Three Step Latent Class Analysis in R and STATA

Daniel Tompsett and Bianca L De Stavola

Great Ormond Street Institute of Child Health, University College London

d.tompsett@ucl.ac.uk

Work funded by MRC Methodology Grant: MR/R025215/1

September 6, 2022

Aims and Content

- Three step Latent Class (LCA-3) analysis is a fairly involved analysis technique from a coding standpoint.
- Two methods are described in [5], a BCH and ML method.
- Dedicated software for both methods are available via Latent GOLD [4] or Mplus [1].
- In STATA the BCH method can be performed with the custom LCA_Distal_BCH function [2].
- Little to no documentation on an implementation of the ML method in STATA.

Why integrate R and STATA?

- LCA-3 via the ML is not currently possible in R.
- Steps 1 and 2 however can be performed in R quite easily.
- Its then relatively straightforward to apply step 3 in STATA.
- We detail how the ML can be performed by integrating R and standard STATA code.
- Show how to perform Causal analysis with LCA.

Setting

- Suppose m binary indicator response variables $Y_1, \ldots, Y_m$.
- We wish to identify specific patterns of response in the Y_i.
- The collection of these patterns forms a categorical latent class variable C.
- We are interested in the effect of some exposure X on patterns of response in C.
- This is confounded by some variable L.

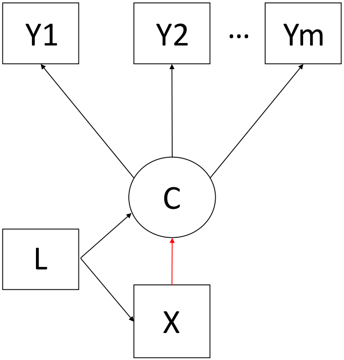


Figure: Latent Class Setting

Step 1: Structural Equation Model

- The first step in 3-step LCA is to estimate the distinct response patterns (C) in the Y_i using a Latent Class Model (LCM), a type of Structural Equation Model (SEM).

$$P(\mathbf{Y}) = \sum_{j=1}^{c} P(C=j) \prod_{k=1}^{m} \prod_{l=1}^{R_k} P(Y_k = l | C = j)^{IY_k = l} \qquad (1)$$

- $P(C=j)$ is the structural element which models the latent class C and its relationship with exogeneous (non indicator) variables.
- $P(Y_k = l | C = j)^{IY_k = l}$ is the measurement element of the model, coding the relationship between the latent classes and indicator variables.

Step 1: Structural Equation Model

- The number of classes in C to fit is typically user specified.
- The key outputs are the response pattern of each fitted class, and the posterior probabilities $P(C = j | Y_1, \ldots, Y_m)$, telling us the probabilities of each individual belonging to each class.
- We do not include X or L in the structural element in 3-step

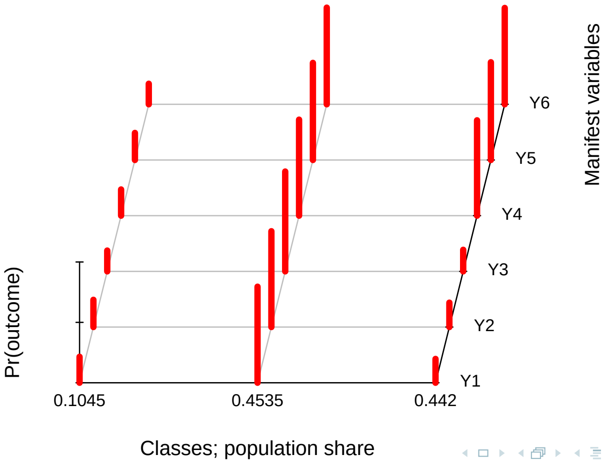
We can fit a LCM using the poLCA package [3].

```
f<-cbind(Y1,Y2,Y3,Y4,Y5,Y6)~1

polca<-poLCA(f,nclass=3,
       data=datasim, nrep = 1,
       na.rm=F, graphs=T,
       maxiter = 100000,verbose=TRUE)
```

The posterior probabilities of belonging to each class are defined as

```
probs<-as.data.table(polca$posterior)
```

Step 1 in R



Y6
Y5
Y4
Y3
Y2
Y1
Manifest variables
Pr(outcome)
0.1045
0.4535
0.442
Classes; population share

In STATA we can fit the LCM with the gsem command

```
gsem(Y1-Y6<-), logit lclass(class 3) nolog
```

The posterior probabilities of belonging to each class are given by

```
estat lcgof
predict classpost*, classposteriorpr
/* these are the individual predictions*/
```

Step 2: Modal Class Assignment

- In step 2 we assign each individual an estimated class W.
- We use modal assignment, that is each individual is assigned to the class for which their posterior probability is the highest.

$$W = argmax_j(P(C = j|Y_1, \ldots, Y_m))$$

- One could then use W as an outcome in any analysis model.
- This will be biased, because not all individuals will be assigned their true class C.
- The probability of misclassification in the data can be defined in a matrix Q where .

$$Q_{j,i} = P(W = i|C = j)$$

- We then estimate Q using

$$Q_{j,i} = P(W = i|C = j) = \frac{P(C = j|W = i) * N_i}{\sum k = 1^c P(C = j, W = k) N_k}$$

Where N_i is the number of individuals classified into class i by W and

$$P(C = j|W = i) = \frac{\sum_{W_n = i} P(C_n = j|\mathbf{Y}_n)}{N_i}.$$

- This can be used to establish the effect of X on C, by correcting for the effect of X on W.

We obtain *W* as

```
probs<-as.data.table(polca$posterior)
datasim$W<-modclass<-apply(probs,1,which.max)
```

Estimating Q is more involved we first obtain $P(C = j|W = i)$

```
nclass=3
Ptable<-cbind(probs,modclass)
Pmatrix<-matrix(0,nclass,nclass)
Npmatrix<-matrix(0,nclass,nclass)
for (i in 1:nclass){
for (j in 1:nclass){
Pmatrix[i,j]<-sum(subset(Ptable,modclass==i)[,..j])
Npmatrix[i,j]<-Pmatrix[i,j]*table(modclass)[i]
}}
```

The Q matrix is then calculated as

```
denom<-colSums(Npmatrix)
Qmatrix<-matrix(0,nclass,nclass)

for (i in 1:nclass){
for (j in 1:nclass){

Qmatrix[j,i]<-Npmatrix[i,j]/denom[j]

}}
```

In our example the Q matrix is calculated as.

```
            [,1]       [,2]      [,3]
[1,] 0.650394116 0.05652605 0.2930798
[2,] 0.007400971 0.89847283 0.0941262
[3,] 0.041348205 0.09644037 0.8622114
```

Q can also be calculated in STATA but is a much longer code. It as such wont be shown here, but is available on request.

- In the final step we refit in LCM, but with W the single manifest variable, and include X and L in the structural element. This simplifies the SEM to

$$P(W|Z) = \sum_{j=1}^{C} P(C = j|X, L) \prod_{l=1}^{C} P(W = l|C = j)^{IW=l}$$

- The measurement element is now just the misclassification probabilities, that we fix to the values in Q.
- The structural element then gives us the effect of X on C, controlled for L

One quirk, as we are fitting a multinomial logistic regression model, (with reference class 1 say), the probabilities in Q must be in the same format.

```
lQ<-log(Qmatrix/Qmatrix[,1])
lQ

     [,1]       [,2]       [,3]
[1,]    0 -2.4428770 -0.7971335
[2,]    0  4.7990852  2.5430252
[3,]    0  0.8468959  3.0374715

datasim$lq<-c(as.vector(t(lQ[,-1])),rep(0,(n-6)))
```

Step 3 in STATA

```
use datasim.dta
local L_12=lq[1]
local L_13=lq[2]
local L_22=lq[3]
local L_23=lq[4]
local L_32=lq[5]
local L_33=lq[6]

capture noisily gsem
(1: 2.W<-_cons@`L_12')(1: 3.W<-_cons@`L_13') \\\
(2: 2.W<-_cons@`L_22')(2: 3.W<-_cons@`L_23') \\\
(3: 2.W<-_cons@`L_32')(3: 3.W<-_cons@`L_33') \\\
(class<- i.X i.L1 L2),mlogit \\\
 vce(robust) lclass(class 3) nocapslatent
```

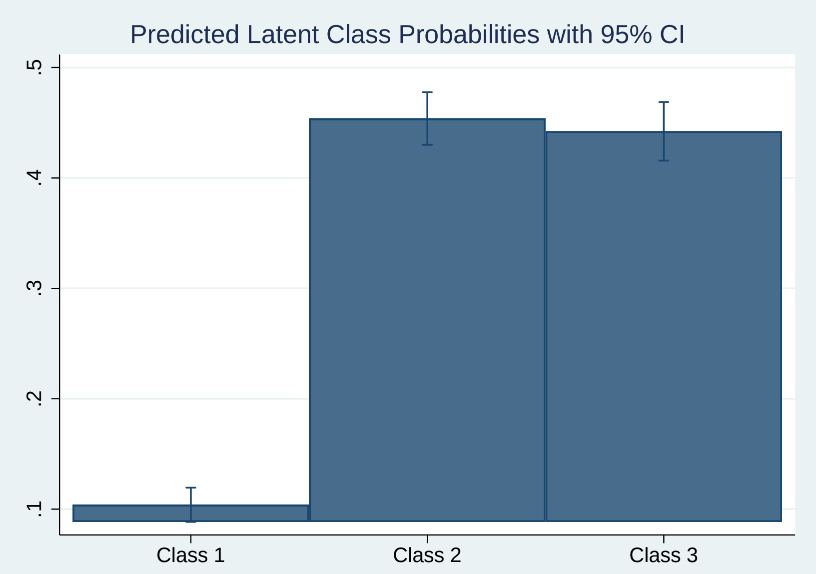
	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
1.class	(base outcome)					
2.class						
1.X	.8934224	.2704057	3.30	0.001	.3634369	1.423408
1.L1	1.674021	.5496482	3.05	0.002	.5967305	2.751312
L2	1.908653	.2027918	9.41	0.000	1.511188	2.306117
_cons	.1137692	.1460407	0.78	0.436	-.1724652	.4000037
3.class						
1.X	.9052126	.287631	3.15	0.002	.3414661	1.468959
1.L1	1.850334	.5646795	3.28	0.001	.7435823	2.957085
L2	1.877882	.2062549	9.10	0.000	1.47363	2.282134
_cons	.0775718	.1748468	0.44	0.657	-.2651216	.4202652

```
. estat lcprob

Latent class marginal probabilities                 Number of obs = 2,500
```

	Margin	Delta-method std. err.	[95% conf.	interval]
class				
1	.1039517	.0079034	.0894499	.1204935
2	.4538213	.0121718	.4300888	.4777654
3	.442227	.0135166	.4159235	.4688586

```
margins, predict(classpr class(1)) \\\
predict(classpr class(2)) \\\
predict(classpr class(3))

marginsplot, recast(bar) xtitle("") ytitle("")\\\
xlabel(1 "Class 1" 2 "Class 2" 3 "Class 3")\\\
title("Predicted Latent Class Probabilities\\\
with 95\% CI")
```

Step 3 in STATA



Calculating the ACE

- We have the effect of X on C on the log odds scale.
- Typically in LCA we are interested the effect X on belonging to a particular class on he probability scale.
- This is often known as the average causal effect (ACE).
- This can be done using gsem with the margins command and dydx.

Calling the ACE

```
. margins,dydx(i.X) predict(classpr class(1))

Average marginal effects                          Number of obs = 2,500
Model VCE: Robust

Expression: Predicted probability (1.class), predict(classpr class(1))
dy/dx wrt:  1.X
```

	dy/dx	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
1.X	-.0554685	.0167355	-3.31	0.001	-.0882694	-.0226676

```
Note: dy/dx for factor levels is the discrete change from the base level.
```

We can also use Inverse Probability Weighting (IPW)

```
logit X i.L1 L2 , nolog base
cap drop ps
predict ps,pr
replace ps=1-ps if X==0
gen wt=1/ps

gsem\\\
(1: 2.W<-_cons@‘L_12’)(1: 3.W<-_cons@‘L_13’)\\\
(2: 2.W<-_cons@‘L_22’ )(2: 3.W<-_cons@‘L_23’)\\\
(3: 2.W<-_cons@‘L_32’)(3: 3.W<-_cons@‘L_33’)\\\
(class<- i.X)[iw=wt],emopts(iterate(25))mlogit\\\
 vce(robust) lclass(class 3) nocapslatent
```

```
. margins,dydx(i.X) predict(classpr class(1))

Conditional marginal effects                    Number of obs = 2,500
Model VCE: Robust

Expression: Predicted probability (1.class), predict(classpr class(1))
dy/dx wrt:  1.X
```

	dy/dx	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
1.X	-.0455998	.0240889	-1.89	0.058	-.0928132	.0016137

Note: dy/dx for factor levels is the discrete change from the base level.

Conclusion

- Three step LCA is an involved method than can be performed either in STATA or by using both STATA and R.
- We demonstrated an alternative means to perform the ML methdology without the use of MPLUS of Latent GOLD.
- Possibility of developing the methodology further, specifically simplifying calculation of Q in STATA.

Tihomir Asparouhov and Bengt Muthn.
Auxiliary variables in mixture modeling: Three-step approaches using m plus.
Structural Equation Modeling: A Multidisciplinary Journal, 21:329–341, 06 2014.

L Huang, J Dziak, B Bray, and A Wagner.
LCA Distal BCH Stata function Users Guide.
The Methodology Centre, Penn State.

Drew A. Linzer and Jeffrey B. Lewis.
poLCA: An R package for polytomous variable latent class analysis.
Journal of Statistical Software, 42(10):1–29, 2011.

J Vermunt and J Magidson.
Latent GOLD.
https://www.statisticalinnovations.com/latent-gold-5-1/.

Jeroen K. Vermunt.
Latent class modeling with covariates: Two improved three-step approaches.
Political Analysis, 18(4):450–469, 2010.