a$$

C 2.4 Estimate the chemical loading q_a using the local 'model'

$$Y_{a-1} = \hat{t}_a q_a + F$$

which gives the solution

$$\hat{q}_a = Y_{a-1}'\hat{t}_a / \hat{t}_a'\hat{t}_a .$$

Subsequently $\hat{q}_a$ is normalized to length 1 (unlike in two-way PLS regression).

C 2.4c Estimate factor scores u_a using the 'model'

$$Y_{a-1} = u_a \hat{q}'_a + F$$

giving the LS solution

$$\hat{u}_a = Y_{a-1}\hat{q}_a(\hat{q}'_a\hat{q}_a)^{-1}$$

C 2.4d Test whether convergence has occurred by e.g. checking that the elements have no longer changed significantly since the last iteration. For tri-PLS1 where there is only one Y-variable, convergence is reached immediately after first iteration when $\mathbf{u}_a$ is initialized as $\mathbf{y}_{a-1}$. The stopping criterion may be 10^{-6}, or less, as desired.

Check convergence: if $| u_a - u_{a-1} |$ < criterion, convergence has been reached. Then go to step 2.5 else go to C 2.1.

C 2.5a Determine inner-relation regression coefficients for estimating $\mathbf{u}_a$ from $\mathbf{t}_a$. Due to non-orthogonal X-scores include all scores from 1 to a.

$$\hat{b}_a^{inner} = (\hat{T}_{1-a}'\hat{T}_{1-a})^{-1}\hat{T}_{1-a}'\hat{u}_a$$

Note that there is a unique set of a inner relation coefficients for each component. Hence, for all components these are held in a matrix (dim A*A) which has zeros on each lower triangular part.

C 2.5b Calculate core array. The core array is used for building the **X** model to obtain X-residuals. This model is a so-called Tucker structure where the core is used to relate the scores and weights using the model

$$\hat{X} = \hat{T}_{1-a} G_a (W_{1-a}^{(2)} \otimes W_{1-a}^{(1)})' + \hat{E}_a$$

The core is determined in a least squares sense from

$$\text{vec}G_a = (S'S)^{-1}S'\text{vec}X_0 + \hat{E}_a$$

where $\text{vec}G_a$ is the core array G_a(dim a*a*a) rearranged to a vector and $\text{vec}X_0$ is defined likewise. The matrix S is defined as

$$S = W_{1\text{-}a}^{(2)} \otimes W_{1\text{-}a}^{(1)} \otimes \hat{T}_{1-a}$$

Note that X-residuals are not used in the algorithm, but only for diagnostic purposes.

C 2.5c Create new y-residuals by subtracting the estimated effect of this factor

$$\hat{F} = Y_{a\text{-}1} - \hat{T}_{1\text{-}a}\hat{B}_{1\text{-}a,a}\hat{q}'_a$$

Replace the former $Y_{a\text{-}1}$ by the new residuals $\hat{F}$ and increase *a* by 1, i.e. set

$$Y_a = \hat{F}$$

$$a = a + 1$$

C 3 Determine A, the number of valid PLS factors to retain in the calibration model.

C 4 Compute $\hat{B}_0$ and $\hat{B}$ for A PLS factors, to be used in the predictor

$$\hat{Y} = \hat{B}_0 + X\hat{B}$$

(optional, see P4 below)

$$\hat{B} = \hat{W}\hat{B}^{\text{inner}}\hat{Q}$$

and

$$\hat{B}_0 = \bar{Y}' - \bar{X}'\hat{B}$$

Note that Q and W are normalized to 1 and orthogonal.

Prediction:

Full prediction

The three-way X(dim I*K*L) is rearranged to a matrix X with I rows and KL columns (unfolded/matricized).

Perform steps P1 to P4, or alternatively, step P5.

P1 Scale input data X like for the calibration variables. Then compute

$$X_0 = X - 1\bar{x}'$$

where $\bar{x}$ is the center for the calibration objects.

P2 Find $\hat{T}$ according to the formula

$$T = XW$$

P3 Predict Y by

$$\hat{Y} = \hat{T}\hat{B}\hat{Q}'$$

P4 Calculate X-residuals

$$E = X_0 - \hat{T}\hat{G}(\hat{W}^{(2)} \otimes \hat{W}^{(1)})'$$

Compute outlier statistics on xiA and $\hat{t}_i$ (Chapters 4 and 5).

Short prediction

P5 Alternatively to steps P1 – P4, find $\hat{Y}$ by using $\hat{B}_0$ and $\hat{B}$ in C 4, i.e.

$$\hat{Y} = \hat{B}_0 + X\hat{B}$$

Data Centering, Interactions and Squares

Data centering and computation of interactions and squares are done automatically in The Unscrambler, according to the formulas given in the sections that follow.

Data Centering

The center value is either 0 (origo) or the x- or y-variable mean:

$$\mathbf{xCent}(1,k) = \begin{cases} 0 & \text{if model center is origo} \\ \frac{1}{I}\sum_{i=1}^{I} \mathbf{xRaw}(i,k) & \text{if model center is mean} \end{cases}$$

$$\mathbf{yCent}(1,j) = \begin{cases} 0 & \text{if model center is origo} \\ \frac{1}{I}\sum_{i=1}^{I} \mathbf{yRaw}(i,j) & \text{if model center is mean} \end{cases}$$

Interactions And Squares

PCA, PCR, PLS, MLR, Classification, Prediction, Response Surface and Analysis of Effects are based on X-Variable Sets which may include Interaction and Square effects. These special X-variables are not stored together with the raw data in your table; they are generated “on the fly” from your data selection, each time you make a new model.

Generating Interactions And Squares From Raw Data

In all cases, except Analysis of Effects, interaction and square effects are calculated from standardized main predictor variables (X-variables). If x_{AB} is the interaction term of variables *A* and *B*, and x_{A^2} is the square term of variable *A*, then:

$$x_{AB}(i) = \text{WeightI\&S}(A) \cdot \left(x_{i,A} - \text{CentI\&S}(A)\right) \cdot \text{WeightI\&S}(B) \cdot \left(x_{i,B} - \text{CentI\&S}(B)\right)$$

$$x_{A^2}(i) = \left(\text{WeightI\&S}(A) \cdot \left(x_{i,A} - \text{CentI\&S}(A)\right)\right)^2$$

where

$$\text{CentI\&S}(k) = \frac{1}{I_C} \sum_{i \in \{\text{calibration samples}\}} x_{i,k}$$

and

$$\text{WeightI\&S}(k) = \frac{1}{\sqrt{\frac{1}{I_c - 1} \sum_{i \in \{\text{calibration samples}\}} \left(x_{i,k} - \text{CentI\&S}(k)\right)^2}}$$

The standardized main variables are only used in calculating the interaction and square effects. The analysis is otherwise based on raw data values of the main variables. Centering and weighting options specific to each analysis (used in PLS, PCR, PCA and Response Surface) are applied to the data, according to user choice, after the interaction and square effects have been generated.

How To Make Predictions With Interactions And Squares

The strategy varies depending on the type of model to be used.

Predictions From PCR And PLS Models

The Unscrambler takes care of predictions from PCR or PLS models with the Predict task. If the X-Variable Set your regression model is based on, contains any interactions and squares, you will get correct results provided that you select the same X-Variable Set for prediction.

SIMCA Classification

The same applies to a Classification with PCA models based on an X-Variable Set containing interactions and squares.

Predictions from MLR and Response Surface Models

MLR and Response Surface models, on the other hand, cannot be used for automatic predictions. If you want to predict response values for new samples, you have to do it manually, using a prediction equation based on the regression coefficients (B-coefficients). If the source model contains any interaction and square effects, you have to generate these variables from your raw data using the CentI&S and WeightI&S values stored together with the model results.

In practice, here is how to do it:

1- Build your model and save it.

2- Import the regression coefficients (stored in matrices B0 and B of your model result file) into a new data table.

3- Import the CentI&S and WeightI&S matrices from your model result file into another data table.

4- Copy those numbers to a worksheet (e.g. Excel), and prepare a formula for computing the interactions and squares from your raw data. Use the equations given above for X_{AB} and X_{A2}.

5- Prepare a prediction formula which combines the imported regression coefficients and the values of the main X-variables, of X_{AB} and X_{A2}.

Note: In Response Surface analysis, the main predictor variables are centered before calculating the B-coefficients. The same must be done to main variables which are used in prediction.

Interactions And Squares in Analysis of Effects

Analysis of Effects uses coded levels of the X-variables instead of raw values (see section Descriptive Statistics Computations). Interaction and square effects are calculated directly from the coded values, without any standardization.

Computation of Main Results

This section gives the principles and formulas applied in the main computations performed by The Unscrambler: various result matrices and warnings.

Residuals, Variances and RMSE Computations

Residuals are computed as the difference between measured values and fitted values, and are thereafter combined into various kinds of ***error measures***. Variances and RMSEC/RMSEC are the most commonly used.

In The Unscrambler, variances are usually computed with a correction for the number of residual degrees of freedom.

Degrees of Freedom

The residual Degrees of Freedom (d.f.) are taken into account in the computation of conservative variance estimates.

The number of Degrees of Freedom varies according to the context. The table below explains how the Degrees of Freedom are determined in The Unscrambler.

Degrees of Freedom

Degrees of freedom	Equation
d_1	$\frac{1}{K}\left(KI_c - KC - a \cdot \max\left[I_c - C, K\right]\right)$
d_2	I_c
d_3	$I_{pr}\left(\frac{K-a}{K}\right)$
d_4	$I_c - C - a$
d_5	I_{pr}
d_6	$\frac{1}{I_c}\left(KI_c - KC - a \cdot \max\left[I_c - C, K\right]\right)$
d_7	$K - a$
d_8	$\frac{1}{I_c}\left(J\left(I_c - C - a\right)\right)$
d_9	J
d_{10}	K

Note: Not all statistical and multivariate data analysis packages correct the variation in the data for degrees of freedom. And there are different ways of doing it. The variances calculated in The Unscrambler may therefore differ somewhat from other packages. You may multiply the result by the adequate “d” factor if you wish to get the uncorrected variance.

Calculation of Residuals

The residuals are calculated as the difference between the actual value and the predicted or fitted value.

$$f_{ij,\mathrm{Cal}} = (y_{ij} - \hat{y}_{ij,\mathrm{Cal}}) \quad , \; i = 1 \ldots I_C \, , \quad j = 1 \ldots J$$

Residuals are calculated for X (Eix) and Y(Fiy).

Individual Residual Variance Calculations

Residual variance is defined as the mean squared residual corrected for degrees of freedom. Residual variances are calculated for models incorporating an increasing number of PCs, $a = 0 \ldots A$.

Replace elements in the equations below with the correct combination as found (CV = Cross Validation; TS = Test Set validation; LC = Leverage Correction):

Elements of the individual Residual Variance calculations

Description	ResVar	CV & TS	LC	n, N	R
Variance per X-variable calibration samples	ResXCalVar	d_2	d_2	i, I_C	Eix
Variance per X-variable validation samples	ResXValVar	d_3	d_1	i, I_{pr}	Eix
Variance per Y-variable calibration samples	ResYCalVar	d_2	d_2	i, I_C	Fiy
Variance per Y-variable validation samples	ResYValVar	d_5	d_2	i, I_{pr}	Fiy
Variances per samples in X calibration samples	ResXCalSamp	d_{10}	d_{10}	k, K	Eix
Variances per samples in X validation samples	ResXValSamp	d_7	d_{10}	k, K	Eix
Variances per samples in Y calibration samples	ResYCalSamp	d_9	d_9	j, J	Fiy
Variances per samples in Y validation samples	ResYValSamp	d_9	d_9	j, J	Fiy
MLR: Variance per Y-variable calibration samples	ResYCalVar	d_4	d_4	i, I_C	Fiy
MLR: Variances per samples in Y calibration samples	ResYCalSamp	d_8	d_8	j, J	Fiy

$$\mathrm{ResVar}(a,z) = \frac{1}{d}\sum_{n=1}^{N}\frac{R_a^2}{(1-H_i)^2}$$

The term $(1-H_i)^2$ is used only when the calibration samples are leverage corrected. For cross validation and test set this equation is used:

$$ResVar(a,z) = \frac{1}{d}\sum_{n=1}^{N} R_a^2$$

Total Residual Variance Calculations

The total residual variance is calculated from the residual variance for *a* PCs, *a* = 0…*A*.

$$ResTot(a,\bullet) = \frac{1}{N}\sum_{n=1}^{N} ResVar(a,n)$$

The different cases of Total Residual Variance matrices are listed below.

Elements of the Total Residual Variance calculations

ResTot	n,N	ResVar
ResXCalTot	k, K	ResXCalVar
ResXValTot	k, K	ResXValVar
ResYCalTot	j, J	ResYCalVar
ResYCalTotCVS	j, J	ResYCalVar for each CVS segment
ResYValTot	j, J	ResYValVar
ResYValTotCVS	j, J	ResYCalVar for each CVS segment

Explained Variance Calculations

The explained variance for PC *a* is expressed in %, and is calculated from the residual variance as:

$$V_{Exp}(0) = 0$$

$$V_{Exp}(a) = \begin{cases} \dfrac{V_{Res}(a-1) - V_{Res}(a)}{V_{Res}(0)} & \text{if } \left(V_{Res}(a-1) - V_{Res}(a)\right) > 0 \\ 0 & \text{if } \left(V_{Res}(a-1) - V_{Res}(a)\right) \le 0 \end{cases}, \quad a = 1 \ldots A$$

where V_{Res} is any of the residual variance matrices listed in Individual Residual Variance Calculations and V_{Exp} is the corresponding explained variance matrix.

Cumulative explained variances after *a* PCs are also computed, according to the following equation:

$$V_{Exp,cum}(a) = \frac{V_{Res(0)} - V_{Res(a)}}{V_{Res(0)}}$$

They are also expressed as a percentage.

RMSEC and RMSEP Formula

The **R**oot **M**ean **S**quare **E**rror is calculated for the prediction or validation samples (RMSEP) and for the calibration samples (RMSEC).

RMSEC (all validation methods):

$$\text{RMSEC} = \frac{1}{\text{yWeight}}\sqrt{\text{ResYCalVar}}$$

RMSEP (leverage correction and test set validation):

$$\text{RMSEP} = \frac{1}{\text{yWeight}}\sqrt{\text{ResYValVar}}$$

RMSEP (cross validation):

$$\text{RMSEP} = \sqrt{\frac{1}{\text{I}_{\text{tot}}}\sum_{\text{s}=1}^{\text{Nseg}}\frac{1}{\text{yWeights}^2}\sum_{\text{i}=1}^{\text{Is}}\text{Fiys(i, j)}^2}$$

SEP and Bias

Bias is the average value of the difference between predicted and measured values.

$$\text{Bias} = \frac{1}{I}\sum_{i=1}^{I}(\hat{y}_i - y_i)$$

SEP (Standard Error of Prediction) is the standard deviation of the prediction residuals.

$$\text{SEP} = \sqrt{\frac{1}{I-1}\sum_{i=1}^{I}(\hat{y}_i - y_i - \text{Bias})^2}$$

Studentized Residuals

Residuals can be expressed raw, or studentized. Studentization takes into account the standard deviation and sample leverage, so that studentized residuals can be compared to each other on a common scale.

$$r_{ij} = \frac{f_{ij,\mathrm{Cal}}}{\hat{\sigma}_j \sqrt{1 - h_i}} \quad , \; i = 1 \ldots I, \;\; j = 1 \ldots J$$

Weighting of individual segments in Cross Validation

With the assumption that the validation should reflect the prediction error for future samples, one has to decide whether samples that are kept out in the current segment should be re-centered and/or re-weighted based on mean and standard deviation for the samples in the segment.

The Unscrambler has in previous versions both re-centered and re-weighted in each cross validation segment, which is a rather conservative approach. The re-weighting is particularly conservative for small (heterogeneous) data sets. Without discussing in more detail which approach is conceptually the best one, we have removed the re-weighting in version 7.5 (and later versions). The effect is that the explained cross validation variance is slightly increased, thus being less conservative.

Two-Variable Statistics Computations

Various statistics can be calculated for two data vectors plotted against each other.

Regression Statistics

The regression statistics are calculated for 2D scatter plots. The Least Squares method is used to fit the elements to the regression line y = ax + b, where a is the slope and b is the offset (or intercept).

$$\mathrm{Slope} = \frac{N \sum y \cdot x - \sum y \sum x}{N \sum x^2 - \left(\sum x \right)^2}$$

$$\mathrm{Offset} = \frac{1}{N} \left(\sum y - \mathrm{Slope} \cdot \sum x \right)$$

$$\mathrm{Bias} = \frac{1}{\mathrm{N}} \sum \left(y - x \right)$$

Correlation Coefficient

The correlation $r_{k_1k_2}$ between two variables $k1$ and $k2$ is calculated as:

$$r_{k_1k_2} = \frac{\sum_{i \in S_k} (x_{ik_1} - \bar{x}_{k_1}) \cdot (x_{ik_2} - \bar{x}_{k_2})}{(I-1) \cdot S_x(k_1) \cdot S_x(k_2)} \qquad \text{for } k_1, k_2 = 1 \ldots K$$

Note that $r_{kk} \equiv 1$ when $k1 = k2$.

RMSED and SED

The Root Mean Squared Error of Deviation is calculated for general 2D scatter plots, and is the same measure as RMSEC and RMSEP.

$$\text{RMSED} = \sqrt{\frac{1}{\text{I}} \sum_{\text{i}=1}^{\text{I}} \left(\text{y}_{\text{i}} - \text{x}_{\text{i}} \right)^2}$$

$$\text{SED} = \sqrt{\frac{1}{\text{N} - 1} \sum \left(y - x - \text{Bias} \right)^2}$$

Descriptive Statistics Computations

Samples and variables can be described by some common statistical measures.

Standard Deviation

The standard deviation of the population from which the values are extracted can be estimated from the data, according to the following formula.

$$S_x(k) = \sqrt{\frac{1}{I_k - 1} \sum_{i=1}^{I} (x_{ik} - \bar{x}_k)^2} \qquad \text{for } k = 1 \ldots K$$

The precision of sample groups is calculated as the standard deviation of the samples in the group.

Histogram Statistics

Skewness and kurtosis are two statistical measures of the asymmetry and flatness, respectively, of an empirical (i.e. observed) distribution.

Skewness

Distributions with a skewness of 0 are symmetrical. Distributions with a positive skewness have a longer tail to the right. Distributions with a negative skewness have a longer tail to the left.

$$\text{Skewness} = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^3}{\left(\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^2}\right)^3}$$

Kurtosis

The reference value for kurtosis is 0; it is the value for the normal distribution N(0,1).

$$\text{Kurtosis} = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^4}{\left(\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_i - \bar{x}\right)^2}\right)^4} - 3$$

Distributions with a kurtosis larger than 0 are more pointed in the middle. Distributions with a kurtosis smaller than 0 are flatter or have thicker tails; this is also the case for symmetrical bi-modal distributions.

Percentiles

The "*u*-percentile" of an x-variable *k* is defined as "the *q*th sample in the *sorted* vector of the I_g samples for x-variable *k*" in a group *g*, where $q = u \cdot I_g$.

For example, the **25% percentile** in a population of **100 samples** is the (0.25·100)th = 25th smallest sample.

The percentiles calculated in The Unscrambler are the following:

- 0% percentile: **Minimum**
- 25% percentile: **Lower Quartile**
- 50% percentile: **Median**
- 75% percentile: **Upper Quartile**
- 100% percentile: **Maximum**.

Effects Computations

Analysis of Effects is based on multiple linear regression (MLR).

The effects are computed as twice the MLR regression coefficients, B. These regression coefficients are based on the coded design data, ie. Low=-1 and High=+1.

Thus, the interpretation of a **main effect** is as follows:

the average change in the response variable when the design variable goes from Low to High.

Significance Testing Computations

Significance testing is used together with MLR-based methods to assess the significance of the estimated b-coefficients. The following results are calculated.

Standard Error of the B-coefficients

$$b_0: \quad \mathbf{bSTDError0}(j) = \hat{\sigma}_j \sqrt{\frac{1}{I_C} + \bar{\mathbf{x}}^{\mathrm{T}} \left(\mathbf{X}_S{}^{\mathrm{T}} \mathbf{X}_S \right)^{-1} \bar{\mathbf{x}}} \qquad j = 1 \ldots J$$

$$b_1 - b_k: \quad \mathbf{bSTDError}(j,k) = \hat{\sigma}_j \sqrt{(\mathbf{X}_S{}^{\mathrm{T}} \mathbf{X}_S)^{-1}{}_{(kk)}} \qquad j = 1 \ldots J, \; k = 1 \ldots K$$

where σ = RMSECal(j)

t-values

$$b_0: \quad \mathbf{tFpValues0}(j, \text{ t - values}) = \frac{\mathbf{B0}(j)}{\mathbf{bSTDError0}(j)} \qquad , j = 1 \ldots J$$

$$b_1 - b_k: \mathbf{tFpValues}(j,k, \text{ t - values}) = \frac{\mathbf{B}(j,k)}{\mathbf{bSTDError}(j,k)} \qquad , j = 1 \ldots J \, , \; k = 1 \ldots K$$

F-ratios

$$b_0: \quad \mathbf{tFpValues0}(j, \text{ F - values}) = \mathbf{tFpValues0}(j, \text{ t - values})^2 \qquad , j = 1 \ldots J$$

$$b_1 - b_k: \mathbf{tFpValues}(j,k, \text{ F - values}) = \mathbf{tFpValues}(j,k, \text{ t - values})^2 \qquad , j = 1 \ldots J \, , \; k = 1 \ldots K$$

p-values

The **2-sided p-values** are derived from the cumulative Fisher F-distribution, using the F-ratio as percentile:

b_0: **tFpValues0**(j, p - values) = P(F > **tFpValues0**(j, F - values))

$b_1 - b_k$: **tFpValues**(j, k, p - values) = P(F > **tFpValues**(j, k, F - values)) , $k = 1 \ldots K$

where $j = 1 \ldots J$. Here, F is a Fisher's F distributed with $\nu 1 = 1$ and $\nu 2 = (I - K - C)$ degrees of freedom.

Multiple Comparisons

When the effect of a variable with more than two levels is found significant, a multiple comparison procedure must be used to determine which levels cause significantly different values of the response. The Unscrambler uses a well-known method for multiple comparisons: Tukey's test.

Comparison with a Scale-Independent Distribution (COSCIND)

The COSCIND method computes a statistic, Ψ value, which is not strictly an F-ratio. This should be remembered for consistency with the other significance testing methods.

The COSCIND Ψ measure is computable in all situations, and is calculated as:

$$\Psi_{jk} = \frac{\left|\tilde{z}_{jk}\right|}{\sqrt{\frac{1}{k-1}\sum_{n=1}^{k-1}\tilde{z}_{jn}^{2}}}$$

where $\tilde{z}_{jk} = z_{j\tilde{k}}$ is the kth *sorted* effect (sorted on decreasing absolute value) for y-variable j. The above expression applies to k=2... K_E. For k=1 (i.e. smallest absolute effect), Ψ is missing.

The approximated p-values are calculated by *Cochran's approximation*:

$$\mathbf{pValEff}(j, k, \text{COSCIND}) = k \cdot \left[1 - \text{betai}\left(\frac{1}{2}, \frac{k-1}{2}, \frac{1}{1+\frac{k-1}{\Psi_{jk}^{2}}}\right)\right] \quad \begin{matrix} , j = 1 \ldots J \\ , k = 1 \ldots K \end{matrix}$$

Here, betai(α, β, x) is the incomplete beta function.

Higher Order Interaction Effects (HOIE)

The F-ratio is found by:

$$\textbf{FValEff}\,(j,k,\ \text{HOIE}) = F_{jk} = \frac{b_{jk}^2}{S_{\text{HOIE,j}}^2 \cdot (\mathbf{X}_S{}^T \mathbf{X}_S)_{(kk)}^{-1}} \quad , j = 1 \ldots J,\ k = 1 \ldots K_E$$

where

$$S_{\text{HOIE},j}^2 = \frac{1}{I_B - K_E - C} \sum_{\substack{i \in \\ \{\text{Cube} \\ \text{samples}\}}} f_{ij}{}^2 \ , j = 1 \ldots J$$

Leverage Calculations

The leverage of an object, a sample or a variable, describes its influential X-"uniqueness" or its actual contribution to the calibration model. A leverage close to zero indicates that the corresponding sample or variable had very little importance for the calibration model.

For MLR, sample leverages are computed according to the following equation:

$$h_i = \frac{1}{I_C} + \mathbf{x}_{S,i}{}^T \left(\mathbf{X}_S{}^T \mathbf{X}_S \right)^{-1} \mathbf{x}_{S,i} \quad , \ i = 1 \ldots I_C$$

For projection methods, sample and variable leverages are computed according to the following equations:

Sample leverages are stored in the Hi matrix.

$$\mathrm{H_i} = \frac{1}{\mathrm{I}} + \sum_{\mathrm{a}=1}^{\mathrm{A}} \frac{\mathrm{t_{ia}^2}}{\mathrm{t_a^T t_a}}$$

Leverages of X-variables are stored in the Hk matrix.

$$\mathrm{H_k} = \sum_{\mathrm{a}=1}^{\mathrm{A}} \frac{\mathrm{w_{ik}^2}}{\mathrm{w_a^T w_a}}$$

The validation method **Leverage Correction** uses the leverages to estimate the prediction error without actually performing any predictions. The correction is done by correcting the y-residuals f with the sample leverage hi:

$$\mathrm{f_{ij}^{corrected}} = \frac{\mathrm{f_{ij}}}{1 - \mathrm{h_i}}$$

Thus, the higher the leverage hi, the larger 1/(1-hi), and extreme samples will have larger prediction residuals than average ones. This is a way to take into account the influence these samples may have on the model.

High Leverage and Outlier Detection

Outlying samples or variables and unusually high leverages are detected in all analyses. The tests listed in the table below are calculated where they are appropriate:

Tests for leverage and outlier detection

Tests	Equation
Leverage limit	$\mathbf{Hi}(a,i) \Big/ \bar{h}_a > C_1$
Ratio of Calibrated to Validated multiple correlation	$\frac{\mathbf{MultCorrCal}(a,j)}{\mathbf{MultCorrVal}(a,j)} > C_3$
Statistical condition number limit	$\gamma >$ C4
Ratio of Calibrated to Validated explained variance	$\frac{\mathbf{ExpXCalTot}(a)}{\mathbf{ExpXValTot}(a)} < C_5$
Total explained variance	$\mathbf{ExpXCalTot(a)} < C_6$
Ratio of Validated to Calibrated multiple correlation	$\frac{\mathbf{MultCorrVal}(a,j)}{\mathbf{MultCorrCal}(a,j)} > C_7$
Sample Outlier limit, Calibration	$\frac{\lvert\mathbf{Eix}(i,k,a)\rvert}{\sqrt{\mathbf{ResXCalVar}(a,k)}} > C_8$
Sample Outlier limit, Validation	$\frac{\lvert\mathbf{Eix}(i,k,a)\rvert}{\sqrt{\mathbf{ResXValVar}(a,k)}} > C_9$
Variable Outlier limit, Calibration	$\sqrt{\frac{\mathbf{ResYCalVar}(a,j)}{\mathbf{ResYCalTot}(a)}} > C_1$
Variable Outlier limit, Validation	$\sqrt{\frac{\mathbf{ResYValVar}(a,j)}{\mathbf{ResYValTot}(a)}} > C_1$
Ratio of Validated to Calibrated explained variance	$\frac{\mathbf{ExpXValTot}(a)}{\mathbf{ExpXCalTot}(a)} < C_{12}$

Exchange X and Y in the matrices above according to the context in which the leverage and outlier detection is done.

The constants “C” that are used as test limits are set in the **Warning Limits** dialog available in all model dialogs from the **Task** menu. A warning is given when the calculated value is higher/lower than the limit.

All issued warnings are found in the Outlier List.

Warning Limits and Outlier Warnings

The warning limits in The Unscrambler are listed in the table hereafter. In principle, more than one limit may be applied for each test formula to distinguish between degrees of severity, eg. “badly described” vs. “very badly described”. However, only one constant is used to keep things as simple as possible.

Constants in The Unscrambler

Constant	Default value	Allowed range	Comment
C_1	3.0	2 – 10	Leverage limit
C_2	6%	0% – 15%	Residual variance increase limit
C_3	2.0	1.5 – 5.0	Ratio of Calibrated to Validated multiple correlation
C_4	50	10 – 1000	Statistical condition number limit
C_5	0.5	0.2 – 0.7	Ratio of Calibrated to Validated residual variance
C_6	20%	5% – 90%	Total explained variance
C_7	1.3	1.0 – 3.0	Ratio of Validated to Calibrated multiple correlation
C_8	3.0	2 – 10	Sample Outlier limit, Calibration
C_9	2.6	2 – 10	Sample Outlier limit, Validation
C_{10}	3.0	2 – 10	Variable Outlier limit, Calibration
C_{11}	3.0	2 – 10	Variable Outlier limit, Validation
C_{12}	0.75	0.5 – 1.0	Ratio of Validated to Calibrated residual variance
C_{13}	3.0	2 - 10	Individual Value Outlier, Calibration
C_{14}	2.6	2 - 10	Individual Value Outlier, Validation

The table below shows which object warning each warning limit is used in. It also shows upon which samples (cal/val) and variables the test is based. The sequence is as shown in the user dialog for Warning Limits (PLS1).

Object warnings and warning limits

Constant	Cal. Samples	Val. Samples	X	Y	Object Warnings (OW)
C_1	X	X			100,102
C_8	X		X	X	121,123,102
C_9		X	X	X	122,124
C_{13}	X		X	X	130,140
C_{14}		X	X	X	131,141
C_{10}	X		X	X	150,160
C_{11}		X	X	X	151,161
C_6	X	X	X	X	152,153,162,163
C_5			X	X	170,171

C_{12}			X	X	172,173
C_2		X	X	X	180,181
C_3					190
C_4					200,201,202,205
C_7					191

The next table shows which warning limit is used in connection with which analysis.

Warning limits and analyses

Const.	STA	PCA	PCR	PLS	MLR	AoE	RS	PRE	CLA
C_1		✓	✓	✓	✓			✓	
C_2		✓	✓	✓					
C_3					✓				
C_4					✓		✓		
C_5		✓	✓	✓	✓				
C_6		✓	✓	✓				✓	
C_7					✓				
C_8		✓	✓	✓	✓	✓	✓		
C_9		✓	✓	✓	✓			✓	
C_{10}		✓	✓	✓	✓	✓	✓		
C_{11}		✓	✓	✓	✓				
C_{12}		✓	✓	✓	✓				
C_{13}		✓	✓	✓	✓	✓	✓		
C_{14}		✓	✓	✓	✓			✓	

Hotelling T2 Computations

The Hotelling statistic is a multivariate t-statistic, and for an object i, it is given by

$$\hat{Var}(t_a)$$

where $\hat{Var}(t_a)$ is the estimated variance of t_a. There is a relationship between T^2 and the F-statistic given by the expression

$$F \approx T_i^2 * I(I - A) / A(I^2 - 1)$$

which is F distributed with A and I-A degrees of freedom

I = Total number of observations in the model training set

A = Used number of components in the model

Then, for an observation *i*, this observation is outside the critical limit if

$$T_i^2 > A(I^2 - 1) / I(I - A) \cdot F_{critical}, \alpha$$

The α value is commonly set to 0.05. The confidence region for a two-dimensional score plot is an ellipse with axis

$$(\hat{Var}(t_a) \cdot F_{2,I-2,\alpha} \cdot 2(I^2 - 1) / (I(I - 2)))^{0.5}$$

where a is [1 2] for the (1,2) score plot.

The T2 statistic for each sample and each PC, together with the critical limits are stored in the result file.

Deviation in Prediction

The Unscrambler gives you an estimate of how reliable the predicted values are. This estimate is calculated as

$$\text{yDeviation} = \sqrt{\text{ResYValVar}\left(\frac{\text{ResXValSamp}_{\text{pred}}}{\text{ResXValTot}} + \text{Hi} + \frac{1}{\text{I}_{\text{cal}}}\right)\left(1 - \frac{\text{a} + 1}{\text{I}_{\text{cal}}}\right)}$$

Note: The deviation calculated here is based on an approximation of the theoretical variance of predictions under certain assumptions. It has recently been improved from a previous formula (implemented in earlier versions of The Unscrambler) to make it more robust.

You will find a detailed reference about this equation in De Vries and Ter Braak, "Prediction Error in Partial Least Squares Regression: A Critique on the Deviation used in The Unscrambler", see Bibliographical References for details.

Classification Statistics

Various statistics are calculated in SIMCA classification to distinguish between members and non-members of the different classes.

Sample to Model Distance

The distance from sample i to the model m is calculated as the orthogonal distance from the sample down to the different classes defined by their principal components.

New Samples

The distance of new samples to the class model m is computed as

$$S_i(m,i) = \sqrt{\mathrm{ResXCal}_{new,m}(a,i)}$$

Calibration Samples

The distance of calibration samples from model q projected onto model m is computed as

$$S_i(m,i) = \sqrt{\frac{1}{d_7}\sum \mathrm{Eix}_{q,m}(i,k)^2}$$

The distance of each sample i in model q to the class m is computed as

$$\mathrm{ModelDist}_q(m,i) = \sqrt{d \cdot \mathrm{ResXCal}_{q,m}(a,i)}$$

where

$$q = m: \quad d = \frac{d_7}{d_6} = \frac{(K-a)I_q}{I_q K - CK - a\max\left[I_q - C, K\right]}$$

$$q \neq m: \quad d = 1$$

Model Distance

The model distance is the distance between pairs of classes in the classification. The distance between two models, q and m, is calculated as follows:

$$\mathrm{ModelDistance}(q,m) = \frac{\frac{1}{K_q}\sum_{k=1}^{Kq} S_q(m,k)^2 + \frac{1}{K_m}\sum_{k=1}^{Km} S_m(q,k)^2}{\frac{1}{K_m}\sum_{k=1}^{Km} S_m(m,k)^2 + \frac{1}{K_q}\sum_{k=1}^{Kq} S_q(q,k)^2}$$

where

$S_q(m,k)$ is the standard deviation for variable k when fitting samples from model q onto model m, as an example.

$$q = m: \quad S_q(m,k)^2 = \mathrm{ResXCal}(a,k)$$

$$q \neq m: \quad S_q(m,k)^2 = \frac{1}{d_3}\sum_{i=1}^{I} \mathrm{Eix}(i,k)^2$$

Discrimination Power

The discrimination power expresses how well a variable discriminates between different classes. The discrimination power for variable k between model q and m (fitting samples from model q onto model m) is computed as

$$\text{DiscrPower} = \frac{S_q(m,k)^2 + S_m(m,k)^2}{S_m(m,k)^2 + S_q(q,k)^2}$$

Modeling Power

The modeling power describes the relevance of variable for one class. The modeling power for variable k in model m is computed as

$$\text{ModelPower} = 1 - \sqrt{\frac{\text{ResXCalVar}(a,k)}{\text{ResXCalVar}(0,k)}}$$

Class Membership Limits

Two limits are used to decide whether a sample belongs to a certain class or not.

Leverage Limit

New samples are found to be within the leverage limits for a class model if

$$\text{HiClass}(m) \le 3 \cdot \frac{a+1}{I_c}$$

Sample to Model Distance Limit, S_{max}

The sample to model distance limit S_{max} for classifying new samples is calculated differently depending on the validation method used for the class model **m**:

Leverage correction: $S_{max}(m) = S_0(m)\sqrt{F_{crit}(1, I - A - C)}$

Cross validation: $S_{max}(m) = S_0(m)\sqrt{F_{crit}(1, I_{cal})}$

Test set: $S_{max}(m) = S_0(m)\sqrt{F_{crit}(1, I_{Test})}$

Where S_0 is the average distance within the model:

$$S_0(m) = \sqrt{ResXValTot(a)}$$

and C = 1 for centered models,

0 otherwise

Computation of Transformations

This chapter contains the formulas for most transformations implemented in The Unscrambler. For some simple transformations (e.g. Average) not listed here, lookup the corresponding menu option (e.g. **Modify - Transform - Reduce (Average)**) using the Index tab in the Unscrambler Help System.

Smoothing Methods

The Unscrambler offers two kinds of smoothing; Moving Average and Savitzky-Golay smoothing.

Moving Average Smoothing

For each point of the curve, a moving average is computed as the average over a segment encompassing the current point. The individual values are replaced by the corresponding moving averages.

Savitzky-Golay Smoothing

The Savitzky-Golay algorithm fits a polynomial to each successive curve segment, thus replacing the original values with more regular variations. You can choose the length of the segment (right and left of each point) and the order of the polynomial. Note that a first-order polynomial is equivalent to a moving average.

The complete algorithm for Savitzky-Golay smoothing can be found in Press, Tewkolsky, Vetterling and Flannery (see Bibliographical References for details).

Normalization Equations

The Unscrambler contains three normalization methods: mean, maximum, and range normalization.

Mean Normalization

$$X(i,k) = \frac{X(i,k)}{|\bar{X}(i,\bullet)|}$$

Maximum Normalization

$$X(i,k) = \frac{X(i,k)}{\max\left(\left|X(i,\bullet)\right|\right)}$$

Range Normalization

$$X(i,k) = \frac{X(i,k)}{\max(i,\bullet) - \min(i,\bullet)}$$

Spectroscopic Transformation Equations

Transformations often needed for spectra are given here.

Reflectance to Absorbance Transformation

We assume that the instrument readings R (Reflectance) or T (Transmittance) are expressed in fractions between 0 and 1. The readings may then be transformed to apparent Absorbance (Optical Density) according to this equation.

$$M_{new}(i,k) = \log(\frac{1}{M(i,k)})$$

Absorbance to Reflectance Transformation

An absorbance spectrum may be transformed to Reflectance/ Transmittance according to this equation.

$$M_{new}(i,k) = 10^{-M(i,k)}$$

Reflectance to Kubelka-Munk Transformation

A reflectance spectrum may be transformed into Kubelka-Munk units according to this equation.

$$M_{new}(i,k) = \frac{(1 - M(i,k))^2}{2 \cdot M(i,k)}$$

Multiplicative Scatter Correction Equations

Multiplicative Scatter Correction (MSC) is a specific transformation for spectra. It consists in fitting a separate regression line to each sample spectrum, expressed as a function of the average value for each wavelength; the a and b coefficients of that regression line are then used to correct the values of each sample.

Full MSC: $$M_{new}(i,k) = \frac{M(i,k) - a}{b}$$

Common Offset: $$M_{new}(i,k) = M(i,k) - a$$

Common Amplification: $$M_{new}(i,k) = \frac{M(i,k)}{b}$$

Added Noise Equations

Proportional and additive noise can be added to selected variables. Proportional noise is typically noise that affects the instrumental amplification. Additive noise is typically noise that affects the measurement signal.

The formula for adding noise is

$$M_{new}(i,k) = M(i,k) \cdot \left[1 + \frac{PN}{100} \cdot N0,1)\right] + N(0, AN)$$

where

PN = Level of proportional noise in %
AN = Level of additive noise
N(m,s) = randomly distributed value with m = mean and s = standard deviation.

The amount of additive noise (AN) depends on the level of the measurement value. You may calculate this value by

$$AN = \frac{P\%}{100} \cdot M(i,k)$$

where

P% is the level of approximate additive noise in percent.

Differentiation Algorithm

The differentiation of a curve (i.e. computing derivatives of the underlying function) requires that the curve is continuous (no missing values are allowed in the Variable Set that is to be differentiated).

Savitzky-Golay Differentiation

The Savitzky-Golay algorithm fits a polynomial to each successive curve segment, thus replacing the original values with more regular variations. You can choose the length of the segment (right and left of each point) and the order of the polynomial. Note that a first-order polynomial is equivalent to a moving average.

The complete algorithm for Savitzky-Golay differentiation can be found in Press, Tewkolsky, Vetterling and Flannery (see Bibliographical References for details).

Mixture and D-Optimal Designs

This chapter addresses the case of non-orthogonal designs, for which the shape of the experimental region is essential in determining which points to include in the design.

Shape of the Mixture Region

The combinations of levels of mixture variables are always located on a simplex. Depending on the nature of the ranges of variations and additional multi-linear constraints, the mixture region may either be a simplex, or have a more complex shape.

Notations

The notations below are used in this section to express formulas and algorithms.

Notations for mixtures

Symbol	Description
q, Q	Mixture variable no. and no. of mixture variables
U	Upper bound of a variable
L	Lower bound of a variable
$MixSum$	Mixture sum of the mixture variables

Simplex Region

If the following criterion is true and there are no multi-linear constraints, then the mixture region is a simplex:

For every U_q

$$MixSum - U_q \leq \sum_{i \neq q} L_i$$

Upper/Lower Bound Consistency

The mixture region is said to be inconsistent if any of the variables' upper or lower bounds are unattainable because of other variables constraints. If an inconsistent mixture region is non-empty, it can be made consistent by lowering inconsistent upper bounds and raising inconsistent lower bounds until they touch the region border.

The region is non-empty if

$$\sum U_q > MixSum \qquad \text{and} \qquad \sum L_q < MixSum$$

If the region is empty by the first criteria, the user is asked to raise the upper constraints of some of variables. If the region is empty by the second criteria, the user must lower the lower constraints of some variables.

Multi-linear constraints are checked for consistency by running the CONVRT algorithm to generate extreme vertex points. If there are inconsistent constraints, the algorithm will return an error code.

Computation of Candidate Points for a D-Optimal Design

Generating candidate points for D-optimal design is based on the algorithms CONVRT and CONAEV (CORNELL, 1990).

The CONVRT algorithm determines the extreme vertices mixture regions with linear constraints. The CONAEV algorithm calculates the centroids of all N-dimensional surfaces of the region, where 0<N<Q. The centroid points are calculated as the average (mean) of the extreme vertices on the design region surface associated with the centroid point.

D-Optimal Selection of Design Points

The D-optimal selection of a design point is based on the algorithm DOPT (MILLER and NGUYEN, 1994).

The FORTRAN algorithm DOPT is used to D-optimally select a number of design points from a set of candidate points. The design points are chosen so that the determinant of X'X is maximized (X is the design point matrix).

DOPT:

6. Start with a default set of design points, or else generate a random set.
7. Calculate X'X for the selected points.
8. As long as there is improvement do:
   a. Find the selected point and unselected candidate point which will improve the X'X determinant the most.
   b. Exchange the points, and calculate new X'X.

Bibliographical References

This chapter contains bibliographical references for the methods and algorithms used in The Unscrambler.

About Statistics and Multivariate Data Analysis

- Albano C, Dunn III W, Edlund U, Johansson E, Nordén B, Sjöström M, Wold S, Four levels of pattern recognition, Anal. Chim. Acta, 1978, 103, 429 - 443
- Beebe KR, Kowalski BR, *An introduction to multivariate calibration and analysis*, Anal. Chem., 1987, **57**(17) 1007A - 1017A
- Box GEP, Hunter WG, Hunter JS, “Statistics for experimenters”, Wiley & Sons Ltd, New York, 1978
- Brown SD, Blank TB, Sum ST, Weyer LG, *Chemometrics*, Anal. Chem., 1994, **66**, 315R - 359R
- De Vries S, Ter Braak Cajo JF, Prediction error in partial least squares regression: a critique on the deviation used in The Unscrambler, Chemometrics and Intelligent Laboratory Systems, 1993, **30**, 239 - 245
- Deming SN, Palasota JA, Nocerino JM, *The geometry of multivariate object preprocessing*, Jour. Chemometrics, 1993, **7**, 393 - 425
- Draper NR, Smith H, “Applied regression analysis”, John Wiley & Sons, Inc, New York, 1981

- Esbensen K, "Multivariate Data Analysis - In Practice", ISBN 82-993330-3-2, CAMO Process AS, Oslo, 5th Edition 2002
- Fisher RA, The use of multiple measurements in taxonomic problems, Ann. Eugenics, 1936, **7**, 179 - 188
- Forina M, Drava G, Boggia R, Lanteri S, Conti P, *Validation procedures in near-infrared spectrometry*, Anal. Chim. Acta, 1994, **295**, 1-2, 109 - 118
- Frank IE, Friedman JH, *A statistical view of some chemometrics tools*, Technometrics, 1993, **35**, 109 - 148
- Geladi P, Kowalski BR, *Partial least-squares regression: a tutorial*, Anal. Chim. Acta, 1986, **185**, 1 - 17
- Golub GH, Loan CF van, "Matrix Computation", The John Hopkins University Press, London, 1989, 2nd ed
- Höskuldsson A, *PLS regression methods*, Jour. Chemometrics., 1988, **2**, 211 - 228
- Jackson JE, "A User's Guide to Principal Components", Wiley & Sons Inc., New York, 1991
- Johnson RA, Wichern DW, "Applied multivariate statistical analysis", 1988, Prentice-Hall, 607 p
- Manne R, Analysis of two partial least squares algorithms for multivariate calibration, Chemom. Intell. Lab. Syst., 1987, **2**, 187 - 197
- Mardia KV, Kent JT, Bibby JM, "Multivariate Analysis", Academic Press Inc, London, 1979
- Martens H, Næs T, "Multivariate Calibration", John Wiley & Sons, Inc, Chichester, 1989
- Massart PL, Vandegiste BGM, Deming SN, Michotte Y, Kaufman L, "Chemometrics: A text book", Elsevier Publ., Amsterdam, 1988
- Wold S, Cross-validatory estimation of the number of components in factor and principal components models, Technometrics, 1978, **20**(4), 397 - 405
- Wold S, Esbensen K, Geladi P, *Principal component analysis - A tutorial*, Chemom. Intell. Lab. Syst., 1987, **2**, 37 - 52
- Wold S, Pattern recognition by means of disjoint principal components models, Pattern recognition, 1976, **8**, 127 - 139

About Martens' Uncertainty Test

- EFRON B., - 1982 -, *The Jackknife, the Bootstrap and Other Resampling Plans*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, ISBN 0-89871-179-7
- MARTENS H., MARTENS M., - 1999 -, Modified Jack-knife Estimation of Parameter Uncertainty in Bilinear Modelling (PLSR), Food Quality and Preference.
- MARTENS H,. MARTENS M., - 1999 -, *Validation of PLS Regression models in sensory science by extended cross-validation)*, PLS'99 (Proceedings, International Symposium on PLS Methods, Paris Oct. 5-6, 1999.
- WESTAD F., BYSTRÖM M., MARTENS H., - 1999 -, Modified Jack-knifing in multivariate regression for variable selection and model stability, NIR-99 (Proceedings, International Symposium on NIR Spectroscopy, Verona June 13-18, 1999.
- WESTAD F., - 1999 -, Relevance and Parsimony in Multivariate Modelling, Ph.D. Thesis, University NTNU Trondheim, Trondheim, Norway.

About Three-way Data and Tri-PLS

- Bro R., "Multiway calibration. Multilinear PLS", in Journal of Chemometrics 10 (1):47-61, 1996
- Bro R., - 1998 -, Multi-way analysis in the food industry. Models, algorithms and applications - PhD thesis, University of Amsterdam, Netherlands
- Bro R., Smilde A. K. and de Jong S., "On the difference between low-rank and subspace approximation: improved model for multi-linear PLS regression", in Chemom. Intell. Lab. Syst. 58 (1):3-13, 2001
- Bro R. and Smilde A. K., "Centering and scaling in component analysis", in Journal of Chemometrics 17:16-33, 2003
- Kiers H. A. L., "Towards a standardized notation and terminology in multiway analysis", in Journal of Chemometrics 14 (3):105-122, 2000
- Sanchez E. and Kowalski B. R., "Tensorial calibration: I. first-order calibration", in Journal of Chemometrics 2:247-263, 1988

About Experimental Design

- Box GEP, Hunter WG, Hunter JS, "Statistics for experimenters", Wiley & Sons Ltd, New York, 1978
- Carlson R, "Design and optimization in organic synthesis", Elsevier, Amsterdam, 1992
- Cornell J. A., - 1990 -, *Experiments with Mixtures*, John Wiley & Sons, Inc., 2nd ed., 632p.
- Esbensen K, "Multivariate Data Analysis - In Practice", ISBN 82-993330-3-2, CAMO Process AS, Oslo, 5th Edition 2002
- Langsrud Ø, Ellekjær MR, Næs T, *Identifying significant effects in fractional factorial experiments*, Jour. Chemometrics, 1994, **8**, 205-219.
- Montgomery DC, "Design and analysis of experiments", John Wiley & Sons, Inc, New York, 1991
- Morgan E, "Chemometrics: Experimental design", John Wiley & Sons Ltd, 1991

About Numerical Algorithms

- Miller A. J., Nguyen, N-K., - 1994 - *A Fedorov exchange algorithm for D-optimal design*, Applied Statistics, Vol. 43, p. 669-678.
- Press WH, Tewkolsky SA, Vetterling WT, Flannery BP, "Numerical Recipes in Fortran: The art of scientific computing", Cambridge University Press, 1992, 2nd edition
- Savitzky A, Golay MJE, Smoothing and differentiation of data by simplified least squares procedures, Anal. Chem., 1964, **36**, 1627 - 1639

Index

2-variable statistics 22
3-dimensional matrix 3
absorbance to reflectance.....35
algorithm
  MLR 5
  n-PLS.....12
  PCA 5
  PCR 7
  PLS1 8
  PLS2 10
  tri-PLS 12
algorithms 4
bias 21
candidate points
  computation 38
centering..... 15

class membership limits....33
classification
  statistics....31
correlation....23
COSCIND....26
data
  centering....15
degrees of freedom....18
derivatives....36
  Savitzky-Golay....37
descriptive statistics....23
deviation in prediction....31
dialog
  Warning Limits....28
differentiation....36
  Savitzky-Golay....37
discrimination power....33
effects....25
equation
  MLR....4
  N-PLS....11
  PCA....5
  PCR....7
  PLS....8
error measures....18
experimental region
  shape for mixtures....37
explained variance....20
f-ratio....25
higher order interaction effects....26
histogram
  statistics....23
HOIE....26
Hotelling T2....30
interaction and square effects....16
interaction effects....16
interactions....16
Kubelka-Munk....35
kurtosis....24
leverage....27; 28
lower bound....38
maximum normalization....35
mean normalization....35
MLR
  algorithm....5
  equation....4
model
  equations....4
model distance....32
modeling power....33
moving average....34
MSC....36
multiple comparisons....26
multiplicative scatter correction....36
NIPALS algorithm....6
noise
  added....36
normalization....34
n-PLS
  algorithm....12
N-PLS
  equation....11
outlier
  list....28
outlier detection....28
outlier warnings....29
OW 29
PCA
  algorithm....5
  equation....5
PCR
  algorithm....7
  equation....7
percentiles....24
PLS
  equation....8
PLS1
  algorithm....8
PLS2
  algorithm....10
preliminary scores....3
p-value....26
range normalization....35
reflectance to absorbance....35
reflectance to Kubelka-Munk....35
regression
  statistics....22
residual variance....19
residuals....18; 19
ResYCalSamp....19
ResYValSamp....19
RMSEC....18; 21
RMSED....23
RMSEP....18; 21
root mean square error....21

root mean squared error of deviation..... *See* RMSED
Savitzky-Golay 34; 37
scalar 3
scores
  preliminary 3
SED 23
SEP 21
Si 32
significance testing 25
SIMCA 31
simplex region 37
singular value decomposition 5
skewness 24
smoothing 34
  moving average 34
  Savitzky-Golay 34
spectroscopic transformations 35
square effects 16
standard
  error of deviation 23
  error of prediction 21
  error of the B-coefficients 25
standard deviation 23
studentized residuals 22
SVD 5
total residual variance 20
tri-PLS
  algorithm 12
t-values 25
upper bound 38
variances 18
vector 3
warning limits 29
Warning Limits dialog 28