Estimating a PLS-SEM

Model Estimation in SEMinR

Lilian Kojan

2021-04-27 (updated: 2021-07-12)

PLS model estimation in SEMinR

1. Before estimation
2. Estimating a model
3. The SEMinR model object
4. Troubleshooting

Before estimation

What you need to estimate a model:

- Data
- Measurement model
- Structural model

Before estimation - example

```
# load seminr library
library(seminr)
# define measurement model mm
mm <- constructs(
  # define mode A composite construct: long form
  composite(construct_name = "Reputation",
            item_names = multi_items("IMAG", 1:5),
            weights = correlation_weights),
  # define mode A composite construct: shorter form
  composite("Satisfaction",
            multi_items("CUSA", 1:3),
            mode_A),
  # define mode A composite construct: even shorter form
  composite("Loyalty",
            multi_items("CUSL", 1:3))
)
```

Before estimation - example

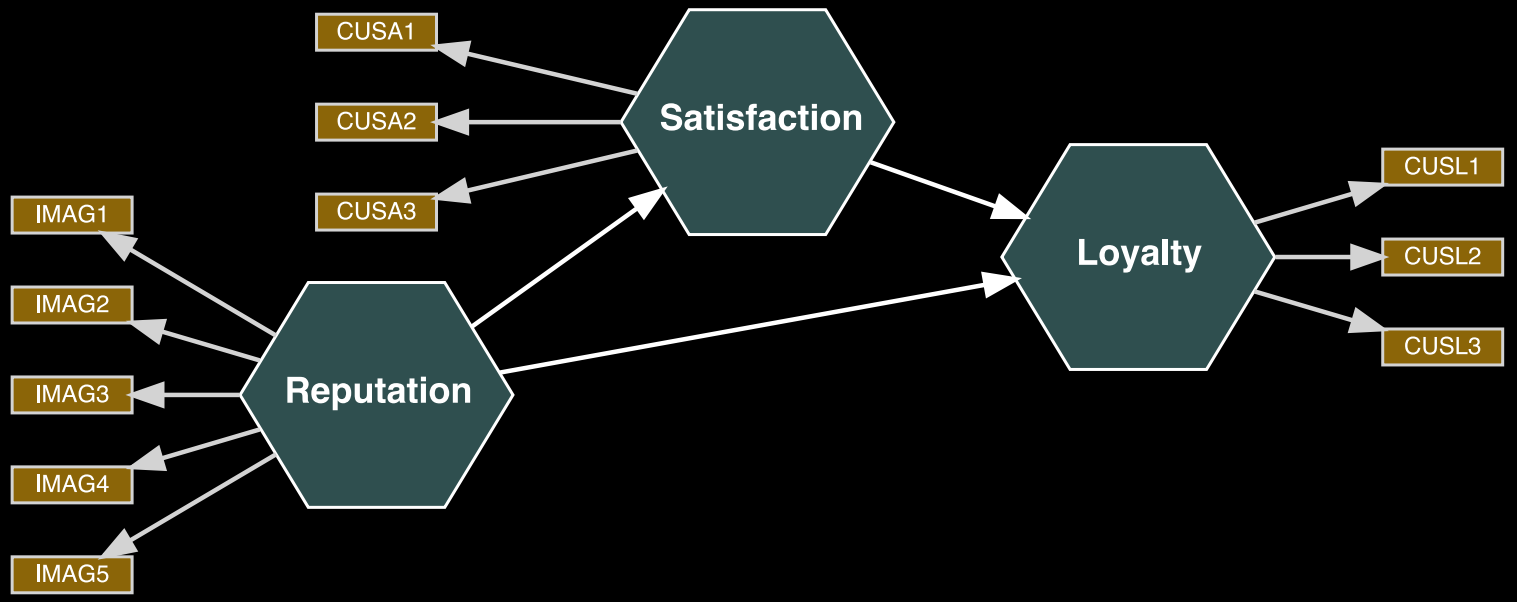
```
# define structural model
sm <- relationships(
  paths(
    from = "Reputation",
    to = c("Satisfaction", "Loyalty")
  ),
  paths(
    from = "Satisfaction",
    to = "Loyalty"
  )
)
```

Before estimation - example

```
# specify model without estimating it
specify_model(measurement_model = mm,
              structural_model = sm) %>%
  # plot model
  plot()
```

Before estimation - example

```
# specify model without estimating it
specify_model(measurement_model = mm,
              structural_model = sm) %>%
  # plot model
  plot()
```



Estimating a model

```
# estimate a pls model
model <- estimate_pls(
  data = mobi,
  measurement_model = mm,
  structural_model = sm,
  model = NULL,
  inner_weights = path_weighting,
  missing = mean_replacement,
  missing_value = NA
)
```

Estimating a model - the fundamentals

```
model <- estimate_pls(
  data = mobi,                  # data as data frame or matrix
  measurement_model = mm,    # measurement model
  structural_model = sm,     # structural model
  model = NULL,
  inner_weights = path_weighting,
  missing = mean_replacement,
  missing_value = NA
)
```

Estimating a model - the fundamentals

```
model <- estimate_pls(
  data = mobi,                  # data as data frame or matrix
  measurement_model = mm,    # measurement model
  structural_model = sm,     # structural model
  model = NULL,
  inner_weights = path_weighting,
  missing = mean_replacement,
  missing_value = NA
)
```

Equivalent code:

```
model <- estimate_pls(mobi, mm, sm)
```

Estimating a model - *model*

```
model <- estimate_pls(
  data = mobi,
  measurement_model = mm,
  structural_model = sm,
  # optional: model object generated by specify_model()
  # only necessary if mm and sm are not provided
  model = NULL,
  inner_weights = path_weighting,
  missing = mean_replacement,
  missing_value = NA
)
```

Estimating a model - *model*

```
model <- estimate_pls(
  data = mobi,
  measurement_model = mm,
  structural_model = sm,
  # optional: model object generated by specify_model()
  # only necessary if mm and sm are not provided
  model = NULL,
  inner_weights = path_weighting,
  missing = mean_replacement,
  missing_value = NA
)
```

```
specified_model <- specify_model(
  measurement_model = mm,
  structural_model = sm
  )
```

Estimating a model - *inner_weights*

```
model <- estimate_pls(
  data = mobi,
  measurement_model = mm,
  structural_model = sm,
  model = NULL,
  # optional: how paths are estimated
  # defaults to path weighting
  inner_weights = path_weighting,
  missing = mean_replacement,
  missing_value = NA
)
```

Estimating a model - *inner_weights*

```
model <- estimate_pls(
  data = mobi,
  measurement_model = mm,
  structural_model = sm,
  model = NULL,
  # optional: how paths are estimated
  # defaults to path weighting
  inner_weights = path_weighting,
  missing = mean_replacement,
  missing_value = NA
)
```

Further reading: Tenenhaus et al. (2005). PLS path modeling.

Estimating a model - *missing*

```
model <- estimate_pls(
  data = mobi,
  measurement_model = mm,
  structural_model = sm,
  model = NULL,
  inner_weights = path_weighting,
  # optional: how missing values are treated
  # defaults to mean replacement
  missing = mean_replacement,
  missing_value = NA
)
```

Estimating a model - *missing*

Alternatively: Disregard observations with missing values

```
model <- estimate_pls(
  data = mobi,
  measurement_model = mm,
  structural_model = sm,
  model = NULL,
  inner_weights = path_weighting,
  # do not use observations with missing values
  missing = na.omit,
  missing_value = NA
)
```

Estimating a model

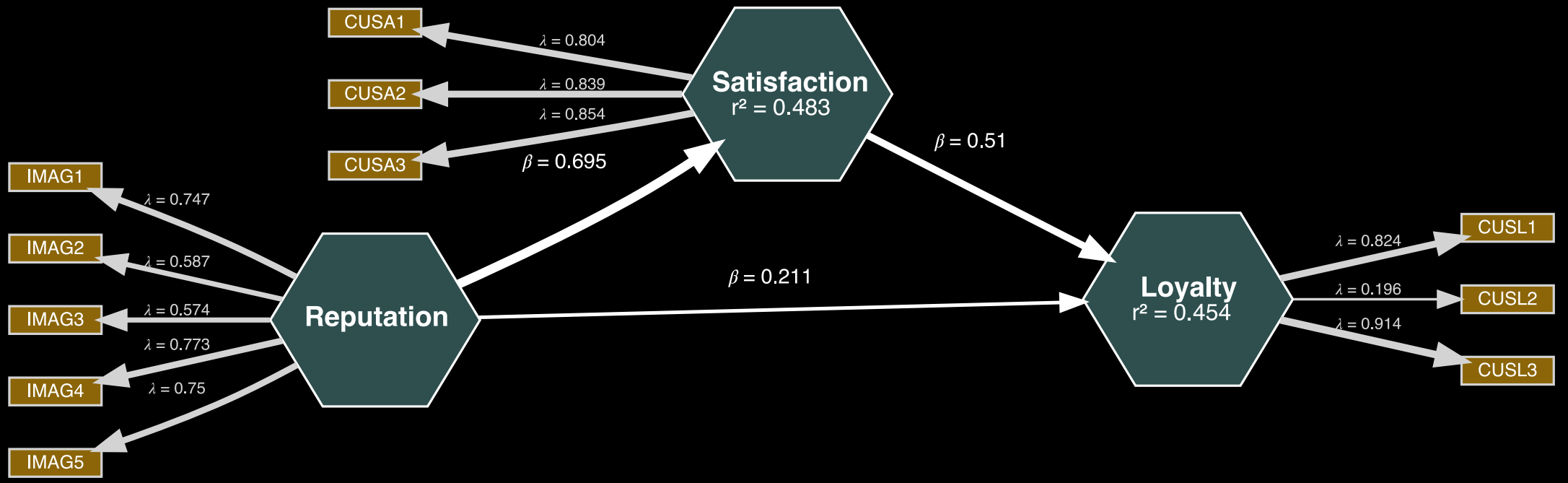
```
model <- estimate_pls(
  data = mobi,
  measurement_model = mm,
  structural_model = sm
)
```

```
## Generating the seminr model

## All 250 observations are valid.
```

The SEMinR model object - plot

```
plot(model)
```



The SEMinR model object - subobjects

```
> model$
```

The SEMinR model object - subobjects

```
> model$
```

Means of all indicators:

```
model$meanData
```

```
## IMAG1 IMAG2 IMAG3 IMAG4 IMAG5 CUSA1 CUSA2 CUSA3 CUSL1 CUSL2 CUSL3
## 7.640 7.780 6.744 7.588 7.932 7.988 7.128 7.316 7.452 4.988 7.668
```

The SEMinR model object - subobjects

```
> model$
```

Means of all indicators:

```
model$meanData
```

```
## IMAG1 IMAG2 IMAG3 IMAG4 IMAG5 CUSA1 CUSA2 CUSA3 CUSL1 CUSL2 CUSL3
## 7.640 7.780 6.744 7.588 7.932 7.988 7.128 7.316 7.452 4.988 7.668
```

Number of iterations until convergence:

```
model$iterations
```

```
## [1] 6
```

And many more!

The SEMinR model object - summary

```
summary(model)
```

```
##
## Results from  package seminr (2.0.3)
##
## Path Coefficients:
##              Satisfaction Loyalty
## R^2                 0.483   0.454
## AdjR^2              0.481   0.449
## Reputation          0.695   0.211
## Satisfaction            .   0.510
##
## Reliability:
##              alpha  rhoC   AVE  rhoA
## Reputation   0.723 0.819 0.478 0.745
## Satisfaction 0.779 0.871 0.693 0.786
## Loyalty      0.472 0.721 0.518 0.743
##
## Alpha, rhoC, and rhoA should exceed 0.7 while AVE should exceed 0.5
```

The SEMinR model object - summary subobjects

```
summarymodel <- summary(model)
```

The SEMinR model object - summary subobjects

```
summarymodel <- summary(model)
```

Informations on discriminant validity:

```
summarymodel$validity$vif_items
```

```
## Reputation :
## IMAG1 IMAG2 IMAG3 IMAG4 IMAG5
## 1.468 1.225 1.259 1.510 1.403
##
## Satisfaction :
## CUSA1 CUSA2 CUSA3
## 1.505 1.762 1.644
##
## Loyalty :
## CUSL1 CUSL2 CUSL3
## 1.415 1.010 1.427
```

Troubleshooting - *'x' must be numeric*

```
# transform variable to character
dt2 <- mobi %>%
  mutate(IMAG1 = IMAG1 %>%
           as.character())
```

Troubleshooting - *'x' must be numeric*

```
# transform variable to character
dt2 <- mobi %>%
  mutate(IMAG1 = IMAG1 %>%
           as.character())
```

```
# this creates an error
estimate_pls(dt2, mm, sm)
```

```
Generating the seminr model
argument is not numeric or logical: returning NA
All 250 observations are valid.
Error in colMeans(x, na.rm = TRUE) : 'x' must be numeric
```

Troubleshooting - *'x' must be numeric*

```
# transform variable to character
dt2 <- mobi %>%
  mutate(IMAG1 = IMAG1 %>%
           as.character())
```

```
# this creates an error
estimate_pls(dt2, mm, sm)
```

```
Generating the seminr model
argument is not numeric or logical: returning NA
All 250 observations are valid.
Error in colMeans(x, na.rm = TRUE) : 'x' must be numeric
```

```
# select variables that are not numeric
dt2 %>% select(!where(is.numeric))
```

Troubleshooting - *undefined columns selected*

```
mm2 <- constructs(
  composite("Reputation",
  # what happens if I make a typo when choosing indicators?
            multi_items("IMGA", 1:5)),
  composite("Satisfaction",
            multi_items("CUSA", 1:3)),
  composite("Loyalty",
            multi_items("CUSL", 1:3))
)
```

Troubleshooting - *undefined columns selected*

```
mm2 <- constructs(
  composite("Reputation",
  # what happens if I make a typo when choosing indicators?
            multi_items("IMGA", 1:5)),
  composite("Satisfaction",
            multi_items("CUSA", 1:3)),
  composite("Loyalty",
            multi_items("CUSL", 1:3))
)
```

```
estimate_pls(dt, mm2, sm)
```

```
Error in `[.data.frame`(data, , mmMatrix[which(!grepl("\\*",
mmMatrix[, : undefined columns selected
```

Troubleshooting - *attempt to apply non-function*

```
sm2 <- relationships(
  paths(
    from = "Reputation",
    to = c("Satisfaction", "Loyalty")
  ),
  paths(
    from = "Satisfaction",
# what happens if I include a rogue construct?
    to = c("Loyalty", "Quality")
  )
)
```

Troubleshooting - *attempt to apply non-function*

```
sm2 <- relationships(
  paths(
    from = "Reputation",
    to = c("Satisfaction", "Loyalty")
  ),
  paths(
    from = "Satisfaction",
# what happens if I include a rogue construct?
    to = c("Loyalty", "Quality")
  )
)
```

```
estimate_pls(dt, mm, sm2)
```

```
Generating the seminr model
All 250 observations are valid.
Error in measurement_mode_scheme[[i]](mmMatrix, i, normData,
construct_scores) : attempt to apply non-function
```

Summary

- Model estimation with estimate_pls()
- Model object and model summary object
- Common errors

Sources for this video

Hair, J. F., Hult, G. T. M., Ringle, C. M., & Sarstedt, M. (2017). A primer on partial least squares structural equation modeling (PLS-SEM) (Second edition). Sage.

Henseler, J., Ringle, C. M., & Sinkovics, R. R. (2009). The use of partial least squares path modeling in international marketing. In R. R. Sinkovics & P. N. Ghauri (Eds.), Advances in International Marketing (Vol. 20, pp. 277–319). Emerald Group Publishing Limited. https://doi.org/10.1108/S1474-7979(2009)0000020014

Ray, S. & Danks. N. (2020). SEMinR Vignette. https://cran.r-project.org/web/packages/seminr/vignettes/SEMinR.html

Tenenhaus, M., Vinzi, V. E., Chatelin, Y.-M., & Lauro, C. (2005). PLS path modeling. Computational Statistics & Data Analysis, 48(1), 159–205. https://doi.org/10.1016/j.csda.2004.03.005