Component based estimation of intangibles

A comparison of PLSPM with related methods

MIRI student: Jinyu Ji
Supervisor: Tomas Aluja
Master in Innovation and Research in Informatics
Specialization: Data Mining and Business Intelligence

Facultat d'informàtica de Barcelona
Index

Component based estimation of intangibles ................................................................. 1
A comparison of PLSPM with related methods ......................................................... 1
Index ................................................................................................................................. 2
Abstract .......................................................................................................................... 4
Chapter 1 Background Technology.............................................................................. 5
  1.1 Measuring and modeling the unobservable ......................................................... 5
  1.2 Intangibles ............................................................................................................... 5
  1.3 Reflective and Formative Measurement Models ................................................... 6
  1.4 From linear modeling to path modeling ............................................................... 9
    1.4.1 Path diagrams ................................................................................................. 9
    Notation and symbols ............................................................................................... 10
    Simple path diagrams ............................................................................................. 10
  1.4.2 Example of path model ................................................................................... 12
Chapter 2. Methodology of PLSPM ............................................................................ 15
  PLSPM-PLS path modeling ....................................................................................... 15
    2.1. The Nature of PLS Path Models ...................................................................... 15
    2.2. The PLS Path Modeling Algorithm ................................................................. 17
      2.2.1 Assessing unidimensionality ...................................................................... 19
    2.3 Latent variables estimation .............................................................................. 21
    2.4 PLSPM algorithm .............................................................................................. 23
      Stage1 .................................................................................................................... 24
      Stage 2: Path Coefficients ................................................................................. 26
      Stage 3: Loadings ................................................................................................. 26
      Wrapping up ......................................................................................................... 26
    2.5 The Quality Indexes .......................................................................................... 29
      2.5.1 Quality index for the outer model ............................................................. 30
      2.5.2 Quality index for the inner model ............................................................. 31
Chapter 3. Methodology of RGCCA .......................................................................... 32
  (Regularized Generalized Canonical Correlation Analysis) ....................................... 32
  3.1 RGCCA optimization ........................................................................................... 35
Chapter 4. Methodology of GSCA (Generalized Structured Component Analysis) .... 38
  4.1 The model ............................................................................................................ 38
  4.2 Estimation of the Parameter .............................................................................. 41
### 4.3 Quality index

Chapter 5. Overall Comparison of The methods

Chapter 6. Experiment with survey data

6.1 The inner model

6.2 Assessing unidimensionality

PCA of each block

6.3 Comparison in Mode A

The latent variables in Mode A

The path coefficients in Mode A

6.4 Comparison in Mode B

The latent variables in Mode B

The quality of the model

Path coefficients in Mode B

The latent variables in Mode B

Conclusion

Reference

Appendix 1

PLSPM

RGCCA

GSCA

Appendix 2

Original codes
Abstract

The using of PLSPM (partial least squares path modeling) Mode A to analysis real problem in Marketing, statistic research and education has good results. PLSPM methodology is the standard for component based estimation of intangibles, but recently other approaches have been proposed, mainly the Regularized Generalized Canonical Correlation Analysis and the Generalized Structure Component Analysis. Our purpose is to assess the differences and common basis for an empirical point of view, using an established benchmark of data.
Chapter 1 Background Technology

1.1 Measuring and modeling the unobservable

In our real life we can measure variety of things in our world. For the point of view of obtain knowledge from the measurements we need to analysis the possible relationships between the measurements, elaborate on descriptions of reality, and propose hypotheses and theories to be confirmed or discarded. To get the goal of analysis the measurements, we should build models to do the analysis. But sometimes the variables and concepts can’t directly measure nor be observed from the models. In these cases Michael Sobel (1994) refers to them as unobserved entities which in our research we called it latent variables. These entities are very common in social and behavioral sciences, such as image, satisfaction, value, and motivation.

1.2 Intangibles

Intangibles are also named as latent variables, sometimes short as LVs. In statistics, latent variables, are variables that are not directly observed but are
rather inferred (through a mathematical model) from other variables that are observed (directly measured). Mathematical models that aim to explain observed variables in terms of latent variables are called latent variable models. Latent variable models are used in many disciplines, including psychology, economics, medicine, physics, machine learning/artificial intelligence, bioinformatics, natural language processing, econometrics, management and the social sciences.

1.3 Reflective and Formative Measurement Models

Structural equation model typically involves multiple indicators of latent variables. Measuring the relational model or an external model specifies indicators and latent variables. Every time the relational model path measurement, therefore, the direction of causality between the latent variables and indicators by formative mode or reflective mode. Reflective measurement model has its classical test theory and psychometrics root (Nunnally & Bernstein, 1994). For each indicator represents errors latent variable torture measurements. Direction of causality is to build from the index; therefore, it is assumed to observe measures to reflect the changes in the underlying variables. In other words, changing the builder is expected to change its performance in all indicators. [Borsboom, D., Mellenbergh, G. J., & van Heerden, J. (2003). The theoretical status of latent variables. *Psychological Review, 110*(2), 203–219. doi:10.1037/0033-295X.110.2.203]
Thus, the latent variable is defined as a weighted score at all representative of the pointer variable, and the implementation itself is a separate variable for each dimension. An indicator of the increase in the value of the conversion value regardless of other indicators into a higher score is a composite variable. Formative measurement model depletion of the entire field of the index, which means that all of the relevant indicators collective dimensions or latent variables independent basis. One implication of this direction of causality is omitted unique econometric model of the formation of an indicator can be omitted, and change the meaning of the variables (Diamantopoulos & Winklhofer, 2001). In the reflective mode, the presence of latent construct measures (in absolute terms) Independent (Borsboom et al., 2004; Rossiter, 2002). Typical examples include reflective scenario attitude and personality of the measures initiated by the reaction of indicators to measure. Here is an example of showing the formative and reflective models:
Figure Example of latent variable measured by formative and reflective indicators

Figure drank clearly shows the formative and reflective model. The latent concept is to measuring "drank". The causes of getting drank could be have too much alcohol or just after drink which build the formative models. And the effects of getting drank could be can't walk in straight line, failed in alcohol blood test, can't speak fluently and so on build the reflective model.
1.4 From linear modeling to path modeling

1.4.1 Path diagrams

Path diagrams are very helpful because they provide a graphical representation of the relationships among a set of variables, with the special property that they can be translated into a system of simultaneous equations.

Notation and symbols

Path diagrams’ (Sewall Wright, 1920s) use in causal models and structural equation models have allowed for a general notation.

Variables can be manifest variables, latent variables, residual variables, or any kind of variables. Observed variables are enclosed in boxes; latent variables are enclosed in circle/ellipses; and residual terms are maintained unclose.

Relationships also can be of three types: causal links (assumed to be linear, and are represented by straight single-headed arrows) represent that variable A cause variable B; correlation links (represented by curved two-head arrows) indicates simply correlation between two variables A and B without implying causality; or the affection of a residual term $\epsilon$ to some variable A(by straight line).

In additional, variables may be grouped in two classed: exogenous, those that are not caused by any other variables in the diagram; endogenous, those that are caused by one or more variables. Exogenous latent variables are usually
represented by the Greek letter $\xi$, while the endogenous latent variables are represented by $\eta$.

**Simple path diagrams**

Relationship equation modeling mathematical model of the relationship between the types usually consists of a set of variables and a set of established (and explanatory variables dependent variable) between variables describes a system.
Multiple equations is the most realistic way, but we must take into account the fact that the dimension of the (variable) are not always fully determined. A simple linear regression model:

\[ X = \beta Y + \varepsilon \]

can be represented in path diagram as follows:

Variable X is the independent variable, which is assumed to explain variable Y, and an error term \( \varepsilon \) is associated to Y. The regression coefficient \( \beta \) is called the path coefficient.

A multiple relationship model can be represented as follows:

The two exogenous latent variables \( \xi_1 \) and \( \xi_2 \) may be correlated as a coved arrow connect them to each other. The \( \eta_1 \) is the endogenous caused by the exogenous LVs, and an error term \( \varepsilon \) is associated with it. The path coefficients are indicated by \( \beta_1 \) and \( \beta_2 \).

Typically in causal modeling the path diagram contains some structural relations among constructs, each one related with its indicators. That means mostly the
model will be a combination of formative and reflective models. For example, a model with three LVs $\xi_1$ and $\xi_2$ causing $\eta$, each one is associated with a block of two indicators. The LV $\xi_1$ and $\xi_2$ are associated in a formative way and no residual terms are considered. But the LV $\eta_1$ is related in a reflective way, so each indicator $y_j$ has its corresponding disturbance term $\varepsilon_j$.

1.4.2 Example of path model

The following model is a work of Tenenhaus [1998]. We want to take the example to illustrate what is path model. The example is taken from a paper by Russet [1964]. “The basic hypothesis in Russet’s paper is that economic inequality leads to political instability.” Have been used to measure the uneven distribution of land three indicators, they are "Gini", "farm" and "Rent." The indicator "Gini", is to focus on the Gini index is a measure of the deviation from the line of equality the Lorenz curve. The indicator "farm" is the land of farmer's own half, beginning with the smallest percentage. Therefore, if the "farm" is 90%, then 10% of the farmers own half the land; the third indicator is the "rent",
which is a percentage of all households rent their land. "Gnpr" and "Rab" is a
measure of industrial development: Indicators "gnpr" in 1955 GDP pro capite (in
dollar terms), and indicates "LABO" is the percentage of the labor force engaged
in agriculture. Four indicators measures political stability. The indicator "Inst" is
the number of administrative and national independence number of princes of
the 1946-1961 function during this period. Indicator "ecks" is Eckstein index
calculated for the period 1946-1961. The indicator "death" is that people in this
period from 1950 to 1962 the number killed by the results of the performance
indicators of violence "demo" classification of three groups of countries: a stable
democracy, unstable democracy and dictatorship. He divided the three reflective
tawny data block set. So the model is built by there latent variables (blocks), they
are “Agricultural Inequality”, “Industrial Development” and “Political Instability”.
The first block, $\xi_1$, consisting of the indicators "gini", "farm" and "rent" measures
the composite indicator “Agricultural Inequality”. The second one, $\xi_2$, formed by
the indicators “gnpr” and “labo”, measures the composite indicator "Industrial
Development". The third block, composed of the indicators “inst", "ecks", “death"
and “demo”, expresses the composite indicator “Political Instability”.
All the LVs are measured with reflective indicators in this model. The $\beta$ means the path coefficient between latent variables. And $\lambda$ means the weight of the latent variable to the corresponding manifest variables.
Chapter 2. Methodology of PLSPM

PLSPM-PLS path modeling

Extending PCA (principal component) and canonical correlation analysis, PLS is a part of alternating least squares algorithms. The method was designed by Wold (1974, 1982, 1985) for the analysis of high dimensional data in a low-structure environment, and experiences various extensions and modifications. [Lohmöller (1989)].

2.1. The Nature of PLS Path Models

Outer model and inner model: PLS path model is built with these two sets of linear constrains. Internal model to describe the relationship between the latent variables and the external model describes the relationship between the latent variable and its manifest variables. [Vinzi, Vincenzo Esposito, Laura Trinchera, and Silvano Amato. "PLS Path Modeling: From Foundations to Recent Developments and Open Issues for Model Assessment and Improvement." *Handbook of Partial Least Squares* (2009): 47-82. Web.]
The manifest variables can be centered or standardized and latent variables should be standardized in the outer and inner model in PLS. We can describe the inner model like this which shows the relationship between LVs:

\[ \xi = B\xi + \zeta \]  

(1)

where \( \xi \) is the latent variables, \( B \) is the matrix of inner model path coefficients, and \( \zeta \) represents the inner model residuals. We define the inner model \( \xi_{endogenous} \) is caused by \( \xi_{exogenous} \). The causal chain should be within the model system. Predictor specification reduces Eq. (1) to:

\[ (\xi_{endogenous}|\xi_{exogenous}) = B\xi_{exogenous} \]  

(2)

The latent variable is linked to its associated manifest variables with causality relation in its block. The manifest variables build a linear function to represent the latent variable and the residual \( \epsilon \) is introduced to represent the error:
\[ X_x = \Lambda_x \xi + \varepsilon \]  

(3)

where \( \Lambda \) represents the loading coefficients. There are no correlations between the outer residuals and the latent variable of the same block – that reduces Eq. (3) to:

\[ (X_x | \xi) = \Lambda_x \xi \]  

(4)

Measurement of formative model has variable causality latent variables from the associated manifest variables. For these blocks, linear relationship are given as following:

\[ \xi = \Pi_x X_x + \varepsilon_x \]  

(5)

Predictor specification is also in effect in this mode, making Eq. (5) to:

\[ (\xi | X_x) = \Pi_x X_x \]  

(6)

Basic PLS algorithm will be described in the next section. We will work with the data matrix and latent variable manifest variables and the score is calculated for all the unknown relationship.

2.2. The PLS Path Modeling Algorithm

The goal of PLS Path Modeling is to estimate the relationships among \( Q \) (\( q = 1, ..., Q \)) blocks of variables, which stand for latent concepts. PLS-PM is based on
simple and multiple regression equation interdependent system. Such a system is estimated network relationship between latent variables and also the relationship between the latent variable and their manifest variables. Formally, we assume P variables (p = 1, ..., P) observed on N units (n = 1, ..., N). From a partitioned data table X, the resulting data \( (x_{npq}) \) are collected:

\[
X = [X_1, \ldots, X_q, \ldots, X_Q]
\]

where \( X_q \) is the generic q-th block made of \( P_q \) variables. [Vinzi, Vincenzo Esposito, Laura Trinchera, and Silvano Amato, 2009]

As well known, two sub-models compose each Structural Equation Model: the measurement model and the structural model. The first consideration is the relationship between the various latent variables and the corresponding list of variables, while taking into account the structural model of the relationship between the latent variables. In the PLS Path Modeling framework, the structural model can be written as:

\[
\xi_j = \beta_{0j} + \sum_{q \neq q_j} \beta_{qj} \xi_q + \zeta_j
\]

where \( \xi_j (j = 1, \ldots, j) \) is the generic endogenous latent variable, \( \beta_{qj} \) is the generic path coefficient interrelating the q-th exogenous latent variable to the j-th endogenous one, and \( \zeta_j \) is the error in the inner relation.

Measurement model formulation is dependent upon the direction of causality relationship between latent variables and the corresponding manifest variables (Fornell and Bookstein 1982). As a fact, different types of measurement models to choose from: reflective model, formative model and a mix of the two models.

In our research we only focus on the reflective model and the formative model.
2.2.1 Assessing unidimensionality

We perform principal component analysis to assess unidimensionality. We need to perform principal component analysis for each block: if the first eigenvalue of its correlation matrix is higher than 1, while the others are smaller, a block may be considered unidimensional (Kaiser’s rule). In the formative way, the model represents a different concept for each sub-block of the basic concepts of each variable in the variable list or manifest. What else, in the reflective mode, the model assumes a uniform one-dimensional block. Always, latent variable is defined as a linear combination of the corresponding list of variables, so that each list variable is a measure of exogenous variables in the model. Therefore, the measurement model can be expressed as:

$$\xi_q = \sum_{p=1}^{pq} \omega_{pq}x_{pq} + \delta_q$$

where $\omega_{pq}$ is the coefficient of manifest variable linking to its corresponding latent variable and the fraction of the corresponding latent variable not accounted for by the block of manifest variables is represented by the error term $\delta_q$. The assumption behind this model is the following predictor specification:
\[ E(\xi_q | x_{pq}) = \sum_{p=1}^{P_q} \omega_{pq} x_{pq} \]

When all manifest variables are in the same scale observation and measurement of all external weights is positive, it is interesting and feasible to express these scores in the original scale (Fornel 1992). This is accomplished by using the normalized weight \( \hat{w}_{pq} \) defined as implemented:

\[ \hat{w}_{pq} = \frac{w_{pq}}{\sum_{p=1}^{P_q} w_{pq}} \text{ with } \sum_{p=1}^{P_q} \hat{w}_{pq} = 1 \forall q : P_q > 1 \]

The weight relation only means that the PLS path modeling, any latent variable is defined as a weighted sum of its own performance variables. It does not affect the direction of the latent variables and the relationship between the outer layers of the model's manifest variables. [Vinzi, Vincenzo Esposito, Laura Trinchera, and Silvano Amato. "PLS Path Modeling: From Foundations to Recent Developments and Open Issues for Model Assessment and Improvement." Handbook of Partial Least Squares (2009): 47-82. Web.]

An iterative procedure allows the estimation of external weights (\( w_{pq} \)) and the latent variable scores (\( \hat{\xi}_q \)) in PLS Path Modeling. Estimates are partial least squares analysis. Because it by alternating single and multiple linear regression method, one block at a time to solve the estimation, the process is named partial. The path coefficients (\( \beta_{qj} \)) are estimated afterwards by means of a regular regression between the estimated latent variable scores in accordance with the specified network of structural relations. We prefer to think of such a network is...
defined as an endogenous latent variable, rather than a causal model to predict the path of the network when taking into account the regression framework of PLS Path Modeling.

2.3 Latent variables estimation

The weights estimation step through the outer and inner alternate, iterate until convergence to achieve. It stressed that the convergence of the algorithm is no formal evidence has been provided to the current models have two or more blocks is very important. However, experience in practice, convergence is usually observed.

The process begins by selecting an arbitrary initial weight initial weight $w_{pq}$ centered (or standardized) applies to the centered (or standardized) manifest variables. Then, the external phase estimation, for each latent variable is estimated as a linear combination of its own manifest variables:

$$v_q \mathcal{Z} \pm \sum_{p=1}^{P_q} w_{pq} x_{pq} = \pm X_q w_q$$

where $v_q$ is the standardized (zero mean and unitary standard deviation) outer estimate of the $q$-th latent variable $q$, the symbol $\mathcal{Z}$ indicates the left side of the equation corresponding to the right side of the standardization and the “±” sign shows the sign ambiguity. Choosing the sign making the outer estimate
positively correlated to a majority of its manifest variables usually solves this ambiguity.

Each latent variable is estimated by considering its links with the other $Q'$ adjacent latent variables in the inner estimation stage:

$$v_q \sim \sum_{q'=1}^{Q'} e_{qq'} v_{q'}$$

The $e_{qq'}$ represents the inner weights. The algorithm will update the outer weights $w_{pq}$ after getting the estimate of the latent variables. [Vinzi, Vincenzo Esposito, Laura Trinchera, and Silvano Amato, 2009]

Two different modes can update outer weights. They are closely related, but not the same with each other, the formative and the reflective modes:

Mode A: each outer weight $w_{pq}$ is updated as the regression coefficient in the simple regression of the $p$-th manifest variable of the $q$-th block ($x_{pq}$) on the inner estimate of the $q$-th latent variable $v_q$. As a matter of fact, since $v_q$ is standardized, the generic outer weight $w_{pq}$ is obtained as:

$$w_{pq} = cov(x_{pq}, v_q)$$

i.e. Regression coefficient is reduced to the corresponding list of each variable and latent variables covariance between estimates. In case the manifest variables has also standardized, covariance become such a correlation.

In Mode B: Updating the vector $w_q$ of the weights $w_{pq}$ to be the vector of the regression coefficients in the multiple regression of the inner estimate of the $q$-th latent variable $v_q$ on the manifest variables in $X_q$:
\[ w_q = (X_q 'X_q)^{-1}X_q '\theta_q \]

where \( X_q \) comprises the \( P_q \) manifest variables \( x_{pq} \) previously centred and scaled by \( \sqrt{1/N} \).

As already mentioned, the choice of selecting Mode A or Mode B depends on the measurement model. The Mode A is more appropriate for a reflective model, while Mode B is better for a formative model. Furthermore, Mode A is suggested for endogenous latent variables, while Mode B for the exogenous ones.

And in our research we will compare separately with mode A and mode B with different algorithms.

### 2.4 PLSPM algorithm

The PLS algorithm is essentially a sequence of regressions in terms of weight vectors. The basic PLS algorithm by Lohmoller (1989) suggests, containing of three stages [Vinzi, Vincenzo Esposito, Laura Trinchera, and Silvano Amato. "PLS Path Modeling: From Foundations to Recent Developments and Open Issues for
Stage 1: latent variable scores, including four-step iterative process is repeated until the convergence of the iterative estimation obtained:

(1) Outer approximation of the latent variable scores,
(2) Estimation of inner weights,
(3) Inner approximation of the latent variable scores,
(4) Estimation of outer weights.

Stage 2: External weights / loadings and the estimated path coefficients.
Stage 3: Location parameter estimation.

**Stage 1**

Step 1: latent variable score outer approximation.

The latent variables outer proxies, $\xi_\text{outer}^n$ are calculated as linear combinations of the respective indicators. These are standardized external proxies. Weight right from the results of a linear combination of the previous iteration step 4. When the algorithm is initialized, and no weights are available, the index of any nontrivial linear combination can be used as external proxies of latent variables.

Step 2: Estimation of inner weight.

Within the weights to reflect the other latent variables are strongly connected to how it is calculated for each potential variable. It can be used to determine the presence of three options within the weights. Wold (1982) originally proposed
the centroid scheme. There are three methods to estimate the weight, centroid, factor-weighting, and path-weighting scheme. The centroid scheme utilizes the sign of the correlation between a latent variable and its adjacent latent variables; the weighting factor scheme utilizes correlations. The path-weighting regime honors the guidance arrows in the path model. The weight of these latent variables that explain the focal latent variable are set to the regression coefficients resulting from a regression of the latent variable focal length (regressing) on its latent variables repressors. The weight of these latent variables, which are explained by the focal latent variable, are determined in the same manner as in the weighting factor scheme. Whatever weighting scheme, a zero weight is given to all non-adjacent latent variables. [Vinzi, Vincenzo Esposito, Laura Trinchera, and Silvano Amato, 2009]

Step 3: Inner approximation of the latent variable scores.

By using the calculated inner weights, $\xi_{n}^{inner}$, are calculated as linear combinations of the external proxies of their respective adjacent latent variables.

Step 4: Estimation of the outer weights.

Calculation of external weight either as each latent variable and its indicators in the proxy covariance between the return of the right (A mode, reflective) or ordinary least squares regression of the interior of each latent variable weight got its proxies (in mode B, formative).

Repeating the four steps until two iterations between the outer weight changes
is less than the predetermined limit, normally 0.0001.

At the end of the algorithm, stage 1 delivers the latent variable scores for all potential variables. Load and then calculate regression coefficients in a simple manner.

**Stage 2: Path Coefficients**

Calculating the path coefficient is the second stage of the algorithm. The coefficients are estimated by ordinary least squares in the multiple regressions of the related latent variables specified in the inner model.

**Stage 3: Loadings**

The third stage of the algorithm consists of calculating the loadings. Loadings are always calculated as correlations between a latent variable and its indicators as convenient. [Attribution-noncommercial-sharealike, C. C., License, U., By-nc-sa, C. C., & Sanchez, G. (n.d.). PLS Path Modeling with R G aston S anchez.]

**Wrapping up**

The algorithm begins with initial weights used to calculate the latent variable outside of any of the approximate weight. The initial weights are used to make the MVs in a linear combination according to their related latent variable. Then, the relationship between the LV is considered to calculate the internal approximations. There are three methods to calculate the inner weight $e_{qq}$, to perform the approximate selection: (1) a centroid, (2) factor, and (3) the path
scheme. Once the approximation obtained, the algorithm turns to the outside when the relationship between the new weights and indicators are calculated considering how to construct them: by mode A (reflective), or by the way B (formative). A simple linear regression model means that, while the B-mode means that multiple linear regression. Simple and / or regression coefficients are then used as the outer approximation new weight. The iterative process continues until the weight reaches convergence. After the outer weight converges, once the latent variable estimation, the model parameters and measuring structure can be obtained. The path coefficients are calculated by ordinary least squares regressions between latent variables. A lot of regressions as endogenous latent variables need to be taking into account. The loading coefficients are also estimated by least squares regressions. But the choice of which kind of mode to be used (reflective or formative) should be considered.

[ Vinzi, Vincenzo Esposito, Laura Trinchera, and Silvano Amato, 2009]

Algorithm 1: PLS Path Modeling based on L’ohmoller’s algorithm with the following options: centroid scheme, standardized latent variable scores, OLS regressions

\[
\text{Input: } X = [X_1, \ldots, X_q, \ldots, X_Q ] \text{ i.e. Q blocks of centred manifest variables;} \\
\text{Output: } w_q, \hat{\xi}_q, \beta_j;
\]

1: for all \( q = 1, \ldots, Q \) do
2: initialize $w_q$

3: $v_q \mathcal{I} \pm \sum_{p=1}^{p_q} w_{pq} x_{pq} = \pm X_q W_q$

4: $e_{qq'} = \text{sign} [\text{cor}(v_q, v_{q'})]$ following the centroid scheme

5: $\mathcal{E}_{q} \mathcal{E} \sum_{q'=1}^{q'} e_{qq'} v_{q'}$

6: update $w_q$:
   (a) $w_q = \text{cor}(x_{pq}, \mathcal{E}_{q})$ for Mode A (outwards directed model)
   (b) $w_q = (\mathcal{E}_{q} x_{q})^{-1}(\mathcal{E}_{q} \mathcal{E}_{q})$ for Mode B (inwards directed model)

7: end for

8: Steps 1–7 are repeated until convergence on the outer weights is achieved, i.e. until:

$$\max \{ w_{pq, \text{current iteration}} - w_{pq, \text{previous iteration}} < \Delta \}$$

where $\Delta$ is a convergence tolerance usually set at 0.0001 or less

9: Upon convergence:
   (1) for each block the standardized latent variable scores are computed as weighted aggregates of manifest variables:

$$\xi_q \propto X_q \mathcal{E}_q$$

(2) for each endogenous latent variable $\xi_j(1, \ldots, j, \ldots, J)$, the vector of path coefficients is estimated by means of OLS regression as:

$$\beta_j = (\mathcal{E}' \mathcal{E})^{-1} \mathcal{E}' \xi_j,$$
where $\zeta$ includes the scores of the latent variables that explain the $j$-th endogenous latent variable $\xi_j$, and $\xi_j$ is the latent variable score of the $j$-th endogenous latent variable.


2.5 The Quality Indexes

PLS Path Modeling doesn’t have global optimization criterion so that there is no global fitting function to assess the goodness of the model. In addition, it is based on the strongly oriented to the prediction variance model. Therefore, model validation focuses primarily on the predictive ability of the model. According to the structure of PLS-PM, every part of the model needs to be validated: the measurement model, the structural model and the general. Therefore, PLS Path Modeling offers these different adjustment indexes: the index of communality, the rate of redundancy and index goodness of fit (GOF). But the GoF is not well provides reasonable result not accepted for all academic community. So we only considering the first two index. And also we consider $R^2$ measurements to evaluate the model.
2.5.1 Quality index for the outer model

For each $q$-th latent variable with more than 1 manifest variable (i.e. for each block with $P_q > 1$) we calculate the means of the communality index to measure the quality of the measurement model:

$$Com_q = \frac{1}{P_q} \sum_{p=1}^{P_q} cor^2(x_{pq}, \xi_q) \forall q: P_q > 1.$$  

The index is a measure of how much variability in the manifest variables in the $q$-th blocks by their own latent variable score $\xi_q$. In addition, the communality index for the $q$-th block is the average of the squared correlations between each manifest variable in the $q$-th block and the corresponding latent variable scores. The average communality index of the whole model could be calculated by the means of the total quality, i.e:

$$Com = \frac{1}{\sum_{q:P_q>1} P_q} \sum_{q:P_q>1} P_q Com_q$$

This is a weighted average of all the blocks communality indexes with weights equal to the number of manifest variables in each block. Moreover, since the communality index for the $q$-th block is the average of the squared correlation in the block, then the average communality is the average of all the squared correlations between each manifest variable and the corresponding latent variable scores in the model, i.e.:
\[
Com = \frac{1}{\sum_{q:p_q>1} p_q \sum_{p:p_q>1}^{p_q} \sum_{p=1}^{p_q} \text{cor}^2(x_{pq}, \xi_q)}.
\]

### 2.5.2 Quality index for the inner model

The redundancy calculated for the \( j \)-th endogenous block, measures the portion of variability of the manifest variables connected to the \( j \)-th endogenous latent variable explained by the latent variables directly connected to the block, i.e.:

\[
Red_j = Com_j \times R^2(\xi_j, \xi_q; \xi_q \rightarrow \xi_j)
\]

average R2 value is obtained as:

\[
\overline{R^2} = \frac{1}{J} R^2(\xi_j, \xi_q; \xi_q \rightarrow \xi_j)
\]
Chapter 3. Methodology of RGCCA

(Regularized Generalized Canonical Correlation Analysis)

The PLSPM involving the same type of data, and share the same goal: How few blocks in the same group of people will be observed variables. The ability of several data analysis blocks in that it includes a wide variety of methods have well-defined standards that optimized. The highway PLS modeling flexibility that allows for the possibility of a link between blocks certain assumptions: researchers decide which block connections, which are not. Unfortunately, the options for optimizing the standard by the PLS path modeling algorithm is often blurred. Tenenhaus proposed a new method called Generalized regularized canonical correlation analysis (RGCCA). This is a generalization of canonical correlation analysis corrected (Vinod 1976; Leurgans, Moyeed & Silverman, 1993) for three or more sets of variables. RGCCA has the power of multi-block data analysis methods and adopts the flexibility of PLS modeling. [Tenenhaus, A., & Tenenhaus, M. (2011)]

RGCCA framework is used to analysis several blocks in a linear relationship between manifest variables observed in the same group. Consider a network connection between these blocks, the goal of RGCCA is to find a block variable
(block element) linear combination allows (i) block components to explain their own block the well and / or (ii) assuming that 'block assembly is strongly correlated connected.

Unlike the PLS method, RGCCA result is the correlation between latent variables and the correlation between manifest variables and their associated latent variables. The method proposed by Tenenhaus allowing the use of the algorithm is very similar to PLSPM algorithm while it provide a global optimize function. The RGCCA is based on a simple iterative algorithm similar to PLS method. Once the algorithm has converged, the optimization we get depends on tau parameter specific function selection parameter results. Tau is a parameter for each potential variable to be set. It enables you to adjust the "Mode" potentially relevant variables. If the Tau = 0, then we will in the case of Mode B, and the results PLSPM and RGCCA are similar. When Tau = 1, we turned to new Mode A (illustrated by M. Tenenhaus a) in the new model. This model is close to PLSPM mode A, while given an optimizing function. When Tau varies between 0 and 1, latent variable models perform between Mode A and Mode B. [Tenenhaus, A., & Tenenhaus, M. (2011)]

In the RGCCA algorithm, we consider the structure of the data matrix X into group (rows partition) or block (column-wise). The X row and column variables associated individuals. Multi-block data analysis analyzes several sets of variables (block) involved, the same group of people was observed. Analyze multiple sets of data relates to a group analysis of the individual groups of observed variables. Note that the term used in the literature on 'multiple-block 'and' 'multi-group' does not build consensus. In the framework of multiblock, a
column partition $X = [X_1, \ldots, X_j, \ldots, X_J]$ is considered. In this case, each $n \times p_j$ data matrix $X_j$ is called a block and represents a set of $p_j$ variables observed on $n$ individuals. The number and nature of the variables are usually from one block to a different but personal cross blocks must be the same. The main purpose is to examine the relationship between the blocks. Data may be preprocessed to ensure comparability between the variables and the blocks. Before doing the comparison, we need standardized data. In order to block the comparability, To make blocks comparable, a possible strategy is to divide each block by $\sqrt{p_j}$ (Wold, Hellberg, Lundstedt, Sjostrom, & Wold, 1987). This two-step procedure leads to $Trace(X_j^t X_j) = n$ for each block. [Tenenhaus, A., & Tenenhaus, M. (2011)]

We perform a row partition $X = [X_1^t, \ldots, X_i^t, \ldots, X_I^t]^t$ in the multi group framework. In multi group framework, the same set of variables is observed on different groups of individuals. Each $n_i \times p_j$ data matrix $X_i$ is called a group. The number of individuals in each group can be different from one organization to another. Main purpose is to examine the relationship between the variables in different groups. Variables are centered and normalized (i.e. set to unit norm) within each group following the proposal of Kiers and Ten Berge (1994). This preprocessing is similar to multi block analysis, leads to $Trace(X_i^t X_i) = p$ for each group.[ Tenenhaus, A., & Tenenhaus, M. (2011). Regularized generalized canonical correlation analysis. Psychometrika. Retrieved from http://link.springer.com/article/10.1007/s11336-011-9206-8]
3.1 RGCCA optimization

Compared with PLSPM Regularized Generalized Canonical Correlation Analysis (RGCCA) proposed in [Tenenhaus & Tenenhaus (2011)] deals the same problem. While RGCCA is defined as below (3.1):

\[
\begin{align*}
\max_{a_1, a_2, \ldots, a_J} & \sum_{j,k=1; j \neq k}^J c_{jk} g(X_j a_j, X_k a_k) \\
\text{subject to} & \quad \alpha_j' \left( 1 - \tau_j \right) \frac{1}{n} X_j X_j^t + \tau_j I \right) a_j = 1, j = 1, \ldots, J
\end{align*}
\]

(3.1)

The RGCCA is proposed as an optimization problem and in this problem, \( g \) may be defined following three schemes, first one is the Horst scheme proposed in (Kramer (2007)) takes \( g = g(x) \), the second one is Centroid scheme proposed in (Wold, 1985) taking \( g(x) = |x| \); and the third one is Factorial scheme proposed in (Lohmoller (1989)) taking \( g(x) = x^2 \). In this problem, parameter \( \tau_j \) varies between 0 and 1. The vector \( a_j \) (resp. \( y_i = X_j a_j \)) is known as the external power vector (respectively. External components) and \( z_j \) referee is an internal components. The Horst scheme penalizes structural negative correlation between components and centroid and factorial schemes can be seen as attractive alternatives to make the two components are negative correlation.

Optimization problem (3.1) is limited, the three schemes, because they are the most commonly used and multiblock partial least squares regression literature.

From the angle of optimization problem (3.1), the shrinkage parameters \( \tau_j \in [0, 1], j = 1, \ldots, J \) interpolate smoothly between the maximization of the covariance (all \( \tau_j = 1 \)) and the maximization of the correlation (all \( \tau_j = 0 \)). The choice of contract parameters needs to be clear RGCCA analysis’ goal.
Guides for the choice of the definition of the regularization constants provides interpretation results of component properties.

- Based on covariance model ($\tau_j = 1$, a.k.a. RGCCA mode A) are often found in "stable" for the first time (big variance) block component $y_i = X_j a_j, j = 1, ..., J$, at the same time, considering the correlation and the surrounding components (second priority) [Tenenhaus, A., & Tenenhaus, M. (2011)].

- Based on correlation model ($\tau_j = 0$, a.k.a. mode B) give priority to the correlation between the adjacent components, often found unstable block component $y_i = X_j a_j, j = 1, ..., J$. It is worth noticing that RGCCA-Mode B gives exactly the same result to PLSPM-mode B [Tenenhaus, A., & Tenenhaus, M. (2011)].

- $0 < \tau_j < 1$ (a.k.a. mode ridge) yields a compromise between stability and correlation. In our research we did not consider this part we only take the previous two situations into consideration.

These two kinds of motivations (block components with large variance (PCA) and correlation with their neighboring components) are against to each other. An algorithm to solve optimization problem is described in the following algorithm 2. And the algorithm is adopted from Tenenhaus & Tenenhaus (2011).

Algorithm 2 Algorithm for Regularized Generalized Canonical Correlation

Analysis with $0 \leq \tau_j \leq 1$
Step A. Initialization:

Choose arbitrary vectors $\mathbf{a}_j^0$ such that holds:

$$
\mathbf{a}_j^0 = \left[ (\mathbf{a}_j^0)^t \left( \tau_j I + \frac{1}{n} \mathbf{X}_j \mathbf{X}_j^t \right) \right]^{-1/2} \mathbf{a}_j^0
$$

repeat $s = 1, 2 \ldots$

for $j = 1, \ldots, J$ do

Step B. Inner component for $\mathbf{X}_j$

$$
z_j^s = \sum_{k=1}^{j-1} c_{jk} w(\text{cov}(\mathbf{X}_j \mathbf{a}_j^s, \mathbf{X}_k \mathbf{a}_k^{s+1})) \mathbf{X}_k \mathbf{a}_k^{s+1} + \sum_{k=j+1}^{J} c_{jk} w(\text{cov}(\mathbf{X}_j \mathbf{a}_j^s, \mathbf{X}_k \mathbf{a}_k^{s+1})) \mathbf{X}_k \mathbf{a}_k^{s+1}
$$

where $w(x) = 1$ for the Horst scheme, $x$ for the factorial scheme and $\text{sign}(x)$ for the centroid scheme

Step C. Outer weight for block $j$:

$$
\mathbf{a}_j^{s+1} = \left[ (z_j^s)^t \left( \tau_j I + \frac{1}{n} \mathbf{X}_j \mathbf{X}_j^t \right)^{-1} \mathbf{X}_j z_j^s \right]^{-1/2} \left[ \tau_j I + \frac{1}{n} \mathbf{X}_j \mathbf{X}_j^t \right]^{-1} \mathbf{X}_j z_j^s
$$

end for

until convergence


In this algorithm Mode B should provide the same result to PLSPM Mode B as the procedures are really the same while the new Mode A in RGCCA should provide some thing new as it has global optimization function.
Chapter 4. Methodology of GSCA

(Generalized Structured Component Analysis)

GSCA is short for generalized structured component analysis. From the name of the algorithm we can know that it's a component based structural equation model method and can be used as PLS Path Modeling. This algorithm is proposed by Hwang and Takane (2011), allows optimizing a global function using an algorithm called Alternating Least Square algorithm (ALS). GSCA offers a global least squares optimization criterion while PLSPM could not provide. So GSCA could has an overall measure of model fit while it also has all the advantages of PLSPM.

4.1 The model

We assume that $Z$ represents an $N$ by $J$ matrix of observed variables. Suppose $Z$ is wise center and extends to unit variance columns. So, for GSCA model can be expressed as

$$ZV = ZWA + E$$
\[ \Psi = \Gamma A + E \]  
\hfill (4.1)

where \( \Psi = ZV \), and \( \Gamma = ZW \). In (4.1), \( \Psi \) is built with all observed endogenous variables regressed and composite matrix \( N \) by \( T \); \( \Gamma \) is built with all observed exogenous variables regressed and composite \( N \) by \( D \) matrix, \( V \) is a built with a \( J \) by \( T \) matrix of its associated components weight of the endogenous variable, \( W \) is a built with a \( J \) by \( D \) matrix of its associated components weight of the exogenous variable, \( A \) is a \( D \) by \( T \) supermatrix consisting of \( C \) and \( B \). \( C \) is the component loadings matrix relating components to their observed variables, in addition, \( B \) is the path coefficients matrix between components, therefore, \( A = [C, B] \), and \( E \) is a matrix of residuals (error matrix).

To illustrate (4.1), I made an example relationship among variables. It is displayed in Figure below, manifest variables are present in square boxes \((z_i, i = 1, \ldots, 6)\), the latent variables are present in circles \((\gamma_1 \text{ and } \gamma_2)\) or residuals \((e_i \text{ and } d)\), and straight arrows stands for the causality relations, which means that the variable at the end of an arrow affects the variable at the head of the arrow. In the example we can find that each of two latent variables is a linear combination of three observed variables, that is, \( \gamma_1 = \sum_{i=1}^{3} z_i w_i \), and \( \gamma_1 = \sum_{i=4}^{6} z_i w_i \), where \( w_i \) is a component weight. The latent variables are specified to affect the manifest variables, that is, \( z_i = \gamma_1 c_i + e_i \) if \( i \leq 3 \), and \( z_i = \gamma_2 c_i + e_i \) otherwise, where \( c_i \) is a corresponding loading. [Generalized Structured Component Analysis Heungsun Hwang Hec Montreal Yoshio Takane (2004), 69(1), 81–99.]

It shows that all variables can be seen as reflective in some sense similar to PLSPM since they are components based.
It is also found that $\gamma_1$ affects $\gamma_2$, that is, $\gamma_2 = \gamma_1 b + d$, where $b$ is a path coefficient which is called as $\beta$ in PLSPM. Let

\[
Z = [z_1, z_2, z_3, z_4, z_5, z_6]
\]

and

\[
E = [e_1, e_2, e_3, e_4, e_5, e_6, d]
\]

This relationship can then be expressed as

\[
Z = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & w_4 \\
0 & 0 & 0 & 0 & 1 & w_5 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
w_4, w_5, w_6
\]
\[
Z = \begin{bmatrix}
  w_1 & 0 \\
  w_2 & 0 \\
  w_3 & 0 \\
  0 & w_4 \\
  0 & w_5 \\
  0 & w_6
\end{bmatrix}
\begin{bmatrix}
  c_1 & c_2 & c_3 & 0 & 0 & b \\
  0 & 0 & 0 & c_4 & c_5 & c_6
\end{bmatrix} + E
\]

\[
ZV = ZWA + E
\]

\[
\Psi = \Gamma A + E
\]  \hspace{1cm} (4.2)

in (4.2), \(\Psi = [Z, \gamma_2]\), and \(A = [C, b]\), where

\[
C = \begin{bmatrix}
  c_1 & c_2 & c_3 & 0 & 0 & 0 \\
  0 & 0 & 0 & c_4 & c_5 & c_6
\end{bmatrix}
\text{and } b = \begin{bmatrix}
  b
\end{bmatrix}
\]

### 4.2 Estimation of the Parameter

The unknown parameters \(V, W, \text{ and } A\) are estimated by calculating the sum of squares of the residuals, \(E = ZV - ZWA = \Psi - \Gamma A\). And our goal is to make the residuals as small as we can. This function is to minimizing

\[
f = SS(ZV - ZWA)
\]

\[
f = SS(\Psi - \Gamma A), \hspace{1cm} (4.3)
\]

with respect to \(V, W, \text{ and } A\), where \(SS(X) = \text{trace}(X'X)\). The components in \(\Psi\) and/or \(\Gamma\) are normalized for identification purposes, for example, \(\gamma_1'\gamma_1\) in (4.2).
As V, W, and A can contain zero or any fixed elements, we cannot solve (4.3) in an analytic way. On the contrary, Leeuw, Young, & Takane develop an algorithm to solve this problem, alternating least squares algorithm (ALS) (de Leeuw, Young, & Takane, 1976) to minimize the residuals (4.3).

The proposed ALS algorithm by Leeuw, Young, & Takane consists of two steps:

Step one, A is updated for fixed V and W.

Step two, V and W are updated for fixed A.

And the ALS algorithm is pasted below:

_____________________________________________________________________________________________

ALS algorithm

Step 0 (Initialization)

For $j = 1, \ldots, J$

choose the $j$th arbitrary weight vector $(w_j^0)$,

$$\eta_j^0 = \frac{x_j w_j^0}{||x_j w_j^0||},$$

End

For $s = 0, 1, 2 \ldots$ (until convergence)

Step 1 (internal Estimation)

For $j = 1, \ldots, J$

$\alpha_j = 1$, if Mode A

$\alpha_j = 0$, if Mode B
\begin{align*}
  f_j^s &= \sum_{q=1}^{Q_j} e_{jq} \eta_{qv}^s \\
  \text{where element } e_{jq} \text{ is the } q\text{th element of} \\
  e_j^s &= (\alpha_j w_j^s w_j^s I + (1 - \alpha_j)X_j X_j)^{-1} X_j X_j^s \eta_j^s \\
  \text{End}
\end{align*}

Step 2 (External Estimation)

For \( j = 1, \ldots, J \)

\begin{align*}
  \alpha_j &= 1, \text{if Mode A} \\
  \alpha_j &= 0, \text{if Mode B} \\
  w_j^{s+1} &= (\alpha_j f_j^s f_j^s I + (1 - \alpha_j)X_j X_j)^{-1} X_j f_j^s, \\
  \eta_j^{s+1} &= \frac{X_j w_j^{s+1}}{||X_j w_j^{s+1}||} \\
  \text{End}
\end{align*}

Check if \( \phi^s - \phi^{s+1} < 0.00001 \). If not, go back to

Step 1.

End


With the ALS algorithm procedure we can obtain the unknown parameters.
4.3 Quality index

In GSCA, the overall fit of a hypothesized model is measured by the total variance of all the endogenous variables explained by the specified model predictions. [Hwang, H., Takane, Y., & Malhotra, N. (2007)] This is given by

\[ Fit = 1 - \frac{SS(\Psi - \Gamma \Lambda)}{SS(\Psi)}. \] (4.9)

This fit with index of the range is 0 to 1. The larger the fitness, the endogenous variable variance is explained by model. It is a function of value is the sum of squared residuals. The difference between models and data are summarized. This overall fit with measuring the whole evaluation of the adequacy of the model (Bollen, 1989, p. 256) and we can compare different models. Even so, it is also important to check the individual parameter estimation of local goodness-of-fit (Bollen, 1989, p. 281). For example, we can check the loadings is equal to the correlation between observed variables and their components and squared multiple correlations (equal to the squared loadings) for individual observed variables to evaluate the adequacy of components. We may also look at the standard errors or confidence intervals by using the reliability of parameter estimation to check them [Hwang, H., Takane, Y., & Malhotra, N. (2007)]. Besides such statistical measures of model fit, Statistic model is suitable for nonstatistical
in addition to these considerations, such as model to explain ability often play a role in the model evaluation, although they are generally more difficult to prove, because they are largely subjectiv (Browne & Cudeck, 1993, p. 136). [Hwang, H., Takane, Y., & Malhotra, N. (2007). MULTILEVEL GENERALIZED STRUCTURED COMPONENT ANALYSIS, 34(2), 95–109.]

We use the method of bootstrap to estimate the standard error in GSCA (Efron, 1982). And then we can evaluate the reliability of the parameter estimates with the bootstrapped standard errors or confidence intervals.

To test some hypotheses on parameters, we can implement linear formulas into the model. Linear formulas may be designated by reparametrization or zero space method (Bockenholt & Takane, 1994; "Ihkane, Yanai, & Mayekawa, 1991).

In GSCA, linear formulas are all imposed by the reparametrization method.
Chapter 5. Overall Comparison of The methods

From the concepts of the three algorithms we already know that the limit of PLSPM is that it can’t provide global criterion and it doesn’t have criterion optimization. While the RGCCA uses Max Compound Bivariate Covariance to optimize the criterion and GSCA uses alternated least squares.

<table>
<thead>
<tr>
<th></th>
<th>PLSPM</th>
<th>RGCCA</th>
<th>GSCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global criterion</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Criterion optimization type</td>
<td>NO</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Manages partial effects between groups</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>No probabilistics assumption</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Convergence of criterion</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Extracts several components / group</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Group size insensitive</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

1:Max Compound Bivariate Covariance
RGCCA and GSCA have global optimization functions that mean they could perform GOF to measure the model quality. And also the GSCA and RGCCA have convergence of criterion.

PLSPM Algorithm (Algorithm 1) and RGCCA algorithm (Algorithm 2) are equivalent when $\tau_j = 0$ for all blocks. We could expect they have exactly the same latent variables.

Before we do the experiment with real dataset we could expect the result as:

<table>
<thead>
<tr>
<th>Mode</th>
<th>GSCA</th>
<th>PLSPM</th>
<th>RGCCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode A</td>
<td>$GSCA \approx$</td>
<td>PLSPM</td>
<td>$\neq$ RGCCA</td>
</tr>
<tr>
<td>Mode B</td>
<td>$GSCA \approx$</td>
<td>PLSPM</td>
<td>$=$ RGCCA</td>
</tr>
</tbody>
</table>

The results in Mode A: GSCA should be almost the same as PLSPM but not exactly the same which the RGCCA, which we says new Mode A should not be the same as PLSPM.

And in Mode B the GSCA should also be almost the same as PLSPM, while the RGCCA should be exactly the same as PLSPM.
Chapter 6. Experiment with survey data

In the experiment we take the UPC graduate students’ survey data.

In 2008 the ICT schools of the UPC started a survey to monitor the satisfaction of alumni three years after their graduation with the performed studies. 147 alumni answered the questionnaire about satisfaction and their drivers. The list of questions were the following:

Ima1   It’s the best to study informatics
Ima2   It is internationally recognized
Ima3   It has a wide range of courses
Ima4   The teachers are good
Ima5   The facilities and equipment are good
Ima6   Is leading research
Ima7   It is highly regarded by companies
Ima8   Can adapt to new needs and technologies
quaf1  Quality of the studies: the theoretical base
quaf2  Quality of the studies: the technical competences
quaf3  Quality of the studies: the applied training
qutr1  Training in business management
qutr2  The written and oral communication skills
qutr3  Planning and time management acquired
The ability to work in teams

Allowed me to find a well-paid job

I have prospects for improvement and promotion

Allowed me to find a job that motivates me

The training received is the basis on which I will build my career

I am satisfied with the training received

I am satisfied with my current situation

I think I'll have a good professional career

I think in the prestige of my work

Questions were recorded in a scale from 0 to 10 (0 = indicates the lowest value whereas 10 is the maximum).

From the data we can put the questions into five blocks which means we can get five latent variables to analysis.

The first latent variable is Image of the school, which is built by 8 indicators which are "ima1", "ima2", "ima3", "ima4", "ima5", "ima6", "ima7", "ima8"; the second latent variable is the quality of the studies, which is built by 3 indicators which are "quaf1", "quaf2", "quaf3"; the third latent variable is skills trained in school, which is built by 4 indicators which are "qutr1", "qutr2", "qutr3", "qutr4"; the fourth latent variable is the value they get after graduated, which is built by 4 indicators, which are "val1", "val2", "val3", "val4"; the last latent variable we can get is the satisfaction of the graduates to the school, which is built by 4 indicators which are "sat1", "sat2", "sat3", "sat4".
6.1 The inner model

The inner model is following the standard ECSI model. It’s good to measure satisfaction of alumni. Then we can build the inner model as described in the figure.

6.2 Assessing unidimensionality

As the foundation of the work is unidimensional, we need to do the PCA of each block to make sure the eigenvalue in the first dimension is bigger than 1 and from the second dimension the eigenvalue is decreased to less than 1.
PCA of each block

Block image

Variables factor map (PCA)

Eigenvalue of the imag

Correlation of the manifest variables in each block

<table>
<thead>
<tr>
<th>Manifest variable</th>
<th>Dim1</th>
<th>Dim 2</th>
<th>Dim 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
From the plot of the image block we can find that all the manifest variables are correlated and from the correlation table we can find that they are strong related to each other. So we can take all of them into the outer model. And the plot of the eigenvalues has a big jump from the second dimension. The eigenvalue of the first dimension is bigger than 1 and the second dimension is smaller than 1. This means that this block suits for unidimensional analysis.

**Block val**

<table>
<thead>
<tr>
<th></th>
<th>ima1</th>
<th>ima2</th>
<th>ima3</th>
<th>ima4</th>
<th>ima5</th>
<th>ima6</th>
<th>ima7</th>
<th>ima8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.7896</td>
<td>-0.16523</td>
<td>0.3728951</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.8211</td>
<td>-0.29793</td>
<td>0.0001086</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.8719</td>
<td>0.12341</td>
<td>-0.0142725</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.7049</td>
<td>0.55404</td>
<td>0.3357829</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.7260</td>
<td>0.40825</td>
<td>-0.4284601</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.8119</td>
<td>-0.21687</td>
<td>-0.3096604</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.8204</td>
<td>-0.27029</td>
<td>0.0552133</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.7948</td>
<td>-0.02714</td>
<td>-0.0020327</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Variables factor map (PCA)
From the plot of the image block we can find that all the manifest variables are correlated besides the val1 but from the correlation table we can find that they are strongly related to each other. So we can take all of them into the outer model. And the plot of the eigenvalues has a big jump from the second dimension. The eigenvalue of the first dimension is bigger than 1 and the second dimension is smaller than 1. This means that this block suits for unidimensional analysis.
Block qutr

Variables factor map (PCA)

Eigenvalue of the qutr

Index
<table>
<thead>
<tr>
<th>Manifest variable</th>
<th>Dim1</th>
<th>Dim 2</th>
<th>Dim 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>0.8153</td>
<td>-0.43032</td>
<td>0.1542990</td>
</tr>
<tr>
<td>10</td>
<td>0.8375</td>
<td>-0.38015</td>
<td>-0.0861015</td>
</tr>
<tr>
<td>11</td>
<td>0.7153</td>
<td>0.51618</td>
<td>0.4659414</td>
</tr>
<tr>
<td>12</td>
<td>0.7492</td>
<td>0.40043</td>
<td>-0.5165442</td>
</tr>
</tbody>
</table>

From the plot of the qutr block we can find that all the manifest variables qutr4 and qutr5 are correlated while qutr2 and qutr3 are related. And from the correlation table we can find that they are strong related to each other. So we can take all of them into the outer model. And the plot of the eigenvalues has a big jump from the second dimension. The eigenvalue of the first dimension is bigger than 1 and the second dimension is smaller than 1. This means that this block suits for unidimensional analysis.

**Block quaf**
Correlation of the manifest variables in each block

<table>
<thead>
<tr>
<th>Manifest variable</th>
<th>Dim1</th>
<th>Dim2</th>
<th>Dim3</th>
</tr>
</thead>
</table>

Variables factor map (PCA)

Eigenvalue of the quaf

Index

pca_quaf eigenvalue

Dim 1 (66.43%)

Dim 2 (24.23%)
From the plot of the quaf block we can find that all the manifest variables are correlated besides quaf1 and from the correlation table we can find that quaf1 is not very strong related to the other two variables. But 0.6559 is acceptable to us as we don’t have too much variables. So we can take all of them into the outer model. And the plot of the eigenvalues has a jump from the second dimension. The eigenvalue of the first dimension is bigger than 1 and the second dimension is smaller than 1. This means that this block suits for unidimensional analysis.

**Block sat**
Correlation of the manifest variables in each block

<table>
<thead>
<tr>
<th>Manifest variable</th>
<th>Dim1</th>
<th>Dim 2</th>
<th>Dim 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 Sat1</td>
<td>0.5909</td>
<td>0.79567</td>
<td>0.1237487</td>
</tr>
<tr>
<td>21 Sat2</td>
<td>0.8906</td>
<td>-0.08089</td>
<td>-0.2788630</td>
</tr>
<tr>
<td>22 Sat3</td>
<td>0.8666</td>
<td>-0.20645</td>
<td>-0.3153123</td>
</tr>
<tr>
<td>23 Sat4</td>
<td>0.7268</td>
<td>-0.30163</td>
<td>0.6170366</td>
</tr>
</tbody>
</table>

From the plot of the image block we can find that all the manifest variables are correlated besides the sat1 and from the correlation table we can find that sat1 is not strong related to the others. But we still want to take this variable into the model. So we can take all of them into the outer model. And the plot of the eigenvalues has a big jump from the second dimension. The eigenvalue of the first dimension is bigger than 1 and the second dimension is smaller than 1. This means that this block suits for unidimensional analysis.
And from the correlation of the indicators we can see that all the indicators in each block are related. So we will take it into our model to do the experiment in two parts, first we compare the results in Mode A and then we compare the results of the three algorithms in Mode B.

6.3 Comparison in Mode A

After implement the algorithm with R packages “plspm”, “Matrix”, “ASGSCA”, we can get the results as bellow. We are going to compare the weights of the indicators, in path coefficient, the loadings and the intengibles.

The latent variables in Mode A

We do the comparison of latent variables between GSCA and PlSPM and also between RGCCA and PLSPM.

The latent variables are computed with the corresponding weights of each manifest variable and the data.

\[(\xi|X_x) = \Pi_x X_x\]

The latent variables are computed and we can compute the mean and the standard deviation of each latent per each method. The GSCA and PLSPM don’t have too much different between each other in the mean and standard deviation. But the RGCCA is quite different from the other two methods.
And we measure the correlation between the LVs using “cor”.

\[
\text{Cor}(\text{GSCA, PLSPM})
\]

\[
\text{Cor}(\text{RGCCA, PLSPM})
\]

And we get the data tables below:

We can find that the cross blocks of all the LVs are quit related to each other.

They are quite close to 1 in the GSCA and PLSPM, we can say that these two algorithms almost described the same latent concepts.
And between RGCCA and PLSPM the cross blocks are mostly equal to 1, which means that these two algorithms are describing the same thing.

### GSCA and PLSPM

<table>
<thead>
<tr>
<th></th>
<th>Image</th>
<th>Quality</th>
<th>Skills</th>
<th>Value</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Image</td>
<td>0.9997810</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Quality</td>
<td></td>
<td>0.9995340</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Skills</td>
<td></td>
<td></td>
<td>0.9999746</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Value</td>
<td></td>
<td></td>
<td>0.9996218</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Satisfaction</td>
<td></td>
<td></td>
<td></td>
<td>0.9925729</td>
</tr>
</tbody>
</table>

### RGCCA and PLSPM

RGCCA new mode A and PLSPM they give different means and standard deviations but they are measuring the same thing.

<table>
<thead>
<tr>
<th></th>
<th>Image</th>
<th>Quality</th>
<th>Skills</th>
<th>Value</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Image</td>
<td>1.0000000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Quality</td>
<td></td>
<td></td>
<td>0.9999998</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Skills</td>
<td></td>
<td></td>
<td>1.0000000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Value</td>
<td></td>
<td></td>
<td></td>
<td>0.9999965</td>
</tr>
<tr>
<td>5</td>
<td>Satisfaction</td>
<td></td>
<td></td>
<td></td>
<td>0.9999962</td>
</tr>
</tbody>
</table>
The weights in all these three algorithms are positive. All these three methods deal with weight well.

<table>
<thead>
<tr>
<th></th>
<th>PLSPM</th>
<th>GSCA</th>
<th>RGCCA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Weights</td>
<td>Loadings</td>
<td>Weights</td>
</tr>
<tr>
<td>ima1</td>
<td>0.195</td>
<td>0.804</td>
<td>0.1733</td>
</tr>
<tr>
<td>ima2</td>
<td>0.146</td>
<td>0.812</td>
<td>0.1466</td>
</tr>
<tr>
<td>ima3</td>
<td>0.182</td>
<td>0.875</td>
<td>0.1657</td>
</tr>
<tr>
<td>ima4</td>
<td>0.154</td>
<td>0.718</td>
<td>0.1565</td>
</tr>
<tr>
<td>ima5</td>
<td>0.146</td>
<td>0.727</td>
<td>0.1588</td>
</tr>
<tr>
<td>ima6</td>
<td>0.14</td>
<td>0.799</td>
<td>0.1506</td>
</tr>
<tr>
<td>ima7</td>
<td>0.141</td>
<td>0.811</td>
<td>0.1469</td>
</tr>
<tr>
<td>ima8</td>
<td>0.159</td>
<td>0.793</td>
<td>0.1629</td>
</tr>
<tr>
<td>qutr2</td>
<td>0.366</td>
<td>0.821</td>
<td>0.3462</td>
</tr>
<tr>
<td>qutr3</td>
<td>0.304</td>
<td>0.826</td>
<td>0.3030</td>
</tr>
<tr>
<td>qutr4</td>
<td>0.261</td>
<td>0.7</td>
<td>0.2943</td>
</tr>
<tr>
<td>qutr5</td>
<td>0.351</td>
<td>0.768</td>
<td>0.3375</td>
</tr>
<tr>
<td>quaf1</td>
<td>0.438</td>
<td>0.731</td>
<td>0.4316</td>
</tr>
<tr>
<td></td>
<td>0.381</td>
<td>0.848</td>
<td>0.3745</td>
</tr>
<tr>
<td>---</td>
<td>-------</td>
<td>-------</td>
<td>--------</td>
</tr>
<tr>
<td>quaf2</td>
<td>0.423</td>
<td>0.852</td>
<td>0.4308</td>
</tr>
<tr>
<td>quaf3</td>
<td>0.362</td>
<td>0.775</td>
<td>0.3614</td>
</tr>
<tr>
<td>val1</td>
<td>0.334</td>
<td>0.86</td>
<td>0.3012</td>
</tr>
<tr>
<td>val2</td>
<td>0.285</td>
<td>0.761</td>
<td>0.2985</td>
</tr>
<tr>
<td>val3</td>
<td>0.299</td>
<td>0.728</td>
<td>0.3204</td>
</tr>
<tr>
<td>val4</td>
<td>0.515</td>
<td>0.782</td>
<td>0.3973</td>
</tr>
<tr>
<td>sat1</td>
<td>0.288</td>
<td>0.825</td>
<td>0.3112</td>
</tr>
<tr>
<td>sat2</td>
<td>0.291</td>
<td>0.781</td>
<td>0.3228</td>
</tr>
<tr>
<td>sat3</td>
<td>0.213</td>
<td>0.636</td>
<td>0.2719</td>
</tr>
</tbody>
</table>

In a reflective model each manifest variable is related to the corresponding latent variable by a simple regression mode.

From the weight/loading table we can get that the results of PLSPM are quit similar to GSCA, that’s because the GSCA is using the alternated least squares algorithm which is adopted from PLSPM algorithm. But the RGCCA results are quit different because the way it does normalization is not the same as the other two algorithms. But even though, we can identify that these algorithm in Mode A are measuring the same thing.

The quality of PLSPM:
<table>
<thead>
<tr>
<th>Type</th>
<th>R2</th>
<th>Block_Communality</th>
</tr>
</thead>
<tbody>
<tr>
<td>ima</td>
<td>Exogenous</td>
<td>0.63</td>
</tr>
<tr>
<td>qutr</td>
<td>Exogenous</td>
<td>0.609</td>
</tr>
<tr>
<td>quaf</td>
<td>Exogenous</td>
<td>0.66</td>
</tr>
<tr>
<td>val</td>
<td>Endogenous</td>
<td>0.612</td>
</tr>
<tr>
<td>sat</td>
<td>Endogenous</td>
<td>0.577</td>
</tr>
</tbody>
</table>

The block communities are almost the same between the 3 algorithms. That’s because with our structural equation model using these three algorithms we get almost the same quality.
The path coefficients in Mode A

In PLSPM

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>PLSPM</th>
<th>GSCA</th>
<th>RGCCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ima</td>
<td>-&gt; val</td>
<td>0.3444</td>
<td>0.3333</td>
<td>0.3845</td>
</tr>
<tr>
<td>ima</td>
<td>-&gt; sat</td>
<td>0.3067</td>
<td>0.2869</td>
<td>0.3648</td>
</tr>
<tr>
<td>qutr</td>
<td>-&gt; val</td>
<td>-0.0422</td>
<td>-0.0312</td>
<td>-0.0709</td>
</tr>
<tr>
<td>qutr</td>
<td>-&gt; sat</td>
<td>-0.0163</td>
<td>-0.0088</td>
<td>-0.0210</td>
</tr>
<tr>
<td>quaf</td>
<td>-&gt; val</td>
<td>0.2178</td>
<td>0.2182</td>
<td>0.2207</td>
</tr>
<tr>
<td>quaf</td>
<td>-&gt; sat</td>
<td>0.1239</td>
<td>0.0624</td>
<td>0.2229</td>
</tr>
<tr>
<td>val</td>
<td>-&gt; sat</td>
<td>0.5657</td>
<td>0.5848</td>
<td>0.4686</td>
</tr>
</tbody>
</table>

The direct effects are the direct path coefficient and in GSCA two negative path coefficients while in PLSPM we have also two negative path coefficients. The two algorithms perform in the same way. The path coefficient simulation in RGCCA is not the same method as PLSPM or GSCA. It only can provide direct path
coefficients. And as the normalization way is different from the other two algorithms the path coefficient should not be compared.

6.4 Comparison in Mode B

The latent variables in Mode B

We do the comparison of latent variables between GSCA and PLSPM and also between RGCCA and PLSPM.

In a formative model each manifest variable is related to the corresponding latent variable by a simple regression model, i.e.:

\[ x_{pq} = \lambda_{pq0} + \lambda_{pq} \xi_q \]

where \( \lambda_{pq} \) is the loading associated to the \( p \)-th manifest variable in the \( q \)-th block.

We can generate the LVs as the same as in Mode A.

In Mode B all the latent variables seem don’t have much different with the mean and standard deviation.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Sd</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLSPM</td>
<td>8.022769</td>
<td>1.915558</td>
</tr>
<tr>
<td>RGCCA</td>
<td>8.022769</td>
<td>1.915558</td>
</tr>
<tr>
<td>GSCA</td>
<td>8.005102</td>
<td>1.88883</td>
</tr>
<tr>
<td><strong>Quality</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLSPM</td>
<td>5.918025</td>
<td>2.072576</td>
</tr>
<tr>
<td>RGCCA</td>
<td>5.918025</td>
<td>2.072576</td>
</tr>
<tr>
<td>GSCA</td>
<td>6.076695</td>
<td>2.072067</td>
</tr>
</tbody>
</table>
And we get the correlation of the latent variables below:

We can find that the cross blocks of all the LVs are quit related to each other. They are quite close to 1 in the GSCA and PLSPM, we can say that these two algorithms almost described the same latent concepts. And between RGCCA and PLSPM the cross blocks are all equal to 1, which means that these two algorithms are describing the really same thing.

### GSCA and PLSPM

<table>
<thead>
<tr>
<th></th>
<th>Image</th>
<th>Quality</th>
<th>Skills</th>
<th>Value</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Image</td>
<td>0.9854258</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Quality</td>
<td>0.9990651</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Skills</td>
<td>0.9997642</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Value</td>
<td></td>
<td>0.9928587</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Satisfaction</td>
<td></td>
<td></td>
<td>0.9966040</td>
<td></td>
</tr>
</tbody>
</table>
The next thing we can get from the loadings/weights in Mode B is that the PLSPM and RGCCA give exactly the same results. And that’s just what we expected before doing the experiment. RGCCA equates to PLSPM in Mode B. The next thing is that all the three algorithms in Mode B give bad loadings especially GSCA while it performs very good in Mode A. And also PLSPM and RGCCA give weak loadings than Mode A. And we have negative weights in each algorithm. The GSCA have 3 negative weights and it’s better than the other two algorithms which each have 5 negative weights.
The quality of the model

In PLSPM the R2 is 0.282 for val and 0.772 for sat. While in Mode A the R2s are 0.236 for val and 0.682 for sat.
<table>
<thead>
<tr>
<th>Type</th>
<th>R2</th>
<th>Block_Communality</th>
</tr>
</thead>
<tbody>
<tr>
<td>ima</td>
<td>Exogenous</td>
<td>0</td>
</tr>
<tr>
<td>qutr</td>
<td>Exogenous</td>
<td>0</td>
</tr>
<tr>
<td>quaf</td>
<td>Exogenous</td>
<td>0</td>
</tr>
<tr>
<td>val</td>
<td>Endogenous</td>
<td>0.283</td>
</tr>
<tr>
<td>sat</td>
<td>Endogenous</td>
<td>0.772</td>
</tr>
</tbody>
</table>

**PLSPM table**

<table>
<thead>
<tr>
<th>Type</th>
<th>R2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ima</td>
<td>Exogenous</td>
</tr>
<tr>
<td>qutr</td>
<td>Exogenous</td>
</tr>
<tr>
<td>quaf</td>
<td>Exogenous</td>
</tr>
<tr>
<td>val</td>
<td>Endogenous</td>
</tr>
<tr>
<td>sat</td>
<td>Endogenous</td>
</tr>
</tbody>
</table>

**GSCA R2**
The R2 between GSCA and PLSPM are almost the same. In RGCCA, the results are just the same as PLSPM in Mode B which we explained before.

<table>
<thead>
<tr>
<th>Type</th>
<th>R2</th>
<th>Block_Communality</th>
</tr>
</thead>
<tbody>
<tr>
<td>ima</td>
<td>Exogenous</td>
<td>0</td>
</tr>
<tr>
<td>qutr</td>
<td>Exogenous</td>
<td>0</td>
</tr>
<tr>
<td>quaf</td>
<td>Exogenous</td>
<td>0</td>
</tr>
<tr>
<td>val</td>
<td>Endogenous</td>
<td>0.283</td>
</tr>
<tr>
<td>sat</td>
<td>Endogenous</td>
<td>0.772</td>
</tr>
</tbody>
</table>

*RGCCA table*

**Path coefficients in Mode B**

In PLSPM, we also have two negative direct path coefficients.

<table>
<thead>
<tr>
<th></th>
<th>PLSPM</th>
<th>GSCA</th>
<th>RGCCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ima</td>
<td>-&gt; val</td>
<td>0.3889</td>
<td>0.39308</td>
</tr>
<tr>
<td>ima</td>
<td>-&gt; sat</td>
<td>0.3798</td>
<td>0.36249</td>
</tr>
<tr>
<td>qutr</td>
<td>-&gt; val</td>
<td>-0.0813</td>
<td>-0.08796</td>
</tr>
<tr>
<td>qutr</td>
<td>-&gt; sat</td>
<td>0.027</td>
<td>0.02119</td>
</tr>
<tr>
<td>quaf</td>
<td>-&gt; val</td>
<td>0.2449</td>
<td>0.23304</td>
</tr>
</tbody>
</table>
While in GSCA Mode B we only have one negative path coefficient. But the total effects on GSCA still have two negative effects. And the Val to Sat effect is strongly weakened as PLSPM. The path coefficients are really the same between RGCCA and PLSPM, and this is what we are expected to get.

**The latent variables in Mode B**

We do the comparison of latent variables between GSCA and PLSPM and also between RGCCA and PLSPM.

We can generate the LVs as the same as in Mode A.

And we get the data tables below:

We can find that the cross blocks of all the LVs are quit related to each other. They are quite close to 1 in the GSCA and PLSPM, we can say that these two algorithms almost described the same latent concepts.

And between RGCCA and PLSPM the cross blocks are all equal to 1, which means that these two algorithms are describing the really same thing.

<table>
<thead>
<tr>
<th>quaf</th>
<th>-&gt;</th>
<th>sat</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.3237</td>
<td>0.30635</td>
<td>0.3237</td>
</tr>
<tr>
<td>val</td>
<td>-&gt;</td>
<td>sat</td>
<td>0.3169</td>
<td>0.35062</td>
<td>0.3169</td>
</tr>
</tbody>
</table>
Conclusion

The experiment of comparison with these three algorithms has the results as we expected before. And the PLSPM Mode A still provides stable and reasonable outputs (loadings and weights). In our experiment we didn't take RGCCA optimization $\tau$ into consideration. We only consider Moda A and Mode B, not the mix Mode. By changing $\tau = 1$, we step into Mode A and taking $\tau = 0$, we get into Mode B. And we can also input $\tau$ as any number between 0 to 1. And we can get an optimal $\tau$ and optimal model. That’s the advantage PLSPM doesn’t have.

GSCA performs also well in experiment. As it is based on ALS algorithm which we can consider that it was generated from PLS. And that’s the reason why they have the similar results in both Modes. And GSCA has global optimization criterion, which is consistently minimized to obtain the estimates of model parameters.

And the most important thing is that in RGCCA even the new Mode A gives different means and standard deviations than the other two algorithms PLSPM, GSCA, but the coefficient of the latent variables are close to 1, which means that even they are based on different methodologies, they are measuring the same latent concepts with our data.

After all, even these three algorithms are proposed in different time, but to our dataset, the performance didn’t have too much difference among each other especially in discovering latent variables. So we can select the algorithm as which we are more familiar when implement the analysis. And also the convenient of
algorithm implementation should be taken into consideration. During these three algorithm I will choose PLSPM as it's easy to implement and also we have R package(PLSPM) which could provide beautiful plot of the inner model which is quite useful to construct the model.
Reference


Hwang, H., & Takane, Y. (2002). Structural equation modeling by extended redundancy analysis. In S. Nishisato, Y. Baba,


doi:10.1016/j.csda.2004.03.005


Appendix 1

The key steps to construct the comparisons:

In this comparison work we used a lot of R functions which are mostly from “plspm”, “matrixpls”, “rgcca”, “PCA”.

https://cran.r-project.org/web/packages/plspm/index.html
https://cran.r-project.org/web/packages/matrixpls/index.html
https://cran.r-project.org/web/packages/RGCCA/index.html

We can get the information of packages using the links above to take deeper look inside the packages.

After we get the packages we need to build the inside and outside models., which in our case the models are built like this:

Inner model:

<table>
<thead>
<tr>
<th></th>
<th>ima</th>
<th>qutr</th>
<th>quaf</th>
<th>val</th>
<th>sat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ima</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>qutr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The outer model is much easier, we just need to connect the corresponding manifest variable to the LVs.

Which in R code is:
After we have the model we need to do the unidimensional analysis to each LVs.

To do this we need to do PCA of each block then we can get the eigenvalue of each block and we want to check how the eigenvalue decreased.

```r
ima <- c(0,0,0,0,0)
qutr<- c(0,0,0,0,0)
quaf <- c(0,0,0,0,0)
val <- c(1,1,1,0,0)
sat <- c(1,1,1,1,0)

inner.models <- rbind(ima,qutr,quaf,val,sat)
outer.models <- list(1:8,13:16,9,11,17:20,21:24)

#do the PCA of each block
pc.cr1 <- PCA(X_imag)
pc.cr1$var$cor

pc.cr2 <- PCA(X_qutr)
pc.cr2$var$cor

pc.cr3 <- PCA(X_quaf)
pc.cr3$var$cor

pc.cr4 <- PCA(X_val)
pc.cr4$var$cor

pc.cr5 <- PCA(X_sat)
pc.cr5$var$cor
```

The eigenvalues of the first dimension are all bigger than 1 and the second dimension are smaller than 1. So we can continue to the algorithms.
PLSPM

In PLSPM, it's quite easy to implement with the data. First we need to standardize the data, then following the model we can build the implement in both Mode A and Mode B. The only thing we need to change when turning Mode A to Mode B is changing the mode function.

```r
datas<-scale(data)
modes.A <- c("A","A","A","A","A")
modes.B<-c("B","B","B","B","B")
pls.mobile2 <- plspm(datas, inner.models, outer.models, modes.A, scaled=FALSE)
pls.mobileb <- plspm(datas, inner.models, outer.models, modes.B, scaled=FALSE)
```

Then we can get the all the information we want with summary the results:

```
> pls.mobileb

Partial Least Squares Path Modeling (PLS-PM)
---------------------------------------------
  NAME     DESCRIPTION
 1 $outer_model       outer model
 2 $inner_model       inner model
 3 $path_coefs        path coefficients matrix
 4 $scores            latent variable scores
 5 $crossloadings     cross-loadings
 6 $inner_summary     summary inner model
 7 $effects           total effects
 8 $unidim            unidimensionality
 9 $gof               goodness-of-fit
10 $boot              bootstrap results
11 $data              data matrix
```
And we are interested with the $\text{path\_coefs}$, $\text{scores}$, $\text{outer\_model}$, $\text{inner\_model}$. From which we can get the loadings, weights, path coefficients and latent variables.

**RGCCA**

In RGCCA, things are different as PLSPM. The package could not provide enough information. RGCCA defines the input data in another format:

The inner model and outer model should be defined as following:

```r
X_imag = as.matrix(datas[,1:8])
X_qutr = as.matrix(datas[,13:16])
X_quaf = as.matrix(datas[,9:11])
X_val = as.matrix(datas[,17:20])
X_sat = as.matrix(datas[,21:24])
A = list(X_imag, X_qutr,X_quaf,X_val,X_sat)
#Define the design matrix (output = C)
C = matrix(c(0, 0, 0, 1, 1,0, 0, 0, 1, 1,1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0), 5, 5)
```

A is the outer model and C is the inner model.

The way to change from Mode A to Mode B is change the tau in the function:

```r
#mode a
result.rgcca = rgcca(A, C, tau = rep(1,5), scheme = "centroid", scale = TRUE)

#mode b
result.rgccak = rgcca(A, C, tau = rep(0,5), scheme = "centroid", scale = TRUE)
```

When all the taus are equal to 1 we go to mode A while all taus are equal to 0 we go to mode B.
The results only provide the loadings and weights. So if we want to computer the path coefficients and latent concepts we need to do like this:

**Path coefficients:**

\[((t(result.rgcca$Y[[1]])%*%result.rgcca$Y[[1]]))^(-1))*(result.rgcca$Y[[1]][,1]%*%result.rgcca$Y[[4]][,1])\]

**Latent variables:**

#latent variable in mode A
result.rgcca$Y[[1]]
rgcca.score.a<-cbind(result.rgcca$Y[[1]],result.rgcca$Y[[2]],result.rgcca$Y[[3]],result.rgcca$Y[[4]],result.rgcca$Y[[5]])
colnames(rgcca.score.a)<-c("ima", "qutr", "quaf", "val", "sat")
Wr1 <- t(attr(matrixpls.res.rgcca,"W"))
resr1.imag <- scale(X_imag %*% Wr1[1:8,1])
resr1.qutr<-scale(X_qutr %*% Wr1[12:15,2])
resr1.quaf<-scale(X_quaf %*% Wr1[9:11,3])
resr1.val<-scale(X_val %*% Wr1[16:19,4])
resr1.sat<-scale(X_sat %*% Wr1[20:23,5])

**GSCA**

In GSCA we can use the inner model of PLSPM to do the analysis. But we need to construct the outer model in another format:
The same as RGCCA we need to compute the latent variables:

```r
W0 <- matrix(c(rep(1,8),rep(0,26),rep(1,4),rep(0,16),rep(1,3),rep(0,27),rep(1,4),rep(0,23),rep(1,4)), nrow=23, ncol=5)
GSCA.res <- GSCA(as.data.frame(datas2), W0, inner.models, estim=TRUE, path.test=FALSE, latent.names=rownames(inner.models))
formative <- matrix(c(0,5,23, dimnames = list(colnames(rownames(inner.models)), colnames(datas2))))
inner <- inner.models
reflective <- matrix(c(rep(1,8),rep(0,26),rep(1,4),rep(0,16),rep(1,3),rep(0,27),rep(1,4),rep(0,23),rep(1,4)), nrow=23, ncol=5, dimnames = list(colnames(datas2), colnames(rownames(inner.models))))

# Estimate using alternating least squares
matrixpls.res2 <- matrixpls(cov(as.data.frame(datas2)), model=list(inner = inner, reflective = reflective, formative = formative), outerEstimators = outer.GSCA, innerEstimator = inner.GSCA)
W <- t(attr(matrixpls.res2, "W"))
result2.imag <- scale(X_imag %*% W[1:8,1])
res2.qutr <- scale(X_qutr %*% W[12:15,2])
res2.quaf <- scale(X_quaf %*% W[9:11,3])
res2.val <- scale(X_val %*% W[16:19,4])
res2.sat <- scale(X_sat %*% W[20:23,5])
gsca.score.a <- cbind(result2.imag, res2.qutr, res2.quaf, res2.val, res2.sat)
```

At last we need to compare the latent variables between the algorithms, we can do like this:

```r
# for instance comparing GSCA and PLSPM
cor(gsca.score.a, pls.mobile2$scores)
```
Appendix 2

Original codes

library(calibrate)
library(pls)
library(FactoMineR)
library(matrixpls)
library(ASGSCA)
library(RGCCA)
library(lavaan)

# Read the data
library(gdata)
data <- read.xls("sat_ICT2008.xlsx", header=TRUE)
print(data)

#ima1 It's the best to study informatics
#ima2 It is internationally recognized
#ima3 It has a wide range of courses
#ima4 The teachers are good
#ima5 The facilities and equipment are good#
#ima6 Is leading research
#ima7 It is highly regarded by companies
#ima8 Can adapt to new needs and technologies
#quaf1 Quality of the studies: the theoretical base
#quaf2 Quality of the studies: the technical competences
#quaf3 Quality of the studies: the applied training
#qutr1 The ability to solve problems from
#qutr2 Training in business management
#qutr3 The written and oral communication skills
#qutr4 Planning and time management acquired
#qutr5 The ability to work in teams
#val1 Allowed me to find a well-paid job
#val2 I have prospects for improvement and promotion
#val3 Allowed me to find a job that motivates me
#val4 The training received is the basis on which I will build my career
#sat1 I am satisfied with the training received
#sat2 I am satisfied with my current situation
#sat3 I think I'll have a good professional career
#sat4 I think in the prestige of my work
# Extract a bootstrap sample of the data, with a specified random set

cor.varPsi = cor(datas, pc.cr$scores)
nd = 6
varimax(cor.varPsi[, 1:nd])

data2 <- data[, c(2, 3, 6, 7, 9, 10, 11, 12, 15, 18, 19, 22, 23)]
colnames(data2) <- c("RECONIZE", "COURSES", "REASCH", "REGARDED", "THEORETICAL", "TECHNICAL", "APPLIED", "SOLVE PROBLEM", "MANAGEMENT", "PROSPECT", "JOB", "CURRUNT SITUATION", "CAREER")

X_imag = as.matrix(datas[, 1:8])
X_qutr = as.matrix(datas[, 13:16])
X_quaf = as.matrix(datas[, 9:11])
X_val = as.matrix(datas[, 17:20])
X_sat = as.matrix(datas[, 21:24])

# do the PCA of each block
pc.cr1 <- PCA(X_imag)
pc.cr1$var$cor

pc.cr2 <- PCA(X_qutr)
pc.cr2$var$cor

pc.cr3 <- PCA(X_quaf)
pc.cr3$var$cor

pc.cr4 <- PCA(X_val)
pc.cr4$var$cor

pc.cr5 <- PCA(X_sat)
pc.cr5$var$cor
View(rbind(pc.cr1$var$cor[,1:3],pc.cr2$var$cor[,1:3],pc.cr3$var$cor[,1:3],pc.cr4$var$cor[,1:3],pc.cr5$var$cor[,1:3]))

#####REPU,PROF,MANAG,TECH,THEO#####inner model
library(plspm)

# new model with the construct of a paper
ima <- c(0,0,0,0,0)
quatr<- c(0,0,0,0,0)
quaf <- c(0,0,0,0,0)
val <- c(1,1,1,0,0)
sat <- c(1,1,1,1,0)

inner.models <- rbind(ima,quatr,quaf,val,sat)
colnames(inner.models) <- rownames(inner.models)

outer.models <- list(1:8,13:16,9:11,17:20,21:24)
datas <- scale(data)
modes.A <- c("A","A","A","A","A")
modes.B <- c("B","B","B","B","B")

pls.mobile2 <- plspm(datas, inner.models, outer.models, modes.A, scaled=FALSE)
pls.mobileb <- plspm(datas, inner.models, outer.models, modes.B, scaled=FALSE)

# same method
summary(pls.mobile2)
summary(pls.mobileb)
View(pls.mobile2$inner_summary)

library(ggplot2)

# barchart of loadings
ggplot(data = pls.mobile2$outer_model,
      aes(x = name, y = loading, fill = block)) +
geom_bar(stat = "identity"
      , position = "dodge"
    ) +
# threshold line (to peek acceptable loadings above 0.7)
geom_hline(yintercept = 0.7, color = "gray50")
) +
# add title
ggtitle("Barchart of Loadings") +
# rotate x-axis names
theme(axis.text.x = element_text(angle = 90))

pls.mobile2$inner_summary
pls.mobile2$path_coefs
pls.mobile2$effects

good_rows=c(3:4,6:10)
path_effs <- as.matrix(pls.mobile2$effects[good_rows, 2:3])
rownames(path_effs) <- pls.mobile2$effects[good_rows, 1]
path_effs

op = par(mar = c(8, 3, 1, 0.8))
# barplots of total effects (direct + indirect)
barplot(t(path_effs), border = NA, col = c("#9E9AC8", 
 ="#DADAEB"),
  las = 2, cex.names = 0.8, cex.axis = 0.8,
  legend = c("Direct", "Indirect"),
  args.legend = list(x = "top", ncol = 2, border = NA,
    bty = "n", title = "Effects"))
# resetting default margins
par(op)

innerplot(pls.mobile2)
outerplot(pls.mobile2)
outerplot(pls.mobile2, what = "loadings")
outerplot(pls.mobile2, what = "weights")

innerplot(pls.mobileb)

###
##GSCA in matrixpls
###

datas2<-datas[,c(12)]
W0<-
matrix(c(rep(1,8),rep(0,26),rep(1,4),rep(0,16),rep(1,3),rep(0,27),rep(1,4),rep(0,23),rep(1,4)),nrow=23,ncol=5)
GSCA.res <- GSCA(as.data.frame(datas2),W0, inner.models, estim=TRUE, path.test=FALSE, latent.names = rownames(inner.models))
#res<-GSCAestim(data=as.data.frame(datas2),W0,inner.models)

formative<-matrix(0,5,23, dimnames = list(colnames(inner.models), colnames(datas2)))

inner <- inner.models

reflective <-
matrix(c(rep(1,8),rep(0,26),rep(1,4),rep(0,16),rep(1,3),rep(0,27),rep(1,4),rep(0,23),rep(1,4)),nrow=23,ncol=5, dimnames = list(colnames(datas2),colnames(inner.models)))

#inner.GSCA(cov(datas2), t(W0), inner)

# Estimate using alternating least squares
matrixpls.res2 <- matrixpls(cov(as.data.frame(datas2)), model=list(inner = inner, reflective = reflective, formative = formative), outerEstimators = outer.GSCA, innerEstimator = inner.GSCA)

#latent variable in mode A
W <- t(attr(matrixpls.res2, "W"))
result2.imag <- scale(X_imag %*% W[1:8,1])
res2.quutr<-scale(X_quutr %*% W[12:15,2])
res2.quaf<-scale(X_quaf %*% W[9:11,3])
res2.val<-scale(X_val %*% W[16:19,4])
res2.sat<-scale(X_sat %*% W[20:23,5])
gsca.score.a<-cbind(result2.imag,res2.quutr,res2.quaf,res2.val,res2.sat)
colnames(gsca.score.a)<-c("ima", "quutr", "quaf", "val", "sat")

cor(gsca.score.a,pls.mobile2$scores)

#latent variable in mode B
Wb <- t(attr(matrixpls.res.ref, "W"))
result1.imag <- scale(X_imag %*% Wb[1:8,1])
res1.quutr<-scale(X_quutr %*% Wb[12:15,2])
res1.quaf<-scale(X_quaf %*% Wb[9:11,3])
res1.val <- scale(X_val %*% Wb[16:19,4])
res1.sat <- scale(X_sat %*% Wb[20:23,5])
gsca.score.b <- cbind(result1.imag, res1.quutr, res1.quaf, res1.val, res1.sat)

colnames(gsca.score.b) <- c("ima", "quutr", "quaf", "val", "sat")
cor(gsca.score.b, pls.mobileb$scores)

summary(matrixpls.res2)
effects(matrixpls.res2)

# inner model same as plsmp

#### mode b

formative2 <- t(reflective)
reflective2 <- matrix(0, nrow=23, ncol=5, dimnames =
list(colnames(datas2), colnames(inner.models)))

matrixpls.res.ref <- matrixpls(cov(as.data.frame(datas2)), model=list(inner =
inner,
    reflective = reflective2,
    formative = formative2),
    outerEstimators = outer.GSCA,
    innerEstimator = inner.GSCA)

summary(matrixpls.res.ref)
effects(matrixpls.res.ref)

#RGCCA

library(Matrix)
datas2 <- as.data.frame(datas)
attach(datas2)

X_imag = as.matrix(datas[,1:8])
X_quutr = as.matrix(datas[,13:16])
X_quaf = as.matrix(datas[,9:11])
X_val = as.matrix(datas[,17:20])
X_sat = as.matrix(datas[,21:24])

A = list(X_imag, X_quutr, X_quaf, X_val, X_sat)
# Define the design matrix (output = C)
\[
C = \text{matrix}(c(0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1), 5, 5)
\]

#C1 = \text{matrix}(c(0, 0, 1, 1, 0, 0, 1, 1, 1, 0), 3, 3)

# mode a
result.rgcca = \text{rgcca}(A, C, \tau = \text{rep}(1, 5), \text{scheme} = \text{"centroid"}, \text{scale} = \text{TRUE})

# mode b
result.rgccak = \text{rgcca}(A, C, \tau = \text{rep}(0, 5), \text{scheme} = \text{"centroid"}, \text{scale} = \text{TRUE})

# optim \tau
result.rgccat = \text{rgcca}(A, C, \tau = \text{optim}, \text{scheme} = \text{"centroid"}, \text{scale} = \text{TRUE})

# latent variable in mode A
result.rgcca$Y[[1]]

rgcca.score.a <- cbind(result.rgcca$Y[[1]], result.rgcca$Y[[2]], result.rgcca$Y[[3]], result.rgcca$Y[[4]], result.rgcca$Y[[5]])
colnames(rgcca.score.a) <- c("ima", "qutr", "quaf", "val", "sat")

Wr1 <- t(attr(matrixpls.res.rgcca, "W"))
resr1.imag <- \text{scale}(X.imag \%*% \text{Wr1}[1:8, 1])
resr1.qutr <- \text{scale}(X.qutr \%*% \text{Wr1}[12:15, 2])
resr1.quaf <- \text{scale}(X.quaf \%*% \text{Wr1}[9:11, 3])
resr1.val <- \text{scale}(X.val \%*% \text{Wr1}[16:19, 4])
resr1.sat <- \text{scale}(X.sat \%*% \text{Wr1}[20:23, 5])

# latent variable in mode B
result.rgccak

rgcca.score.b <- cbind(result.rgccak$Y[[1]], result.rgccak$Y[[2]], result.rgccak$Y[[3]], result.rgccak$Y[[4]], result.rgccak$Y[[5]])
colnames(rgcca.score.b) <- c("ima", "qutr", "quaf", "val", "sat")

Wr2 <- t(attr(matrixpls.res.rgcca1, "W"))
resr2.imag <- \text{scale}(X.imag \%*% \text{Wr2}[1:8, 1])
resr2.qutr <- \text{scale}(X.qutr \%*% \text{Wr2}[12:15, 2])
resr2.quaf <- \text{scale}(X.quaf \%*% \text{Wr2}[9:11, 3])
resr2.val <- \text{scale}(X.val \%*% \text{Wr2}[16:19, 4])
resr2.sat <- \text{scale}(X.sat \%*% \text{Wr2}[20:23, 5])

(cor(X.qutr \%*% (as.matrix(result.rgcca$a[[2]])), X.val \%*% (result.rgcca$a[[4]])))
(cor(X.qutr \%*% (as.matrix(result.rgcca$a[[2]])), X.sat \%*% (result.rgcca$a[[5]])))
(cor(X.imag \%*% (as.matrix(result.rgcca$a[[1]])), X.val \%*% (result.rgcca$a[[4]])))
# 0.5050689
(cor(X.imag \%*% (as.matrix(result.rgcca$a[[1]])), X.sat \%*% (result.rgcca$a[[5]])))
#0.786386
(cor(X_quaf%*%(as.matrix(result.rgcca$a[[3]])),X_val%*%(result.rgcca$a[[4]])))
(cor(X_quaf%*%(as.matrix(result.rgcca$a[[3]])),X_sat%*%(result.rgcca$a[[5]])))
(cor(X_val%*%(as.matrix(result.rgcca$a[[4]])),X_sat%*%(result.rgcca$a[[5]])))

#####factorial function to compute the relation between blocks
(cov(result.rgcca$Y[[1]],result.rgcca$Y[[4]])) #%*% (result.rgcca$Y[[1]])
#0.5085283
(cov(result.rgcca$Y[[1]],result.rgcca$Y[[5]]))
#0.7917722
(cov(result.rgcca$Y[[2]],result.rgcca$Y[[4]]))
(cov(result.rgcca$Y[[2]],result.rgcca$Y[[5]]))
(cov(result.rgcca$Y[[3]],result.rgcca$Y[[4]]))
(cov(result.rgcca$Y[[3]],result.rgcca$Y[[5]]))
(cov(result.rgcca$Y[[4]],result.rgcca$Y[[5]]))

#Computing of path coefficients (OLS)
#(relations between intangibles)
#same as the a computation fumular
((t(result.rgcca$Y[[1]])%*%result.rgcca$Y[[1]])^(-1))*(result.rgcca$Y[[1]][,1]%*%result.rgcca$Y[[4]][,1])
#0.5050689
((t(result.rgcca$Y[[1]])%*%result.rgcca$Y[[1]])^(-1))*(result.rgcca$Y[[1]][,1]%*%result.rgcca$Y[[5]][,1])
#0.786386
((t(result.rgcca$Y[[2]])%*%result.rgcca$Y[[2]])^(-1))*(result.rgcca$Y[[2]][,1]%*%result.rgcca$Y[[4]][,1])
((t(result.rgcca$Y[[2]])%*%result.rgcca$Y[[2]])^(-1))*(result.rgcca$Y[[2]][,1]%*%result.rgcca$Y[[5]][,1])
((t(result.rgcca$Y[[3]])%*%result.rgcca$Y[[3]])^(-1))*(result.rgcca$Y[[3]][,1]%*%result.rgcca$Y[[5]][,1])

#loading
#relations between observed variables and their intangible
cor(result.rgcca$Y[[1]],datas[,1])
#0.8917952
cor(result.rgcca$Y[[1]],datas[,2])
#0.6289619
cor(result.rgcca$Y[[1]],datas[,3])
#0.8283751
cor(result.rgcca$Y[[1]],datas[,4])
cor(result.rgcca$Y[[1]], datas[,5])  # 0.7679079
cor(result.rgcca$Y[[1]], datas[,6])  # 0.655003
cor(result.rgcca$Y[[1]], datas[,7])  # 0.6077648
cor(result.rgcca$Y[[1]], datas[,8])  # 0.6738276
# 0.655003
cor(result.rgcca$Y[[1]], datas[,9])  # 0.6077648
cor(result.rgcca$Y[[1]], datas[,10])  # 0.6738276

cor(result.rgcca$Y[[2]], datas[,13])  # 0.8949236
cor(result.rgcca$Y[[2]], datas[,14])
# 0.7603772
cor(result.rgcca$Y[[2]], datas[,15])  # 0.3950113
cor(result.rgcca$Y[[2]], datas[,16])  # 0.7370211
cor(result.rgcca$Y[[2]], datas[,17])  # 0.8718107
cor(result.rgcca$Y[[2]], datas[,18])  # 0.7559191
cor(result.rgcca$Y[[2]], datas[,19])  # 0.5419517
cor(result.rgcca$Y[[2]], datas[,20])  # 0.7649028
cor(result.rgcca$Y[[2]], datas[,21])  # 0.9932738
cor(result.rgcca$Y[[2]], datas[,22])  # 0.4889039
cor(result.rgcca$Y[[2]], datas[,23])
cor(result.rgcca$Y[[5]], datas[,24])
#0.3292631

tau = rep(1,5)
W.rgcca <- as.matrix(bdiag(lapply(result.rgcca$a,
          function(x){matrix(x, nrow=1)})))

W.mod <- (W.rgcca != 0) * 1

S <- cov(do.call(cbind, A))
W.matrixpls <- weight.pls(S, list(inner = C,
                                reflective = t(W.mod),
                                formative = matrix(0, nrow(W.mod), ncol(W.mod))),
                                W.mod = W.mod,
                                innerEstimator = inner.centroid,
                                outerEstimators = outer.RGCCA, tau = 0)

# get the effect
effects(modeA)

model <- list(inner = inner,
              reflective = reflective,
              formative = formative)

fixed <- matrixpls(S, model, weightFunction = weight.fixed)
optimR2 <- matrixpls(S, model, weightFunction = weight.optim)
modeA <- matrixpls(S, model, outerEstimators = outer.RGCCA)

modeB <- matrixpls(S, model, outerEstimators = outer.modeB)

rbind(ModeA = r2(modeA),
       ModeB = r2(modeB),
       Fixed = r2(fixed))
View(W.rgcca)
print(W.matrixpls)

cor(cbind(do.call(cbind, result.rgcca$Y),
           do.call(cbind, A) %*% t(W.matrixpls)))
model.sem <-
  # measurement model
  Image =~ ima1 + ima2 + ima3 + ima4 + ima5 + ima6 + ima7 + ima8
  Quaf =~ quaf1 + quaf2 + quaf3
  Qutr =~ qutr2 + qutr3 + qutr4 + qutr5
  Val =~ val1 + val2 + val3 + val4
  Sat =~ sat1 + sat2 + sat3 + sat4

  # structural
  Val ~ Sat
  Image ~ Sat
  Quaf ~ Sat
  Qutr ~ Sat
  Image ~ Val
  Quaf ~ Val
  Qutr ~ Val

fitFR <- sem(model.sem, data = datas2, std.lv = TRUE)
standardizedSolution(fitFR)$est.std[7]

# PCA
pca_imag <- princomp(scale(X_imag))
plot(pca_imag$eig$, type = "l")  
# 2 components
plot(pca_imag, ncomp = 2)

pca_sat <- PCA(scale(X_sat))
pca_sat$var$cor
plot(pca_sat$eig$eigenvalue, type = "l", main = "eigenvalue of the sat")

pca_quaf <- PCA(scale(X_quaf))
pca_quaf$var$cor
plot(pca_quaf$eig$eigenvalue, type = "l", main = "eigenvalue of the quaf")

pca_qutr <- PCA(scale(X_qutr))
pca_qutr$var$cor
plot(pca_qutr$eig$eigenvalue, type = "l", main = "eigenvalue of the qutr")

pca_val <- PCA(scale(X_val))
pca_val$var$cor
plot(pca_val$eig$eigenvalue, type = "l", main = "eigenvalue of the val"