The SMH Algorithm : An Heuristic for Structural Matrix Computation in the Partial Least Square Path Modeling

Odilon Yapo M. Achiepo[1]
Agropastoral Management School
UPGC University of Korhogo
BP 1328 Korhogo, Cote d'Ivoire

Edoete Patrice Mensah[2]
Dpt. of Maths. and Computer Science
INPHB Institut of Yamoussoukro
BP 1093 Yamoussoukro, Cote d'Ivoire

Edi Kouassi Hilaire[3]
Lab. of Maths. and Computer Science
UNA University of Abidjan
02 BP 801 Abidjan 02, Cote d'Ivoire

***Abstract*—The Structural equations modeling with latent's variables (SEMLV) are a class of statistical methods for modeling the relationships between unobservable concepts called latent variables. In this type of model, each latent variable is described by a number of observable variables called manifest variables. The most used version of this category of statistical methods is the partial least square path modeling (PLS Path Modeling). In PLS Path Modeling, the specification of the relashonships between the unobservable concepts, knows as structural relationships, is the most important thing to know for practical purposes. In general, this specification is obtained manually using a lower triangular binary matrix. To obtain this lower triangular matrix, the modeler must put the latent variables in a very precise order, otherwise the matrix obtained will not be triangular inferior. Indeed, the construction of such a matrix only reflects the links of cause and effect between the latent variables. Thus, with each ordering of the latent variables corresponds a precise matrix.The real problem is that, the more the number of studied concepts increases, the more the search for a good order in which it is necessary to put the latent variables to obtain a lower triangular matrix becomes more and more tedious. For five concepts, the modeler must test $5! = 120$ possibilities. However, in practice, it is easy to study more than ten variables, so that the manual search for an adequate order to obtain a lower triangular matrix extremely difficult work for the modeler. In this article, we propose an heuristic way to make possible an automatic computation of the structural matrix in order to avoid the usual manual specifications and related subsequent errors.**

***Keywords*—Structural equations modeling; PLS algorithm; latents variables; structural matrix; R programming language**

I. Introduction: PLS Path Modeling in R

The PLS Path Modeling in a structural equation modelling with latent variables (SEMLV), is a method in which the partial least square (PLS) algorithm is used to estimate the model ([1], [2], [3]). Generally, the structural equation models (SEM) are describe graphically by specifying the latent variables (inobservable). For each latent variable, the manifest variables (observable) that are related to it are also specified. Latent variables represent concepts such as loyalty, quality, poverty, abilities, etc. The manifest variables are indicators that describe these latent variables and they are collected in a dataset. An example of such model, called European Customer Satisfaction Index (ECSI) Model, that can be found in [4], is giving in the Figure 1 below:

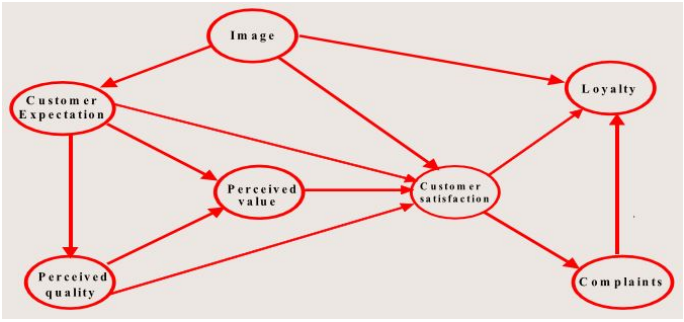

Fig. 1. The European Customer Satisfaction Index Model.

The Figure 1 shows an example of the structural relationship between latent variables. It is known as the European Customer Satisfaction Index model (ECSI model) and is often used in marketing studies. This article focuses on the specification of this kind of relations in practice. When we use a computer to estimate the model, the graph is often specified as binary low triangular matrix. The operation may be time-consuming because one has to find the best order of the latent variables in a table in order to get the lower triangular matrix. The goal of this paper is to give a method which automatically get the right order and automatically compute the structural relationship matrix.

II. Conceptualization : Main idea behind the SMH Algorithm

A square (lower triangular) boolean matrix representing the inner model (i.e. the path relationships between latent variables) is a matrix of zeros and ones that indicates the structural relationshipsbetween latent variables. This path matrix must be a lower triangular matrix that has a 1 when column j affects row i, and a 0 otherwise.

The latent variables can be classified in three categories according to their roles in the structural equations in which they appear. The SMH is based on the following classification:

- **The exogenous variables :** It's the latent variables which have no other latent variables related to them.
- **The endogenous variables :** It's the latent variables which are not related to any other latent variables.

- **The neutral variables :** It's the latent variables that are related to others in both directions.

The main idea of this heuristic is to classify all the latent variables within these differents groups (Exogenous, Endogenous, Neutral) and find a way to order them to obtain a lower triangular matrix. To find the rigth order of the latents variables, we can remark than the exogenous latents variables must be ordered first (left side), then the neutral latents variables must follow them (middle), and finally, the endogenous latent variables must be the last ones to use (right side). This groups order is found by analyzing some simple cases.

For a formal purpose, let consider the following mathmatical notations :

- N the numbers of latent variables
- ξ_j the j^{th} latent variable
- Θ_j the endogenous statut of the latent variable ξ_j
- Γ_j the exogenous statut of the latent variable ξ_j
- E_j the number of latents variables that the variable ξ_j is related to
- F_j the number of latents variables that are related to ξ_j
- K_j the numbers of exogenous, latent variables that are related to the variable ξ_j
- μ_j the order score of the latent variable ξ_j

The variables Θ_j and Γ_j can be express using the kroneker notation :

$$\Theta_j = \begin{cases} 1 & \text{if } \xi_j \text{ is endogenous} \\ 0 & \text{if } \xi_j \text{ is not endogenus} \end{cases} \quad (1)$$

$$\Gamma_j = \begin{cases} 1 & \text{if } \xi_j \text{ is exogenous} \\ 0 & \text{if } \xi_j \text{ is not exogenous} \end{cases} \quad (2)$$

This conceptualisation, will be use to find an ordered metric for each variable. The variables will be orderd according to the value of this metric. Hight the metric's value of a variable is, hight will be it rank.

III. Computing : The order metric of the SMH Algorithm

The heuristic method is based on three general empirical principles where it foundation can be seen.

A. About the Exogenous Variables

The exogenous latent variables are the only ones with $\Gamma_j = 1$ and they must have the lowest values μ_j to be in the first position in the structural matrix. Different exogenous latent variables are distinguished according to the number of latent variables F_j they are related to. The higher F_j is, the lower the score μ_j has to be. Some variables that an exogenous latent variable is related to can be endogenous. Therefore, exogenous variables are to be characterized by the number of endogenous variables they belongs to (K_j) they are related to. The higher K_j is, the higher the score μ_j has to be. To take into account these realities, the order score of the endogenous latent variables is taken to be $-10^4 F_j + K_j$. In this case, the minimum score is obtained when all latent variables are exogenous except for one which is exogeneous $(F_j = N-1, K_j = 1)$ and the maximum score is obtained when all the latent variables are endogenous except for one which is endogenous $(F_j = 1, K_j = N-1)$. The scores of the exogenous latent variables are in the interval $[-10^4(N-1)+1, -10^4+N-1]$.

B. About the Endogenous Variables

The endogenous latent variables are the only ones with $\Theta_j = 1$ and they must have the highest values of μ_j to be in the last position in structural matrix. Different endogenous latent variables are distinguished according to the number of latent variables (E_j) related to them. The higher E_j is, the higher the score μ_j must be. To take into account this reality, the order score of the endogenous latent variables is taken to be $10^4 E_j$. In this case, the maximum score is obtained when all latent variables are endogenous except for one $(E_j = N-1)$ which is exogenous and the minimum score is obtained when all the latent variables are exogenous except for one $(E_j = 1)$ which is endogenous. The scores of the exogenous latent variables are in the interval $[10^4, -10^4(N-1)]$.

C. About the Neutral Latent Variables

The neutral latent variables are the ones with the $\Theta_j + \Gamma_j = 0$. They must have the values of μ_j which are higher than the highest exogenous variable value and less than the lowest endogenous variable value in order to be between exogenous and endogenous latent variables in the structural matrix. Different neutral latent variables are distinguished according to the number of latent variables(F_j) they are related to. The higher F_j is, the lower the score μ_j must be. Some variables that a neutral latent variable are related to can be endogenous. Therefore, exogenous variables are to be characterized by the number of neutral variables(K_j) they are related to. The higher K_j is, the higher the score μ_j have to be. Neutral variables are also distinguished according to the number of latent variables(E_j) they are related to. The higher E_j is, the higher the score μ_j have to be. To take into account all these realities, the order score of the endogenous latent variables is taken to be $10^{3/2} E_j - 10 F_j + K_j$. In this case, the maximum score is obtained when all latent variables are endogenous except for one $(E_j = N-1, F_j = 1, K_j = 1)$ which is exogenous and the minimum score is obtained when all latent variables are exogenous except for $(E_j = 1, F_j = N-1, K_j = N-1)$ which is endogenous. The scores of the exogenous latent variables are in the interval $[10^{3/2}(N-1) - 9, 10^{3/2} - 9(N-1)]$.

D. Order Score Computation

To compute the structural matrix, the latent variables must be ordered properly. The correct order give a lower triangular matrix. As it has been said before the main objective of the heuristic is to find the best set of ordered variables to compute

the correct structural matrix. This order is based on the score that can be defined by

$$\mu_j = \begin{cases} 10^4 E_j & \text{if } \xi_j \text{ is endogenous} \\ 10^{3/2} E_j - 10 F_j + K_j & \text{if } \xi_j \text{ is neutral} \\ -10^4 F_j + K_j & \text{if } \xi_j \text{ is exogenous} \end{cases} \quad (3)$$

Mathematically, these descriptions can be summarize in the single function defined as :

$$\mu_j = 10^4 E_j \Theta_j + (10^{3/2} E_j - 10 F_j + K_j) * (1 - \Theta_j - \Gamma_j) - (10^4 F_j - K_j)\Gamma_j \quad (4)$$

The latent variables are then ordered based on their μ scores. For two latent variables ξ_i and ξ_j , the position of ξ_i in the structural matrix is before ξ_j if $\mu_i \leqslant \mu_j$.

The problem solved by our method is a similar problem to that of the well-known traveling salesman problem in operations research ([5], [6]). However, the metaheuristics used in operational research, such as tabu search, simulated annealing, genetic algorithms, etc. have the disadvantage of requiring significant resources in terms of calculation. In addition, the implementation of these algorithms is very complex and require a good mastery of their operating principles. Compared to these methods, the approach developed in this article is very easy to use. The method is limited to a simple classification of latent variables and manifests variables, to their enumeration and to the application of a simple arithmetic formula to obtain scores for ordering latent variables. The computation time is more than one hundred lower than that of conventional optimization metaheuristics. Our approach is therefore an optimization metaheuristic that applies to a very particular problem, namely, the search for a structural matrix in the PLS Path Modeling.This heuristic is the core method of used in the R package *plspm.formula* ([7]) we have already developed and which is available for free download on the mirror sites of the R software. The following Figure 2 shows the performance of the heuristic when the number of latent variables is growing :

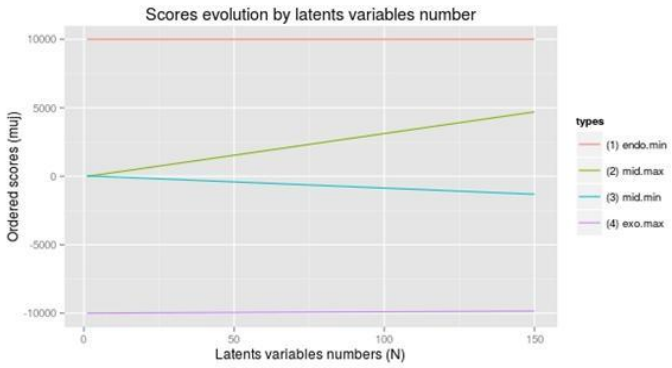

Fig. 2. The exogeneous minimum and maximum

According to the figure, the heuristic is able to give correct response with more than 100 latent variables.Based on this result, we can state that the heuristic method is very robust since the reasonable numbers of latent variables one can use in practice is generally less than twenty.

IV. Programming : The SMH algorithm code in R

A. The plspm.shm R Function

This section present the implementation of the SMH algorithm in R language [8]. The fonction is based on the R Package *plsmp* ([9]) basis of this scientific computing language can be found in . The SMH algorithm in R is as follows:

```
require(plspm)
plspm.shm <- function(latents,latlist,
                      mat=TRUE,iplot=TRUE)
{
   N <- length(latents)
   vldroite <- unique(unlist(latlist[[2]]))
   vlgauche <- latlist[[1]]
   calc.nfois.exo <- function(vlat) {
      nbfois.exo <- function(vtot) {
         return(sum(vlat %in% vtot))
      }
      return(sum(sapply(latlist[[2]],
             nbfois.exo)))
  }
  calc.nfois.exo <- Vectorize(calc.nfois.exo)
  vlexo <- latents[1-as.numeric(
                   latents %in% vldroite)]
  calc.equ.exo <- function(vlat){
     indx <- which(latlist[[1]] == vlat)
     if(length(indx) < 1){res <- 0}
     else {
        res <- sum(as.numeric(
        vlexo %in% latlist[[2]][[indx]]))
     }
     return(res)
  }
  calc.equ.exo <- Vectorize(calc.equ.exo)
  ntotF <- sapply(latlist[[2]], length)
  calc.nbequ <- function(vlat){
     indx <- which(latlist[[1]] == vlat)
     if (length(indx) < 1) {res <- 0}
     else {res <- ntotF[indx]}
     return(res)
  }
  calc.nbequ <- Vectorize(calc.nbequ)
  Thetaj <- 1-as.numeric(latents %in% vldroite)
  Ej <- as.vector(calc.nbequ(latents))
  Gammaj <- 1-as.numeric(latents %in% vlgauche)
  Fj <- as.vector(calc.nfois.exo(latents))
  Kj <- as.vector(calc.equ.exo(latents))
  muj <- 10^4*Ej*Thetaj+(10^(3/2))*Ej-10*Fj+Kj)
         *(1-Thetaj-Gammaj)-(10^4*Fj-Kj)*Gammaj
  olatents <- latents[order(muj)]
  reslist <- list(mu = muj, ordre = olatents)
  if(mat){
    matlist <- function(vect){
       return(as.numeric(olatents %in% vect))
    }
    Mlist <- lapply(latlist[[2]], matlist)
    mat.vect <- function(j){
        indj <- which(
                  latlist[[1]] == olatents[j])
        if (length(indj) < 1){
           return(rep(0, N))
        }
        else {return(unlist(Mlist[indj]))}
   }
   mat.vect <- Vectorize(mat.vect)
```

```
   Mat <- t(mat.vect(1:N))
   rownames(Mat) <- olatents
   reslist <- c(reslist, list(matrice = Mat))
 }
 if (iplot) {innerplot(Mat)}
 return(reslist)
}
```

B. The Parameters and Results of the plspm.shm Function

The algorithm take essentially two inputs:

latent : a character vector containing the latent variable names
latlist : a list to specify which latents variables explain another

The parameter latlist is a R list structure and must contain two R objects:

1) a vector of the endogenous latent variables.
2) a list of vector objects for each endogenous variable. For an endogenous variable, the vector contains exogenous latent variables which are related to it. The order of vector objects in the internal list must correspond to the one of the endogenous variable.

The main output of the plspm.shm function is an ordered vector of all the latent variables. This order is the one one can use to have a structural matrix in the form of lower trianguler binary matrix needed to estimate PLS Path Model, for example the *plspm()* function in the *plspm* R package (*plspm*). But, the functions have the logical parameter mat that permits to compute the corresponding inner matrix *(mat=TRUE)* or not *(mat=FALSE)*. This prevents from using a manual ordered latent variables vector to find the matrix. By default, the function compute that matrix. The function also have an other logical parameter name igraph that specifies if the relationship graph must be compute *(igraph=TRUE)* or not *(igraph=FALSE)*.

V. Illustration: Test of the SMH Algorithm in R

A. Applications on a Relative Simple Problem

To show the simple usage of SMH algorithm, we generate four (4) latent variables "A1", "A2", "A3" and "A4". Weassume that the relations between these latent variables can be described by two rules :

- First: "A1" and "A4" have an impact on "A2"
- Secondly: "A3" have an impact on "A1" and "A2".

The implementation in R is giving by the code below :

```
R> lvect <- paste("A",1:4,sep="")

R> lvlist <- list(
paste("A",1:3,sep=""),
list("A3", c("A1","A3","A4"),"A4")
)

R> res <- plspm.shm(lvect,lvlist,
                    mat=TRUE,iplot=TRUE)
```

The different results obtained in R concerning the latent variables vector, the latent variables list and the structural matrix are :

```
R> print(lvect)
[1] "A1" "A2" "A3" "A4"

R > print(lvlist)
[[1]]
[1] "A1" "A2" "A3"
[[2]]
[[2]][[1]]
[1] "A3"
[[2]][[2]]
[1] "A1" "A3" "A4"
[[2]][[3]]
[1] "A4"

R> print(round(res,2))
$mu
[1]  21.63 30000.00 11.62 -20000.00
$ordre
[1] "A4" "A3" "A1" "A2"
$matrice
    [,1] [,2] [,3] [,4]
A4    0    0    0    0
A3    1    0    0    0
A1    0    1    0    0
A2    1    1    1    0
```

We can then see that the algorithm is capable of finding the correct order of the latent variables and capable of giving the correct structural matrix (triangular inferior). The graph Figure 3 given by the algorithm is :

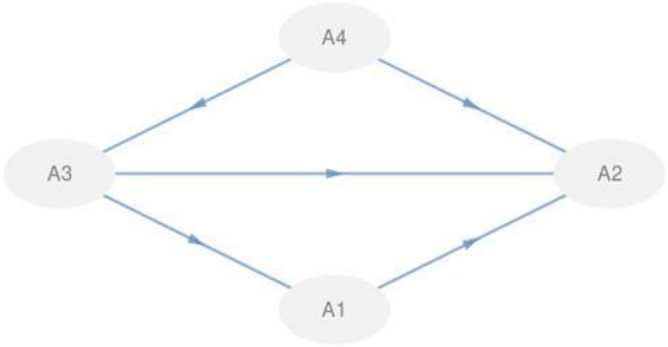

Fig. 3. The inner graph of the simple example

This graph is the graphical version of the structural matrix and it use makes easier the understanding of the structural relationships. Notice that in this example, we have four latent variables. The next example will use six latent variables and is concerned with a real example of the ECSI model as presented in the plspm package on the satisfaction dataset.

B. Application on a More Complex Problem

In this second example, the latent variables are denoted : image ("IMAG"), expectations ("EXPE"), quality ("QUAL"), value ("VAL"), satisfaction ("SAT") and loyalty ("LOY"). The

relations between the latent variables are more complex and can be described by the following five rules :

- Image have an influence on expectations, satisfaction and loyalty
- Expectation have an influence on quality, value and satisfaction
- Quality have an influence on value and satisfaction
- Value have influence on satisfaction
- Satisfaction have influence on loyalty.

The implementation of these different rules and the application of the RSH algorithm in R are given by the following code :

```
R> satvect <- c("IMAG", "EXPE", "QUAL",
                "VAL", "SAT", "LOY")

R> satlist <- list(
               c("EXPE","QUAL", "VAL","SAT", "LOY"
               list(
                 c("IMAG"),
                 c("EXPE"),
                 c("EXPE","QUAL"),
                 c("IMAG", "EXPE", "QUAL", "VAL"),
                 c("IMAG", "SAT"))
               )

R> satres <- plspm.shm(satvect,satlist,
                       mat=TRUE,iplot=TRUE)
```

The different results obtained in R and concerning the latent variables vector, the latent variables list and the structural matrix are :

```
R> print(satvect)
[1] "IMAG" "EXPE" "QUAL" "VAL"  "SAT"  "LOY"

R> print(satlist)
[[1]]
[1] "EXPE" "QUAL" "VAL"  "SAT"  "LOY"
[[2]]
[[2]][[1]]
[1] "IMAG"
[[2]][[2]]
[1] "EXPE"
[[2]][[3]]
[1] "EXPE" "QUAL"
[[2]][[4]]
[1] "IMAG" "EXPE" "QUAL" "VAL"
[[2]][[5]]
[1] "IMAG" "SAT"


R> print(satres)
$mu
[1] -30000.0   2.6   11.6   53.2   117.5   20000.0
$ordre
[1] "IMAG" "EXPE" "QUAL" "VAL"  "SAT"  "LOY"
$matrice
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
IMAG	0	0	0	0	0	0
EXPE	1	0	0	0	0	0
QUAL	0	1	0	0	0	0
VAL	0	1	1	0	0	0
SAT	1	1	1	1	0	0
LOY	1	0	0	0	1	0

We can again see that the algorithm is capable of finding the correct order of the latent variables and capable of giving the correct structural matrix (triangular inferior). The graph Figure 4 given by the algorithm is :

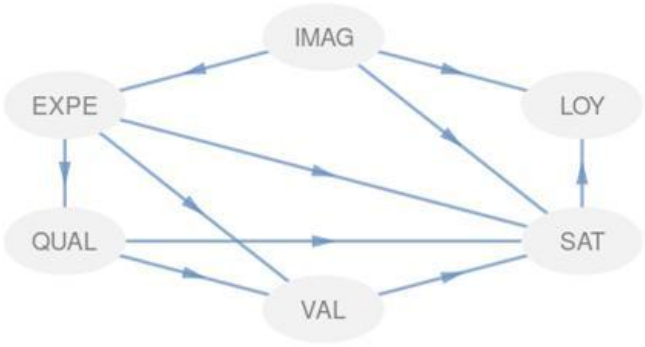

Fig. 4. The inner graph of the complex example

This graph is the graphical version of the structural matrix. It confirms the fact that the heuristic is able to handle problems with large variables.

VI. Conclusion

In the field of PLS Path modeling, the task of specifying structural matrices has always been tedious because of its purely manual nature. The method proposed in this article freed the modeler of this constraint by providing a means of automatic search of the correct order in which the latent variables must be placed in order to obtain a lower triangular matrix. The algorithm even calculates this matrix directly, which saves time and avoids errors related to the manual specification of such matrices. The heuristic described in this paper makes easier the process of finding automatically the PLS Path Modeling specifications. The simulations carried out show that, theoretically, this heuristic can easily be used for models involving more than one hundred latent variables. This possibility increases the scope of the PLS Path Modeling that was, until now, used on a limited number of latent variables because of the difficulties related to the manual specification of the structural relationships. However, one must take care of the fact that the structural relation rules are not circular because the matrix, in this case, is not triangular and that the problem can be misspecified in practice. The SMH heuristic also avoids the need of exploring all of the possible ordered latent variables configurations. It is an elegant solution to this combinatory problem. The use of this heuristic avoids the test of all arrangements of latent variables in order to find the best which gives the correct structural matrix. Future work will focus on the generalization of the principle of our method on the traveling salesman problem. Such a generalization will allow the algorithm to apply a much larger set of problems.

In this case, the study of the algorithmic complexity of the method and its comparison with the existing heuristics will make it possible to better understand its advantages over the optimization metaheuristics known to deal with the traveling salesman problem.

REFERENCES

[1] Avkiran, N. K., Ringle, C. M., Low, R. K. Y. (2018); *Monitoring Transmission of Systemic Risk: Application of Partial Least Squares Structural Equation Modeling in Financial Stress Testing*. Journal of Risk, forthcoming, 2018.

[2] Ahrholdt, D. C., Gudergan, S. P., Ringle, C. M.; *Enhancing Service Loyalty: The Roles of Delight, Satisfaction, and Service Quality*. Journal of Travel Research, Volume 56, Issue 4, pp. 436-450, 2017.

[3] Mikko Ronkk, Cameron N. McIntosh, John Antonakis, Jeffrey R. Edwards (2016); *Partial least squares path modeling: Time for some serious second thoughts*. Journal of Operations Management, Elsivier, 2016.

[4] D. Christian (2009); *Free Model for Generalized Path Modeling and Comparison with Bayesian Network*, EDF Research and Development, 2009.

[5] Ahmad Fouad El-Samak, Wesam Ashour (2015); *Optimization of Traveling Salesman Problem Using Affinity Propagation Clustering and Genetic Algorith*. JAISCR, Vol. 4, No. 4, pp. 239-245, 2015.

[6] Johann Dréo, Alain Pétrowski, Patrick Siarry, Eric Taillard (2003); *Métaheuristiques pour l'optimisation difficile*. Eyrolle, 2003.

[7] Odilon Yapo M.,Achiepo (2015); *plspm.formula: Formula Based PLS Path Modeling in R*, R package version 1.0., 2015.

[8] R Core Team (2015); *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, 2015.

[9] G. Sanchez and L., Trinchera and G. Russolillo (2015);*plspm: Tools for Partial Least Squares Path Modeling*,R package version 0.4.7, 2015.