How to use the ProcessPLS packages

Geert and Tim

September 23, 2021

Abstract

Tutorial on how to use Process PLS packages. Currently in R and almost available in Matlab.

Right now, the R package output does not comply with the terminology used in the paper. We are working hard to change this for the next version. In the meantime, please use this document to understand what the many matrices in the R output mean, and how to evaluate your model. Note that the R package `pathmodelr` with use the same terminology as the upcoming Matlab library.

1 ReadMe - how to read model output

The main function in pathmodelr, `process_PLS()`, outputs a listenvironment with many results. Currently the terminology in these lists is not straightforward. Here, we describe the current names, and what they will be, to ensure correct interpretation of the model parameters and results.

Using the provided Val de Loire analysis as working example (starting on the next page):

If you need any help, feel free to contact me at g.h.v.kollenburg@tue.nl or the old department at chemometrics@science.ru.nl

```
install.packages(c('caret', 'dplyr', 'ggplot2', 'listenv',
    'reshape2', 'R6', 'network', 'ggnetwork', 'diagram'))
install.packages("D:\\pathmodelr_0.12.tar.gz", repos= NULL, type="source")
library(pathmodelr)

#load the Val De Loir data
install.packages("FactoMineR")
library(FactoMineR)
data(wine)
wine_data <- wine[,c(-1,-2, -31)]

#model specification
inner_model_spec <- t(matrix(c(0,0,0,0,0,
                               1,0,0,0,0,
                               1,1,0,0,0,
                               1,1,1,0,0,
                               1,1,1,1,0),
                             nrow=5, ncol=5))

rownames(inner_model_spec) <- c("smell_at_rest", "view",
                                "smell_after_shaking", "tasting", "global_quality")
colnames(inner_model_spec) <- rownames(inner_model_spec)

outer_model_spec <- list(1:5, 6:8, 9:18, 19:27, 28)

# Non-parametric bootstrapping does not actually make much sense
# The data set is small small and contains entire population


model <- process_PLS(wine_data, inner_model_spec, outer_model_spec, bootstrap = FALSE)

#Main inner model result (v0.12)
model$path_variances_explained
#plot the model
innerplot(model$path_variances_explained, show.values =T)

#note: newwer builds of R draw the model parameters very lightly

#for v0.13:
# model$inner_model
# innerplot(model$inner_model, show.values =T)
```

1.1 main output

Elements of the Process PLS `model` object can be called using the $ sign (one can also use the auto-complete options in IDEs like RStudio).

1.2

With a few exceptions like `connection_matrix`, each list contains more lists of matrices, each with its own name. Some of these matrices are of specific interest to reseachers. In the next tables you may find the meaning of those sub-lists and what they will be named in v0.13 (as well as in the upcoming Matlab library).

And lastly, the output related to the outer model is renamed as :

current name (v0.12)	new name (v0.13)	description
`nodes`	`blocks`	information on the blocks
`loggers`	`loggers`	technical logs of the performed analyses
`connection_matrix`	`C`	The inner model specification
`block_names`	`block_names`	names of the blocks
`path_effects`	`LV_on_LV_effects`	In path modelling terms: direct, indirect, and total effects of each LV on LVs of other blocks
`path_variances_explained`	`inner_model`	Main model results, based on element in the row above, provided as R^2 values
`inner_effects`	`MV_with_other_blocks`	relations of manifest variables with LVs of or other blocks, somewhat like cross-loadings
`outer_effects`	`outer_model`	PLS loadings for each block.

Table 1: Names and meaning of elements in the the first level of the Process PLS `model` object

current name (v0.12)	new name (v0.13)	description
`effects_on_LV_per_block`	`MV_with_other_LV_ordered`	relations of variables with individual LVs of other blocks, with separate matrix per block (may be removed in 0.13)
`effects_on_LV`	`MV_with_other_LV`	relations of manifest variables with individual LVs of other blocks
`effects`	`MV_with_other_block`	relations of variables with an entire other blocks

Table 2: Names and meanings of elements in `model$MV_with_other_blocks` (v0.13) / `model$inner_effects` (v.0.12). Elements can be interpreted somewhat like cross-loadings.

current name (v0.12)	new name (v0.13)	description
`individual`	`all`	one list with direct, indirect, and total effects (may be removed in 0.13, as it is superfluous)
`direct`	`direct`	direct effects of LVs on LVs from subsequent blocks
`indirect`	`indirect`	indirect effects of each LV on LVs from subsequent blocks
`total`	`total`	total reproduced relation between each LV and LVs from subsequent blocks (i.e., direct + indirect).

Table 3: Names and meanings of elements in `model$LV_on_LV_effects` (v0.13) / `model$path_effects` (v.0.12). Elements are the main direct and indirect effects of the inner, structural, model

current name (v0.12)	new name (v0.13)	description
`outer_effects`	`variable_contribution`	The total contribution of a manifest variable to the variance of the LVS in that block
`outer_effects_on_LV`	`loadings`	the loadings, relating the manifest variables to each LV in that block

Table 4: Names and meanings of elements in `model$outer_model` (v0.13) / `model$outer_effects` (v.0.12). info on the outer model parameters.