An alternative modeling strategy: Partial Least Squares

Dr. Jon Starkweather, Research and Statistical Support consultant.

Partial Least Squares (PLS) modeling is often used as an alternative to traditional modeling techniques. Unlike traditional modeling techniques which rely upon covariance decomposition, PLS is a variance based (or components based) technique and does not carry with it many of the assumptions of covariance methods (i.e. distributional assumptions). It is sometimes considered an analysis of last resort because large samples are not as necessary with it, and PLS is less sensitive to multicollinearity. However, PLS is primarily descriptive when used with small samples and is still constrained with respect to making inferences about parameters when sample sizes are small. The benefit of having the ability to do descriptive analysis with small samples is that PLS can fit models with non-linear relationships and non-Gaussian distributions among the variables in addition to the traditional linear and Gaussian situations.

PLS is also quite versatile; it can be used as a regression technique, a principal components technique, a canonical correlation technique, or a path modeling (or structural equation modeling) technique. It is well documented that PLS is biased because the optimization is local rather than global level; however, as sample size increases PLS becomes less bias. PLS can be used to make inferences about parameters when sample sizes are large. PLS is often used when other methods fail (i.e. a slightly biased estimate is better than no estimate).

As an example, we will first model a simulated data set using traditional modeling techniques using a popular method and package. John Fox’s (2010) package 'sem' is one of the more established modeling packages in R and will be used here to demonstrate how certain data sets do not converge on a specified model.

Example

First, import the data from the internet and run the ubiquitous ‘head’ function to get a look at the data. The example data contains 20 variables (v1 – v20) and 1000 cases. Here we will name the data ‘pls.data’.
Next, create a covariance matrix object which will be passed on to the ‘sem’ function. The covariance object is named ‘cov.m’ (some of the matrix in the image below is not shown).
Next, load the ‘sem’ package by typing: `library(sem)` in the R console. Then, specify the measurement model (i.e. confirmatory factor model). The model specification syntax is given below (not in an image) due to its length.

```
measurement.model <- specify.model()
```

```
P1 -> v1, lam11, NA
P1 -> v2, lam12, NA
F2 -> v3, lam21, NA
F2 -> v4, lam22, NA
F3 -> v5, lam31, NA
F3 -> v6, lam32, NA
F3 -> v7, lam33, NA
F3 -> v8, lam34, NA
F4 -> v9, lam41, NA
F4 -> v10, lam42, NA
F5 -> v11, lam51, NA
F5 -> v12, lam52, NA
F5 -> v13, lam53, NA
F5 -> v14, lam54, NA
F5 -> v15, lam55, NA
F5 -> v16, lam56, NA
v1 <- v1, var1, NA
v2 <- v2, var2, NA
v3 <- v3, var3, NA
v4 <- v4, var4, NA
v5 <- v5, var5, NA
v6 <- v6, var6, NA
```
Next, we run the measurement model; but unfortunately, it does not converge. 

So, we detach the ‘sem’ package using the following command: `detach("package:sem")` and decide to use a PLS strategy. The ‘plspm’ package (PLS Path Modeling; Sanchez & Trinchera, 2010) provides functions for conducting and graphing a variety of PLS techniques; such as PLS regression with a single outcome, PLS canonical correlation, PLS regression with multiple outcomes (similar to canonical correlation, but with directionality implied between the two composite variates), PLS principal components analysis, and PLS path modeling (i.e. SEM).

**PLS Path Modeling**

Load the package (which three dependencies [amap, diagram, shape]).
First, we must create a matrix which expresses the inner (structural) model; this model simply shows the relationships among the latent variables; where the column variable 'causes' the row variable(s) if a 'one' is in the intersecting cell (e.g. f1 and f2 cause f3 –> columns 1 and 2 cause row 3).

Next, create the list which expresses the outer (measurement) model; this model simply shows the relationships between the manifest variables and the latent variables (e.g. variables v1 and v2 are related to the first factor [f1]). Although we create a list object in R, this is often referred to as the outer matrix in the PLS literature.
Next, create a vector which identifies the "mode" of indicators which were used (i.e. "A" for reflective measurement or "B" for formative measurement). Recall, 'Reflective' measurement is said to occur when each manifest variable is "caused by" a latent variable and 'Formative' measurement is said to occur when each manifest variable "causes" the latent variable. Below, all 5 latent variables in our model are "reflectively" measured (i.e. each latent causes the observed scores on the manifest variables).

```r
> mode.vec <- c("A", "A", "A", "A")
> mode.vec
```

Finally, we can run the Partial Least Squares Path Model. One of the benefits of using the ‘plspm’ package rather than one of the other PLS packages available in R, is that the ‘plspm’ package offers some very easy to use and interpret output. Each function provides a description of the function’s output items and shows how to extract or reference them.

```r
> pls.model.1 <- plspm(x = pls.data[,4:23], inner = inner.matrix, outer = outer.list, mod$s
+   scheme = "factor", scaled = TRUE, plsr = TRUE, tol = 0.00001, iter = 100)
> pls.model.1

PARTIAL LEAST SQUARES PATH MODELING (PLS-PM)
---------------------------------------------
Results available in the following objects:

Name    Description
1 $outer.mod  "outer model"
2 $inner.mod  "inner model"
3 $latents  "scaled T(N)"
4 $scores  "LVs for scaled=FALSE"
5 $out.weights  "outer weights"
6 $loadings  "loadings"
7 %path.coefs%  "path coefficients matrix"
8 $r.sqr  "R-squared"
9 $outer.cor  "outer correlations"
10 $inner.sum  "summary inner model"
11 $effects  "total effects"
12 %unidim%  "unidimensionality"
13 $gof  "goodness-of-fit"
14 $data  "data matrix"
```

Using the ‘summary’ function on a ‘plspm’ object provides a well-documented and indexed summary of the analysis’ output. Below you can see that the current summary provides a very thorough summary with labels for each element which makes interpretation very straightforward. In fact, the output (from the ‘summary’) is so large that is necessitates four screen capture images to display it all here.
Continued below.
Continued below.
$f3$
concept  value
1 R2  0.7419
2 Intercept  0.0000
3 path_f1  0.6743
4 path_f2  0.5355

$f4$
concept  value
1 R2  0.8105
2 Intercept  0.0000
3 path_f2  0.5338
4 path_f3  0.5056

$f5$
concept  value
1 R2  0.8695
2 Intercept  0.0000
3 path_f3  0.4773
4 path_f4  0.5099

-----------------------------------------
CORRELATIONS BETWEEN LVs
                     f1  f2  f3  f4  f5
f1  1.0000  0.0005 0.675 0.324 0.523
f2  0.0005  1.0000 0.536 0.704 0.706
f3  0.6746  0.5359 1.000 0.784 0.877
f4  0.3243  0.7936 0.784 1.000 0.884
f5  0.5231  0.7060 0.877 0.884 1.000

-----------------------------------------
SUMMARY INNER MODEL
LV.Type Measure M V s R.square Av.Commu Av.Redun AVE
f1 Exogen Rflct  2 0.000 0.866 0.000 0.866
f2 Exogen Rflct  3 0.000 0.692 0.000 0.692
f3 Endogen Rflct  6 0.742 0.638 0.473 0.638
f4 Endogen Rflct  4 0.810 0.754 0.611 0.754
f5 Endogen Rflct  5 0.870 0.786 0.657 0.786

-----------------------------------------
GOODNESS-OF-FIT
 Gof  value
1 Absolute  0.7632
2 Relative  0.9754
3 Outer.mod  0.9996
4 Inner.mod  0.9765

-----------------------------------------
TOTAL EFFECTS
relationships dir.effects ind.effects tot.effects
1 f1->f2 0.000 0.000 0.000
2 f1->f3 0.674 0.000 0.674
3 f1->f4 0.000 0.340 0.340

Continued below.
Another big advantage to using the ‘plspm’ package (rather than others available for PLS modeling) is the ability to produce a path diagram based on the model fitted.
Another advantage to using the ‘plspm’ package is the ability to conduct bootstrapped validation of a PLS path model using the ‘boot.val’ optional argument to the ‘plspm’ function.

```
> pls.model.2 <- plspm(x = pls.data[,4:23], inner = inner.matrix, outer = outer.list, mod$ +   scheme = "factor", scaled = TRUE, pls.r = TRUE, boot.val = TRUE, bx = 200, tol = 0.005
> pls.model.2

PARTIAL LEAST SQUARES PATH MODELING (PLS-PM)

Results available in the following objects:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$outer.mod&quot;</td>
<td>outer model</td>
</tr>
<tr>
<td>$inner.mod&quot;</td>
<td>inner model</td>
</tr>
<tr>
<td>$slatents&quot;</td>
<td>scaled LVs</td>
</tr>
<tr>
<td>$scores&quot;</td>
<td>LVs for scaled=FALSE</td>
</tr>
<tr>
<td>$out.weights&quot;</td>
<td>outer weights</td>
</tr>
<tr>
<td>$loadings&quot;</td>
<td>loadings</td>
</tr>
<tr>
<td>$path.coefs&quot;</td>
<td>path coefficients matrix</td>
</tr>
<tr>
<td>$r.sqr&quot;</td>
<td>R-squared</td>
</tr>
<tr>
<td>$outer.cor&quot;</td>
<td>outer correlations</td>
</tr>
<tr>
<td>$inner.sum&quot;</td>
<td>summary inner model</td>
</tr>
<tr>
<td>$effects&quot;</td>
<td>total effects</td>
</tr>
<tr>
<td>$unidim&quot;</td>
<td>unidimensionality</td>
</tr>
<tr>
<td>$gof&quot;</td>
<td>goodness-of-fit</td>
</tr>
<tr>
<td>$boot&quot;</td>
<td>bootstrap results</td>
</tr>
<tr>
<td>$data&quot;</td>
<td>data matrix</td>
</tr>
</tbody>
</table>

You can also use the function 'summary'
```

Notice in the above table, there is a “$boot” element in the output. The rest of the output is identical to what was displayed above. The “$boot” element contains the cross validation output, which is the only part of the output displayed below.
Continued below.
Interpretation was excluded from this article because the output of the functions covered is considered fairly intuitive. However, if one would like more information on interpreting PLS models, see Chin (2010).

Until next time, I’ll drive my Chevy to the levee.
References & Resources


