Multivariate statistical methods for the analysis, monitoring and optimization of processes

Jay Liu
Dept. of Chemical Engineering
Pukyong National University

3. Partial Least Squares

We will cover ….

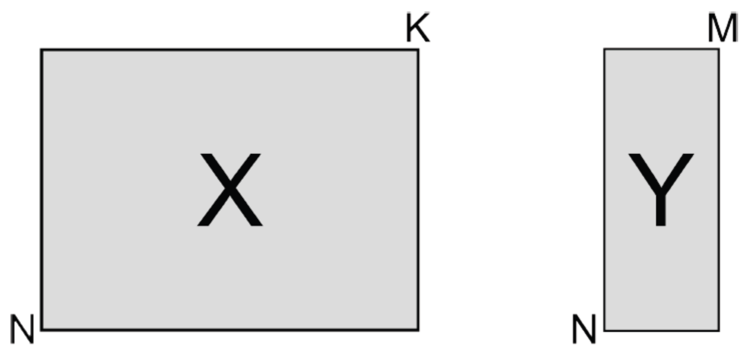
- Multiple Linear Regression (least squares) [MLR]
- Principal Component Regression [PCR]
- Partial Least Squares or Projection to Latent Structures [PLS]

Quantitative modeling

- Relationships between two sets of multivariate data, **X** and **Y**
  - In process modeling and optimization
    - Process variables ↔ yield / quality
  - Chemical composition ↔ quality
    physical measurements biological activity
  - Chemical structure ↔ reactivity properties
    biological activity
  - In multivariate calibration
    signals (spectra) ↔ concentrations
    energy contents, etc

Quantitative modeling

- Starting point



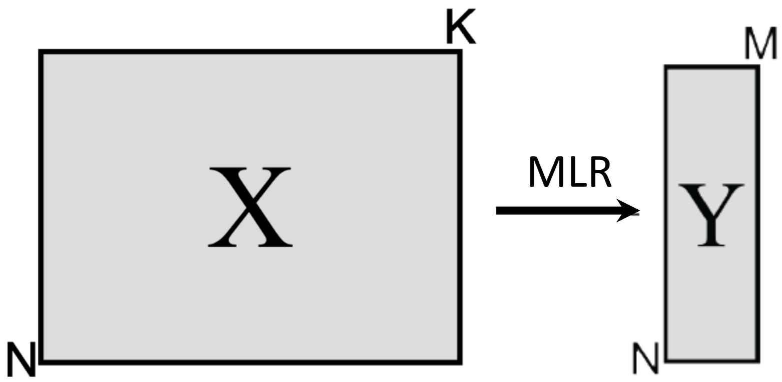
  - Data set = tables (matrices) of N rows (objects, samples, …) and K & M columns (variables, properties, …)

Objects (cases, samples, rows, …)
- Analytical samples
- Process time points
- Trials (experiment runs)

Variables (tags, properties, columns, …)
- Sensors (T, P, flow, pH, conc., …)
- Spectra, chromatograms, …
- quality measures, yields, costs, …

X: what is “always” available
Y: what is “not always” available

Multiple linear regression



$$\hat{\mathbf{Y}} = \mathbf{Xb}$$

$$\mathbf{B} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y}$$

- Can’t remember?
- Let’s review engineering statistics

Multiple linear regression

- Matrix representation of MLR (M=1)

$$y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_m x_m + e$$

$$\Rightarrow \quad \mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{m1} \\ 1 & x_{12} & \cdots & x_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1n} & \cdots & x_{mn} \end{bmatrix} \qquad \begin{aligned} \mathbf{y}^T &= \lfloor y_1 \ \ y_2 \ \ \cdots \ \ y_n \rfloor \\ \mathbf{b}^T &= [b_0 \ \ b_1 \ \ \cdots \ \ b_m] \\ \mathbf{e}^T &= \lfloor e_1 \ \ e_2 \ \ \cdots \ \ e_n \rfloor \end{aligned}$$

m+1: number of coefficients
n: number of data points

Multiple linear regression

- Example
  - Fitting quadratic polynomials to five data points

x	-1.0	-0.5	0.0	0.5	1.0
y	1.0	0.5	0.0	0.5	2.0

$$y = b_0 + b_1 x + b_2 x^2 + e$$

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}$$

$$\begin{bmatrix} 1.0 \\ 0.5 \\ 0.0 \\ 0.5 \\ 2.0 \end{bmatrix} = \begin{bmatrix} 1 & -1.0 & 1.0 \\ 1 & -0.5 & 0.25 \\ 1 & 0.0 & 0.0 \\ 1 & 0.5 & 0.25 \\ 1 & 1.0 & 1.0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{bmatrix}$$

Three unknowns
Five equations

Multiple linear regression

- Solutions

Sum of squares of errors

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}$$

$$S_r = \sum e_i^2 = \mathbf{e}^T\mathbf{e} = (\mathbf{y} - \mathbf{X}\mathbf{b})^T (\mathbf{y} - \mathbf{X}\mathbf{b})$$

$$\frac{\partial S_r}{\partial \mathbf{b}} = 0 \longrightarrow (\mathbf{X}^T\mathbf{X})\mathbf{b} = \mathbf{X}^T\mathbf{y}$$

- 1. LU decomposition or other methods to solve L.A.E

$$(\mathbf{X}^T\mathbf{X})\mathbf{b} = \mathbf{X}^T\mathbf{y} \quad \Rightarrow \text{"}\mathbf{A}\mathbf{x} = \mathbf{b}\text{"}$$

- 2. Matrix inversion

$$(\mathbf{X}^T\mathbf{X})\mathbf{b} = \mathbf{X}^T\mathbf{y} \quad \Rightarrow \mathbf{b} = \boxed{(\mathbf{X}^T\mathbf{X})^{-1}}\mathbf{X}^T\mathbf{y}$$

What if **X**'s are correlated

- High/no correlation between x_1 and x_2

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 1.0000 & 0.9999 \\ 0.9999 & 1.0000 \end{bmatrix}$$

$$\left(\mathbf{X}^T\mathbf{X}\right)^{-1} = \begin{bmatrix} 5000.25 & -4999.75 \\ -4999.75 & 5000.25 \end{bmatrix}$$

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 1.0000 & 0.0 \\ 0.0 & 1.0000 \end{bmatrix}$$

$$\left(\mathbf{X}^T\mathbf{X}\right)^{-1} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

- What if very small (measurement) noises added to **X**

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 1.0001 & 0.9999 \\ 0.9999 & 1.0000 \end{bmatrix}$$

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} 1.0000 & 0.0 \\ 0.0 & 1.0001 \end{bmatrix}$$

- What will happen to your MLR model?

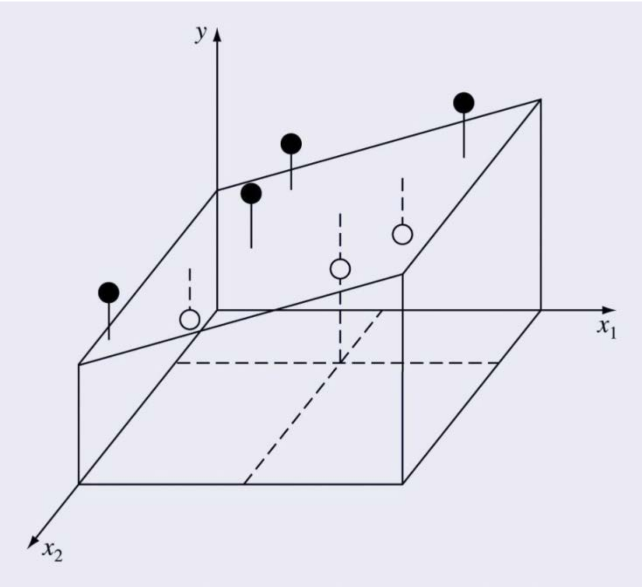
What if **X**’s are correlated

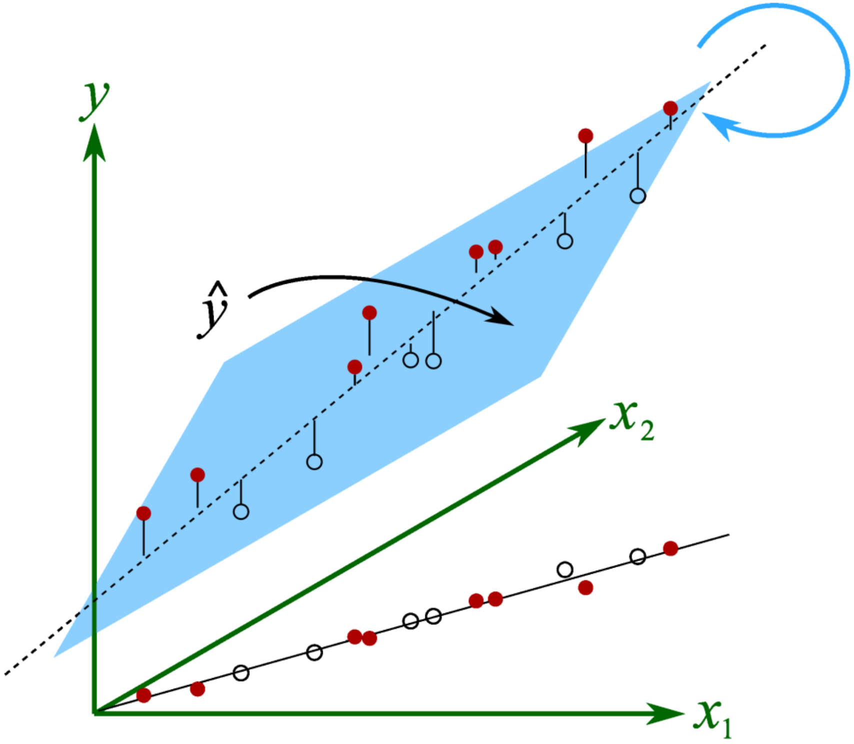
- If high correlation among columns in **X**:
  - unstable solutions for **b**
  - predictions uncertain also
- What to do about it?
  - Select uncorrelated columns from **X**
- Other issues:
  - **X** has (measurement) error; MLR assumes it doesn't.
  - MLR cannot handle missing values

PCR and PLS can avoid these drawbacks.

What if **X**'s are correlated

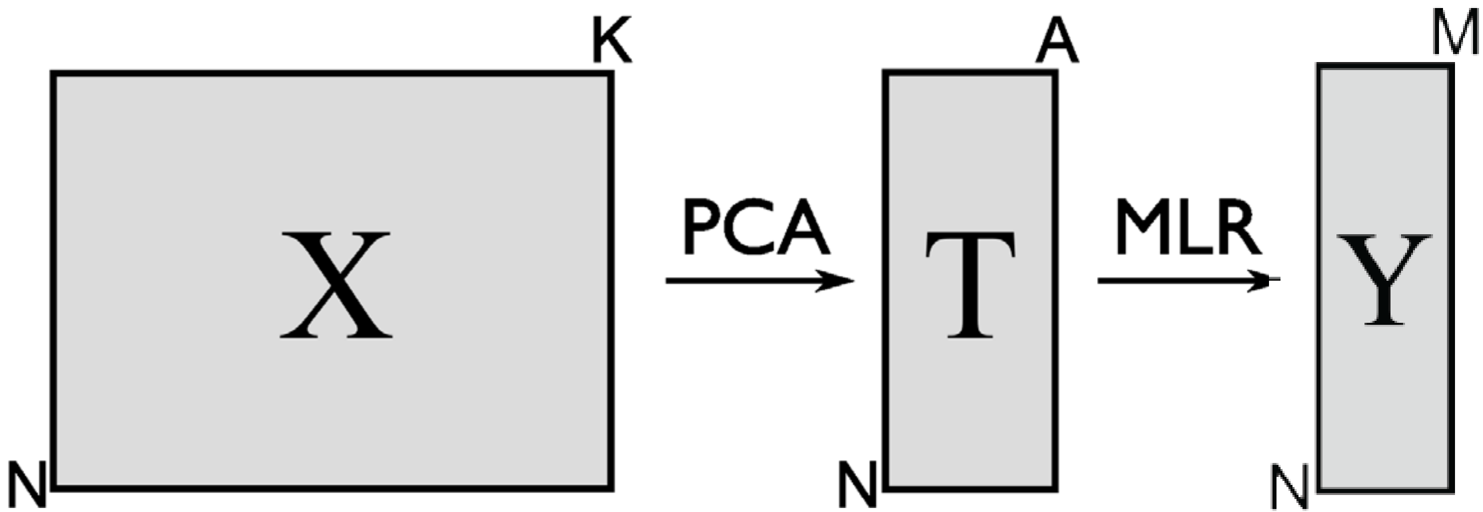
- Geometrically speaking





High correlation between x_1 and x_2

Principal component regression



Two step model:

1. $\mathbf{T} = \mathbf{XP}$
2. $\hat{\mathbf{Y}} = \mathbf{TB}$ and $\mathbf{B}$ can be calculated as, $\mathbf{B} = \left(\mathbf{T}^T\mathbf{T}\right)^{-1}\mathbf{T}^T\mathbf{Y}$

Principal component regression

- Two blocks, **X** and **Y**
- Objective: model both X and Y and the relationship between **X** and **Y**

- **X** summarized by PC scores (t's) in matrix **T**

T = XP

- PC scores used as independent variables in MLR

$$\hat{\mathbf{Y}} = \mathbf{Tb}, \text{ where } \mathbf{B} = \left(\mathbf{T}^T\mathbf{T}\right)^{-1}\mathbf{T}^T\mathbf{Y}$$

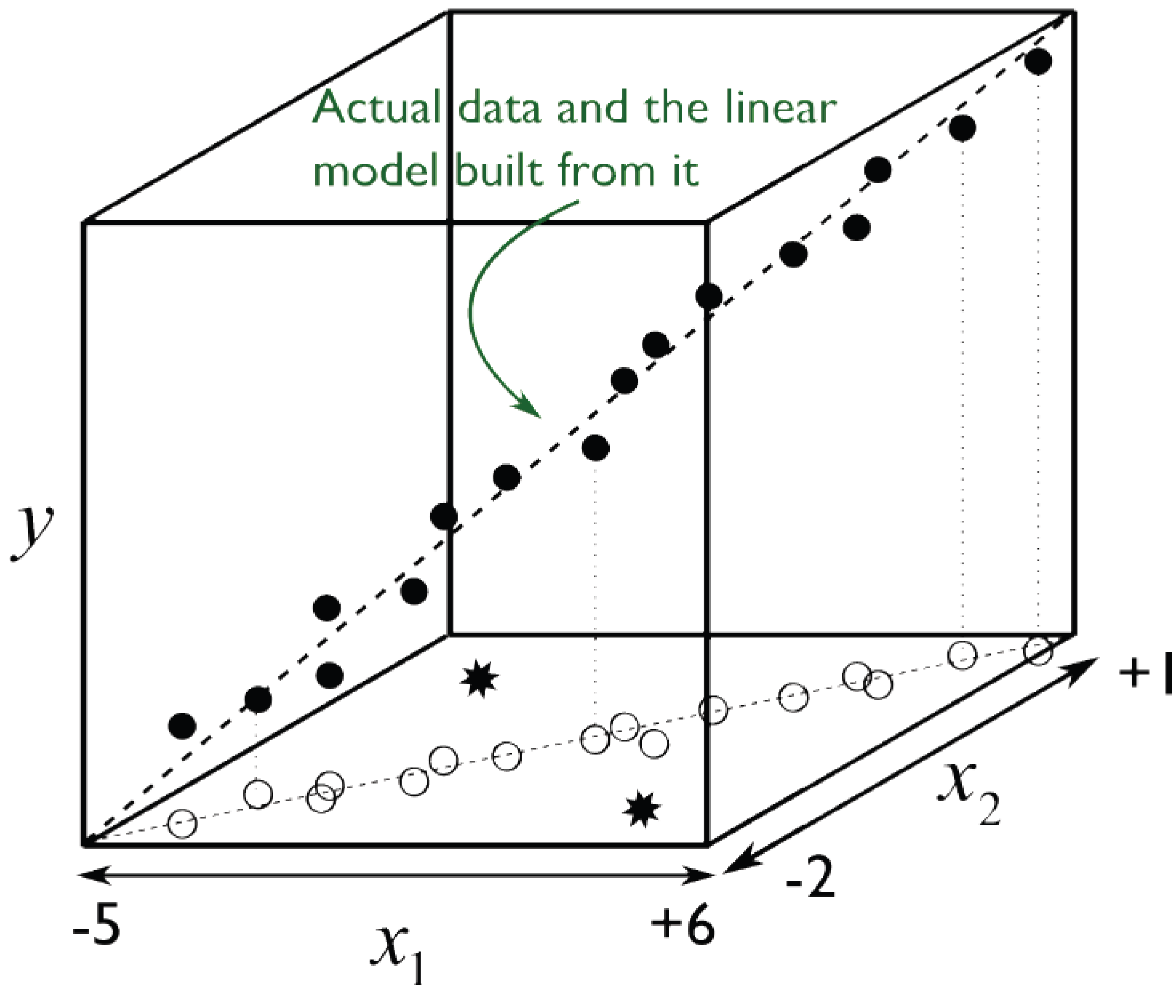
Principal component regression

- Building PCR model
  - Indirect modeling (no direct modeling between **x**'s and **y**'s)

$$\mathbf{T} = \mathbf{X}\mathbf{P}$$
$$\hat{\mathbf{Y}} = \mathbf{T}\mathbf{b}, \text{ where } \mathbf{B} = \left(\mathbf{T}^{T}\mathbf{T}\right)^{-1}\mathbf{T}^{T}\mathbf{Y}$$

- Advantages:
  - Columns in **T** are orthogonal
  - Can Handle missing values
  - **T** has much less error than **X**
  - Less need for variable selection

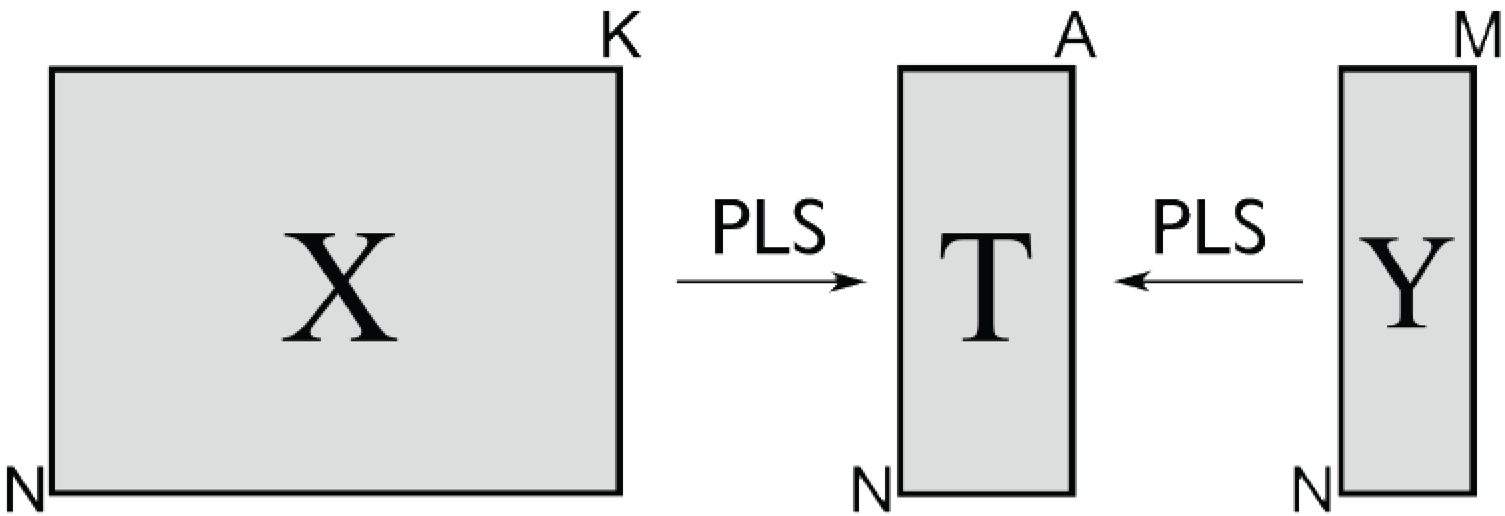
Principal component regression



Principal component regression

- Using a PCR model: can check consistency before predicting **y**'s
  - Check SPE_{new}
  - Check T^2_{new}
- Projection to latent structures (PLS) aka Partial least squares
  - better alternative to PCR
  - Indirect modeling (inner model and outer model)
  - Idea?
    - Build **a regression model between scores** of **X** and **Y**.

Projection to Latent Structures (PLS)



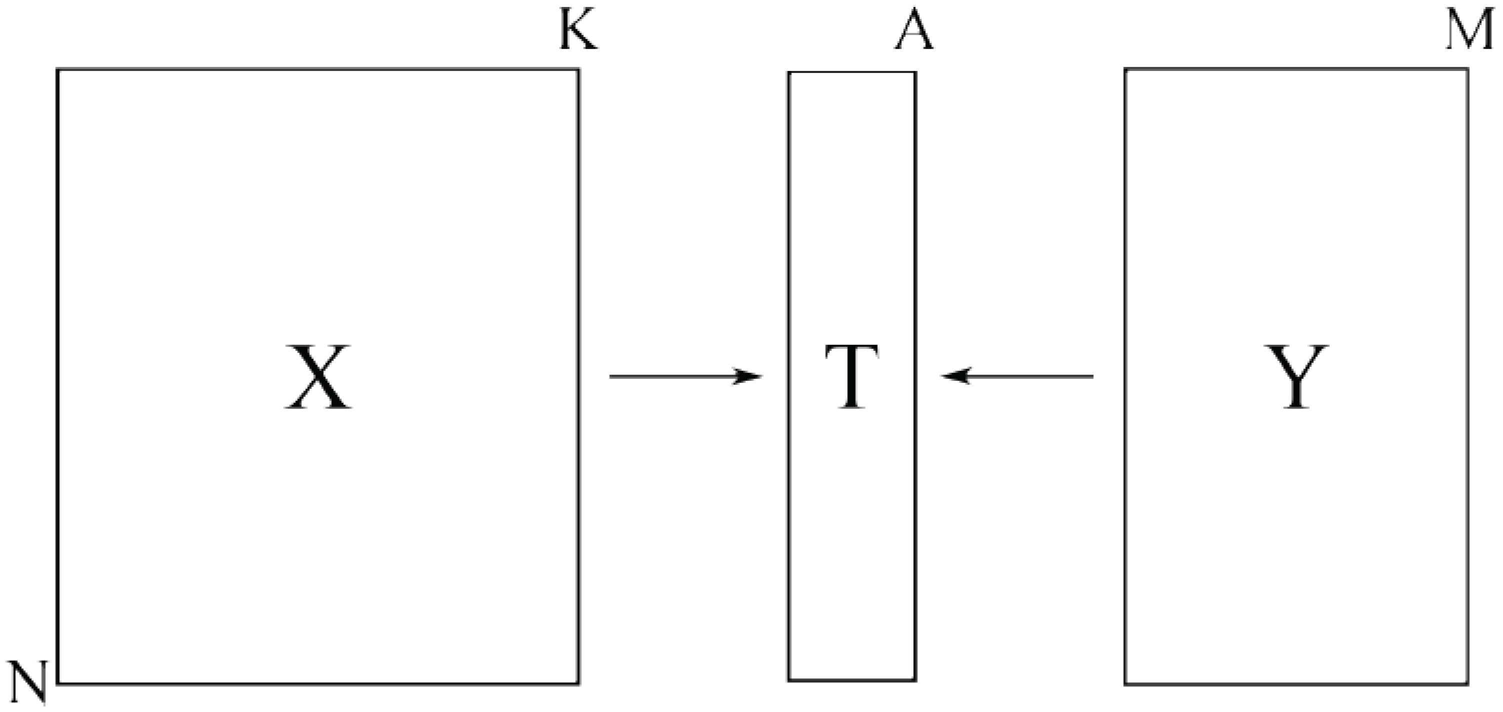
- 2 blocks of data
- Often used to predict **Y** given **X**
- Also used for monitoring, optimization, product development

Projection to Latent Structures (PLS)

- PLS
  - Generalization of PCA to deal with the relationship **X → Y**
- Advantages over PCR:
  - Has a model of Y space.
  - Can handle correlation in **Y**.
- Assumes there is error both in **X** and in **Y**

Projection to Latent Structures (PLS)

- Extracts each component sequentially
- Use cross-validation to check the number of components
- Scores calculated from **X** and **Y** *simultaneously*
- Makes engineering sense: system is driven (moved around) by the *same underlying latent variables*



Projection to Latent Structures (PLS)

- Objective function for PCA: best explanation of **X**-space
  - Optimization formulation of PCA
- Objective function for PLS: has 3 parts
  1. Best explanation of the **X**-space
  2. Best explanation of the **Y**-space
  3. Maximize relationship between **X**- and **Y**-space

PLS

Projection of X **both** approximates X well **and** correlates with Y (least squares fit)

PCA

Projection of X is an **optimal** approximation of X

Review of PCA formulation

- For PCA: best explanation of X-space:

$$\underset{\mathbf{p}_a}{\arg\max}\left(\mathbf{t}_a^T\mathbf{t}_a\right) \quad \text{s.t } \mathbf{p}_a^T\mathbf{p}_a = 1.0$$

  - gives greater variance of $\mathbf{t}_a$ (variance proportional to $\mathbf{t}_a^T\mathbf{t}_a$)
  - How do we get the scores?
    - $\mathbf{t}_a = \mathbf{X}_a\mathbf{p}_a$

Back to PLS

1. PLS scores explain **X**:
   - $\mathbf{t}_a = \mathbf{X}_a\mathbf{w}_a$ for the **X**-space
   - $\max\left(\mathbf{t}_a^T\mathbf{t}_a\right)$ subject to $\mathbf{w}_a^T\mathbf{w}_a = 1.0$
2. PLS scores also explain **Y**:
   - $\mathbf{u}_a = \mathbf{Y}_a\mathbf{c}_a$ for the **Y**-space
   - $\max\left(\mathbf{u}_a^T\mathbf{u}_a\right)$ subject to $\mathbf{c}_a^T\mathbf{c}_a = 1.0$
3. PLS maximizes relationship between **X**- and **Y**-space
   - How?

PLS: maximize relationship

- We have two scores: $\mathbf{t}_a$ and $\mathbf{u}_a$
  - t_a: summary of the X-space
  - u_a: summary of the Y-space
- The objective function of PLS:
  - Maximizes covariance: Cov ($\mathbf{t}_a$, $\mathbf{u}_a$)
  - This actually does three simultaneous things …

$$Cov(\mathbf{t}_a, \mathbf{u}_a) = \varepsilon\left\{(\mathbf{t}_a - \overline{\mathbf{t}}_a)(\mathbf{u}_a - \overline{\mathbf{u}}_a)\right\}$$

$$= \frac{1}{N}\mathbf{t}_a^T \mathbf{u}_a$$

PLS: maximize relationship

Correlation is easier to interpret: between -1 and $+1$

$$\text{Corr}(\mathbf{a}, \mathbf{b}) = \frac{\text{Cov}(\mathbf{a}, \mathbf{b})}{\sqrt{\text{Var}(\mathbf{a})} \cdot \sqrt{\text{Var}(\mathbf{b})}}$$

$$\text{Cov}(\mathbf{a}, \mathbf{b}) = \text{Corr}(\mathbf{a}, \mathbf{b}) \cdot \sqrt{\text{Var}(\mathbf{a})} \cdot \sqrt{\text{Var}(\mathbf{b})}$$

$$\text{Cov}(\mathbf{t}_a, \mathbf{u}_a) = \text{Corr}(\mathbf{t}_a, \mathbf{u}_a) \cdot \sqrt{\text{Var}(\mathbf{t}_a)} \cdot \sqrt{\text{Var}(\mathbf{u}_a)}$$

$$\text{Cov}(\mathbf{t}_a, \mathbf{u}_a) = \text{Corr}(\mathbf{t}_a, \mathbf{u}_a) \cdot \sqrt{\mathbf{t}_a^T \mathbf{t}_a} \cdot \sqrt{\mathbf{u}_a^T \mathbf{u}_a}$$

PLS: maximize relationship

- Maximizing covariance between $\mathbf{t}_a$ and $\mathbf{u}_a$ is actually:

$$\text{Cov}(\mathbf{t}_a, \mathbf{u}_a) = \text{Corr}(\mathbf{t}_a, \mathbf{u}_a) \cdot \sqrt{\mathbf{t}_a^T \mathbf{t}_a} \cdot \sqrt{\mathbf{u}_a^T \mathbf{u}_a}$$

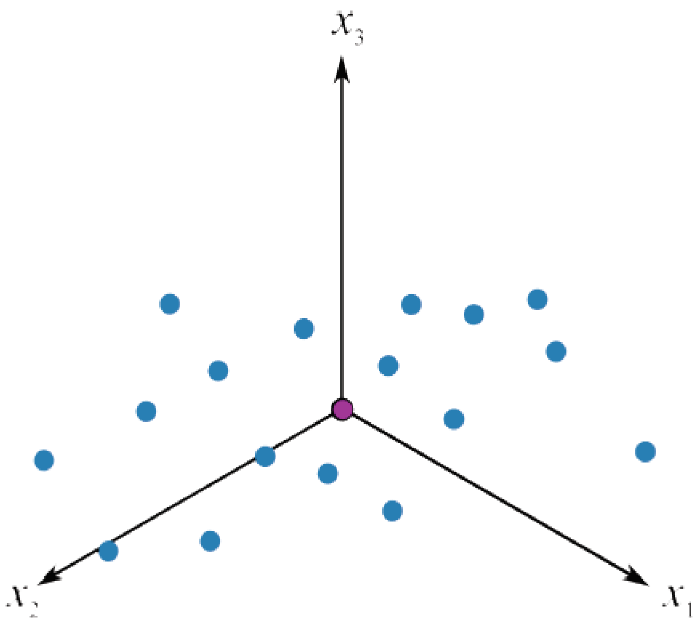
1. Explaining $\mathbf{X}$-space: given by $\mathbf{t}_a^T \mathbf{t}_a$
2. Explaining $\mathbf{Y}$-space. given by $\mathbf{u}_a^T \mathbf{u}_a$
3. Maximizing relationship between $\mathbf{X}$- and $\mathbf{Y}$-space: $\text{Corr}(\mathbf{t}_a, \mathbf{u}_a)$

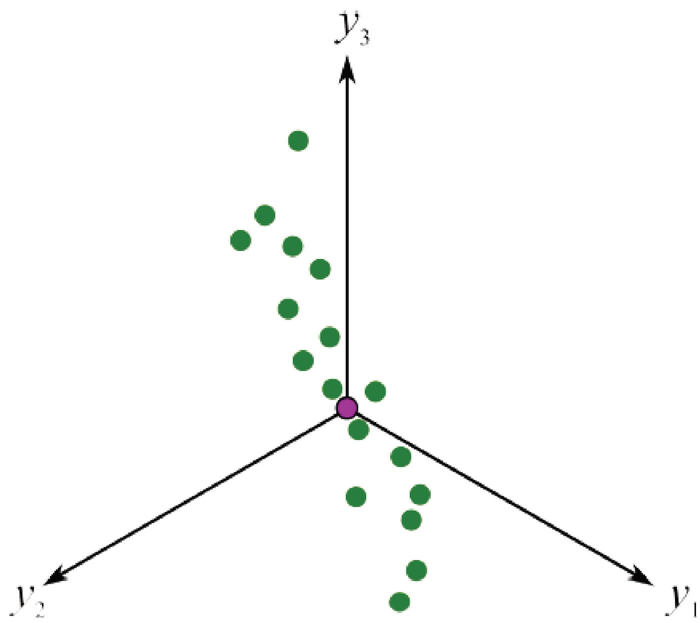
- Footnotes:
  - The above description is for SIMPLS (simple PLS)
  - The other variant of PLS is a little different (NIPALS)
  - SIMPLS = NIPALS when $M = 1$

PLS: geometric interpretation

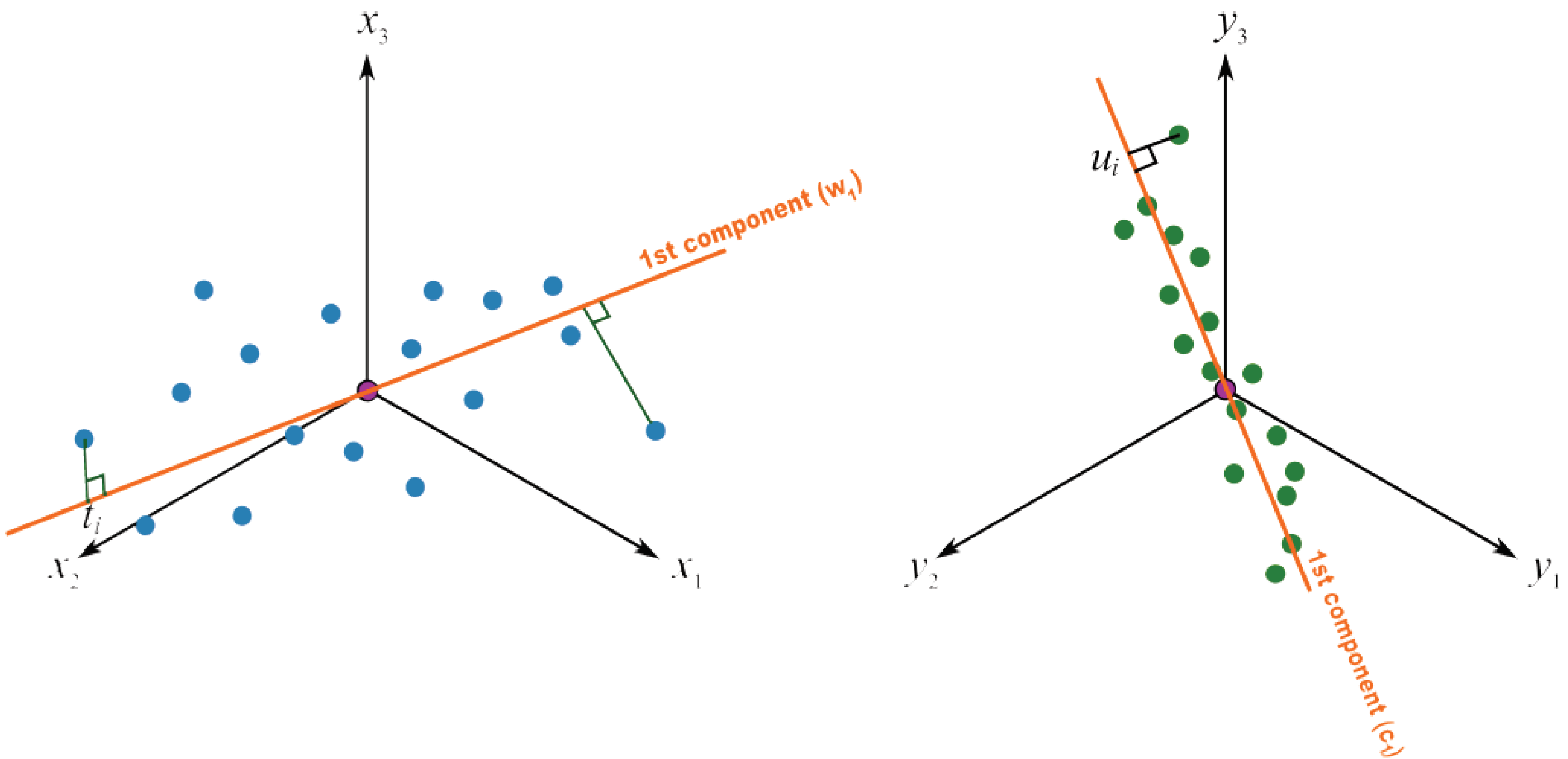
- For each matrix **X** and **Y**, we have K- and M-dimensional space.
- Each object is one point in the **X**- and **Y**- space.
- **X** and **Y** are two connected swarm of points in these two spaces.

- Mean-centering and scaling: same as in PCA.
- Calculate the average of each variable.
- These averages are subtracted from **X** and **Y**. And then, scaled to unit variance (usually)





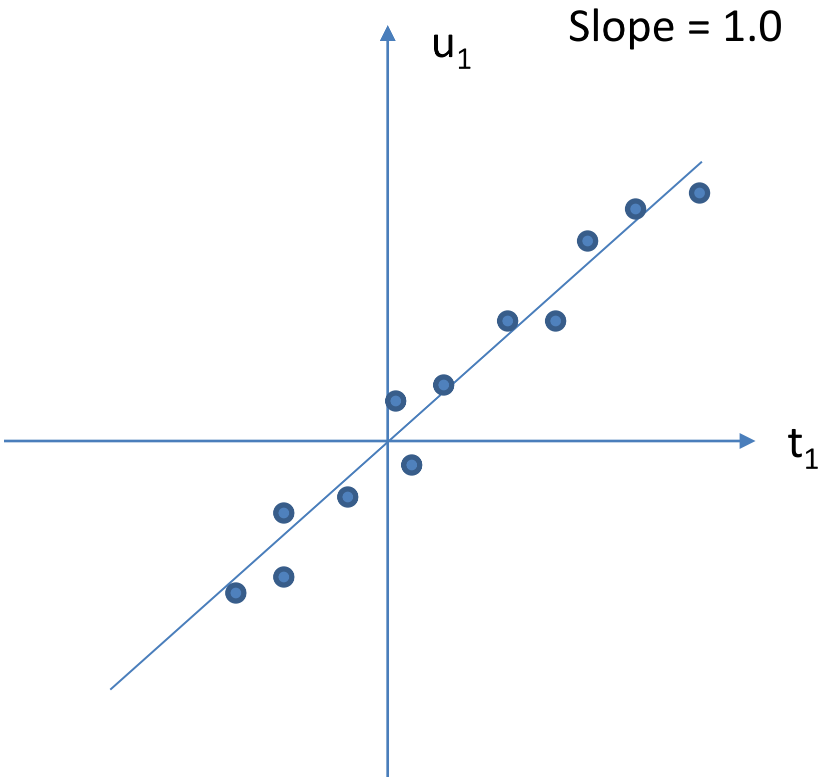
PLS: geometric interpretation



- 1st PLS component is a line in **X**- and **Y**- spaces, through the average points, such that

1. The lines well approximate the data
2. The projection ($\mathbf{t}_1$ and $\mathbf{u}_2$) are well correlated. (see next slide)

PLS: geometric interpretation

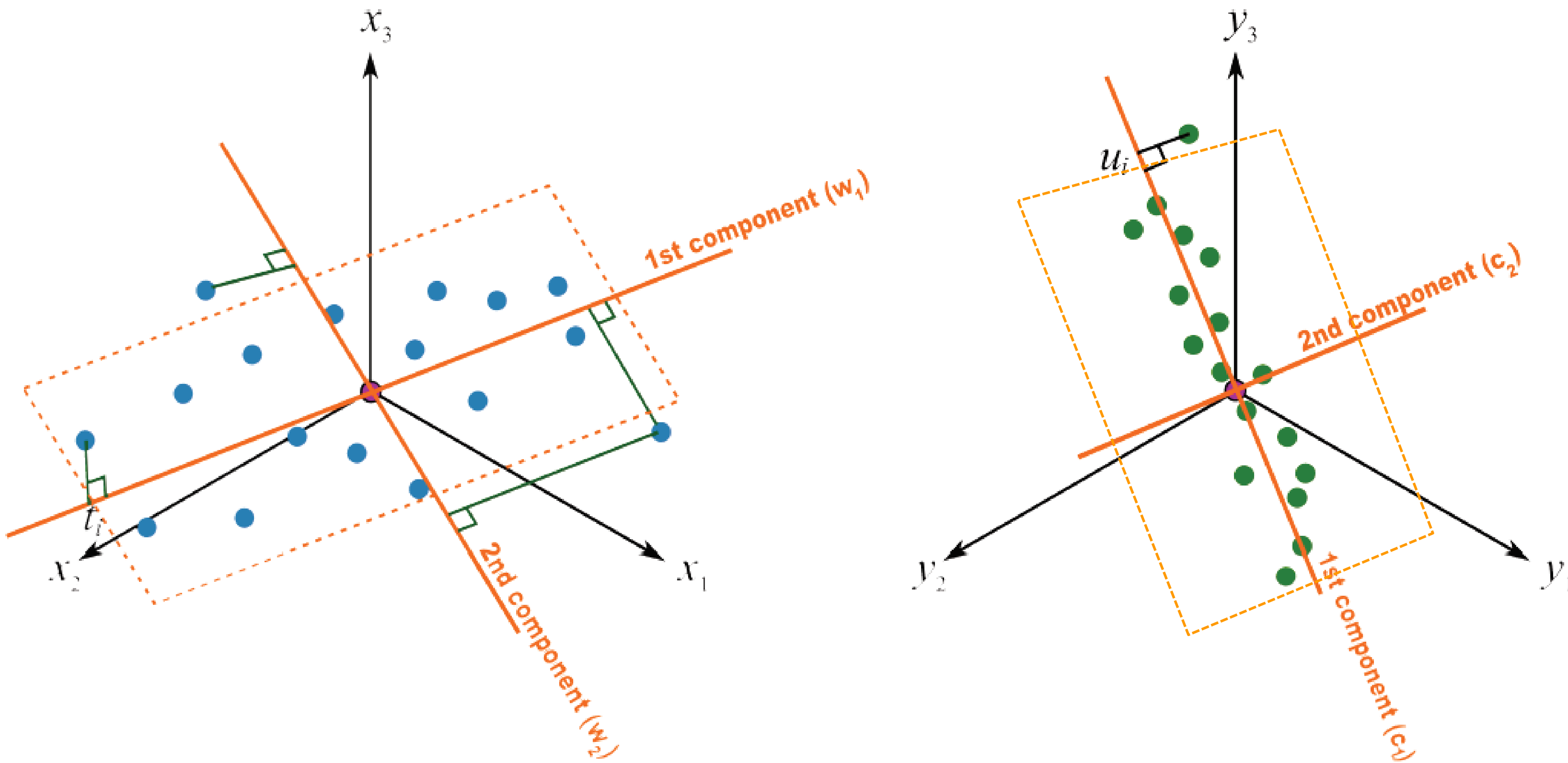


- The projected coordinates in the two spaces (**u_1** and **t_1** in **Y** and **X** are correlated in the inner relation

$u_{i1} = t_{i1} + h_i$

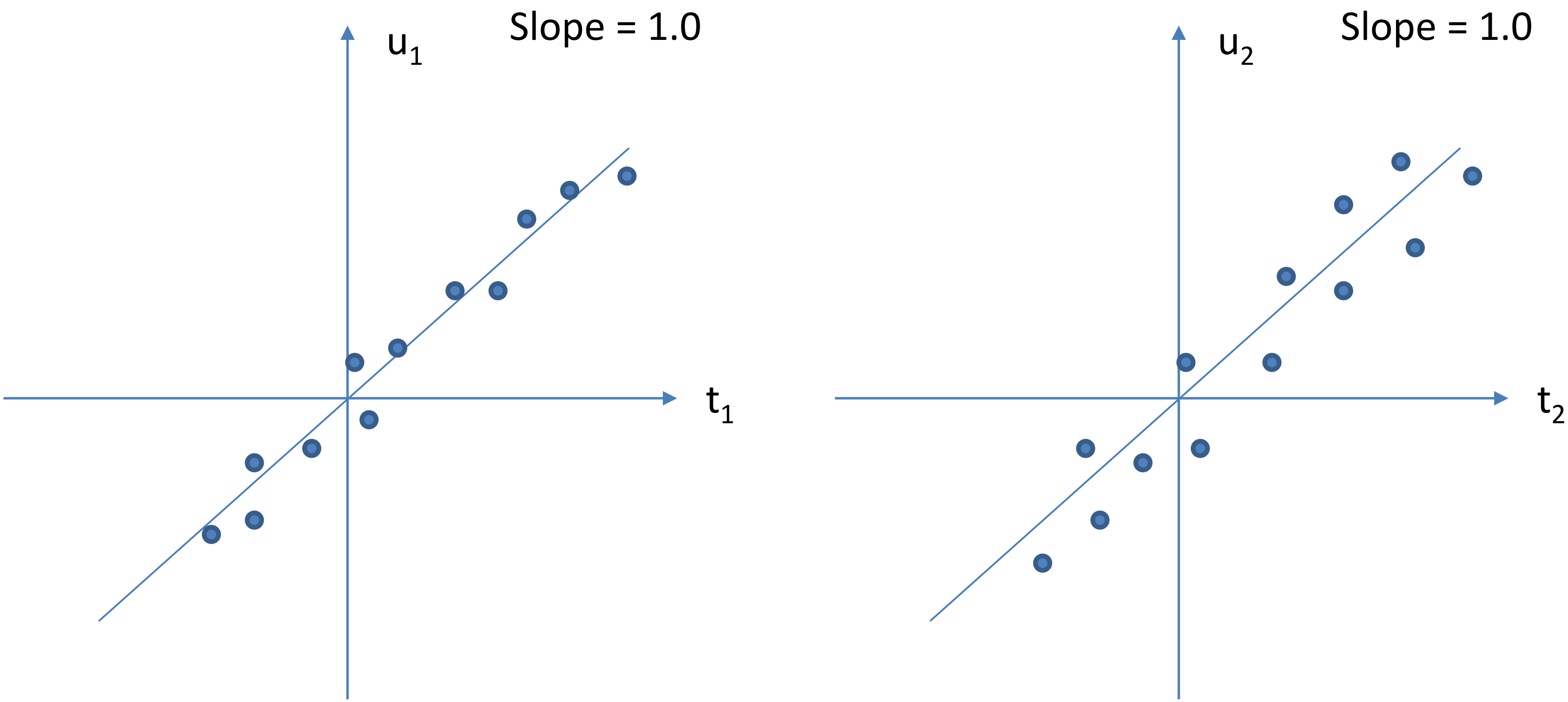
(h_i us a residual)

PLS: geometric interpretation



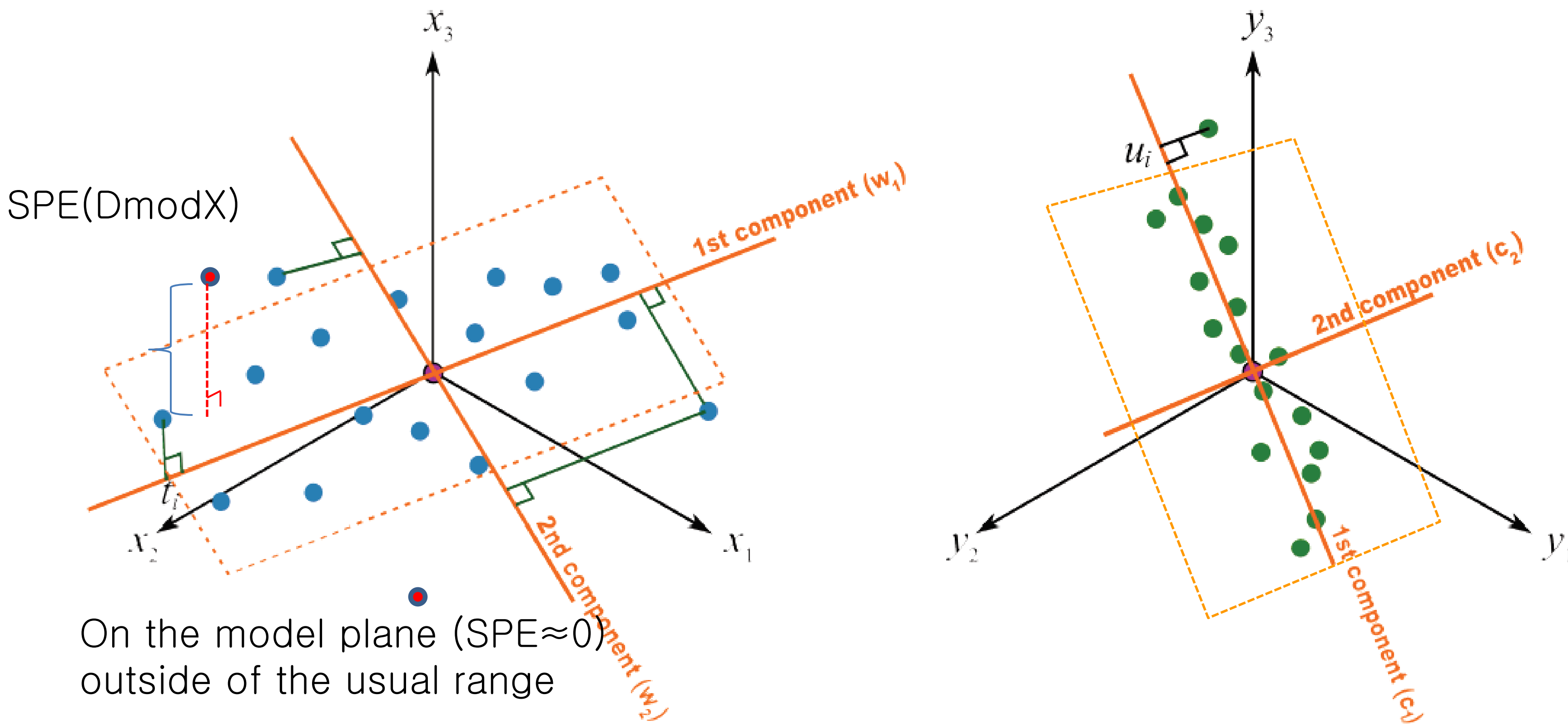
- The 2nd PLS component: lines in the **X**- and **Y**- spaces, through the average.
- The lines in **X**-space are orthogonal. Lines in the **Y**-space are **not** orthogonal.
- These lines improve the approximation and the correlation as much as possible.

PLS: geometric interpretation



- The 2nd projection coordinates (u_2 and t_2) are correlated, but usually less well than the first.

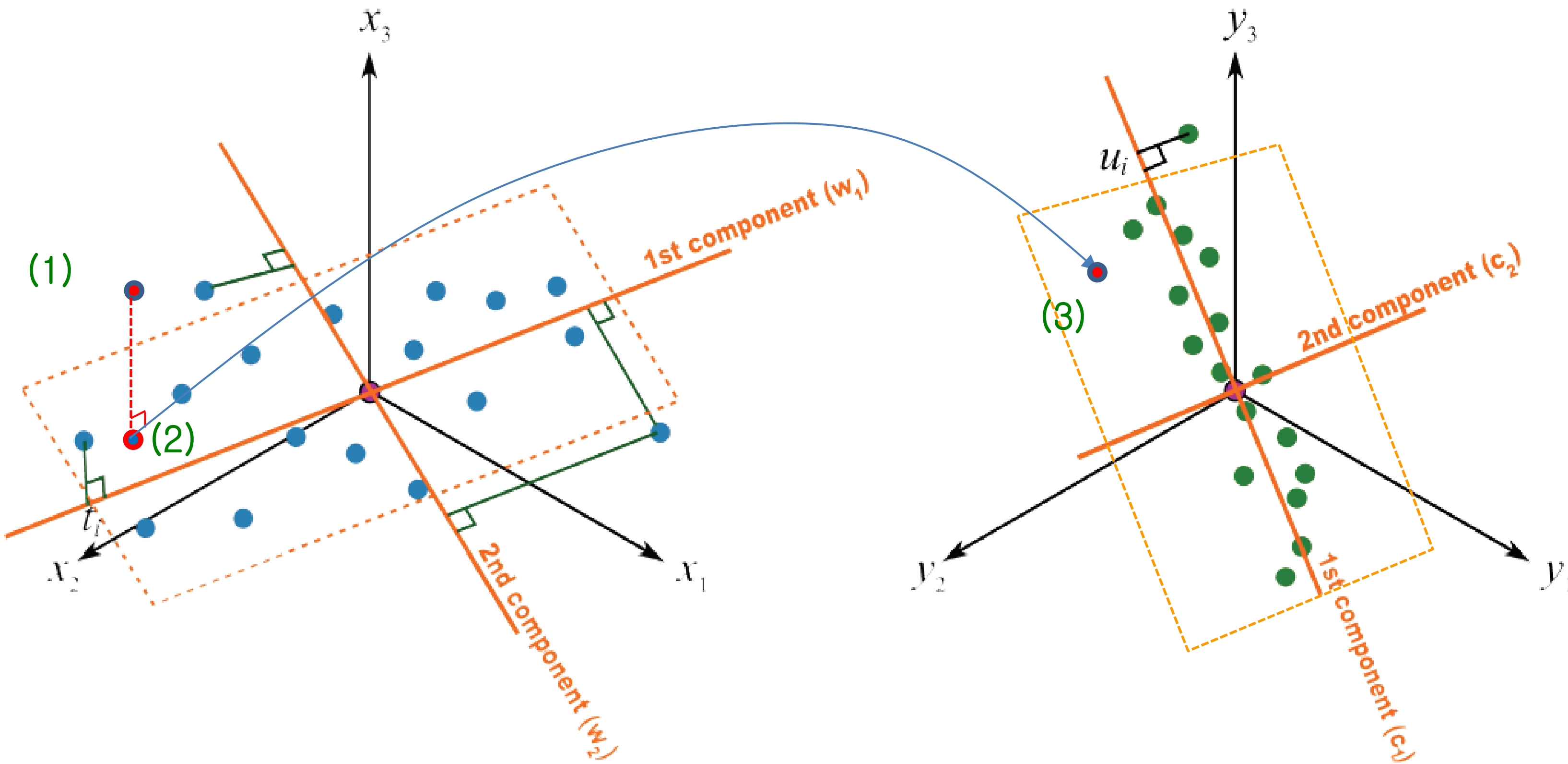
PLS: geometric interpretation



- The PLS components together form planes (or hyperplanes) in **X** and **Y**-space.
- The variability around the X-plane is used to calculate a tolerance interval within which new objects similar or the training set (calibration set) will be situated.

PLS: geometric interpretation

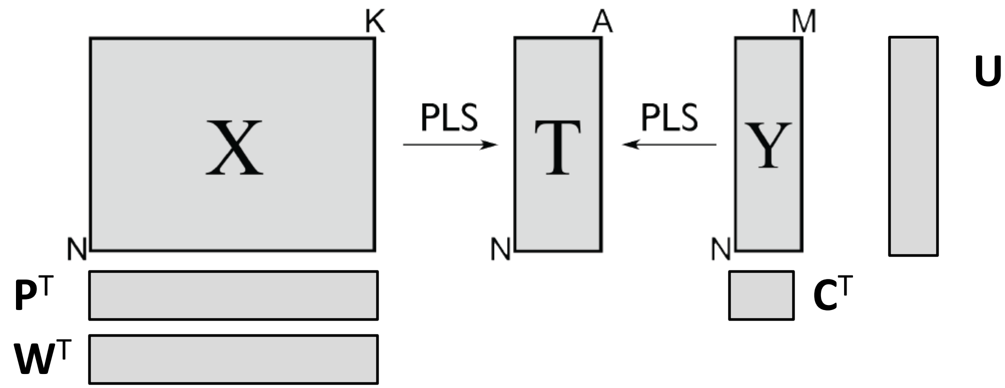
- For a new object,



•By inserting the **x**-values of a new object in X-space, we obtain its t_1 & t_2, which give predicted values of u_1 & u_2, which give predicted values of **Y**.

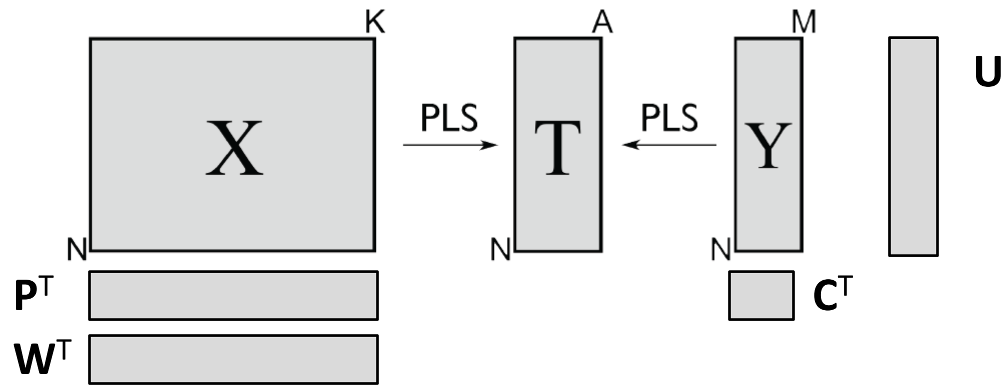
Projection to Latent Structures (PLS)

- Summary



  - *K*, *M*: number of X, Y variables
  - *N*: number of objects
  - *A*: number of PLS components
  - *k*(=1,2,…,*K*), *m*(=1,2,…,*M*): indices for X and Y variables
  - **T**, **U**: score matrices of X and Y
  - **P**: loading matrix
  - **W**: X-weight matrix
  - **C**: Y-weight matrix

Projection to Latent Structures (PLS)



- Summary
  1. Preprocessing
  2. PLS projection of data (X and Y) onto hyperplanes
  3. Scores, **t** and **u** are coordinates in the hyperplanes.
  4. Loadings **p** and weights **w** and **c** Define the direction of the hyperplane.
  5. PLS is also a regression model.

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{C}^T + \mathbf{F}$$

$$\mathbf{Y} = \mathbf{T}\mathbf{C}^T + \mathbf{F}'$$

$$\mathbf{U} = \mathbf{T} + \mathbf{H}$$

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E}$$

$$\mathbf{B} = \mathbf{W}\left(\mathbf{P}^T\mathbf{W}\right)^{-1}\mathbf{C}^T$$