Latent class analysis and finite mixture models with Stata

Isabel Canette

Principal Mathematician and Statistician

StataCorp LLC

2017 Stata Users Group Meeting
Madrid, October 19th, 2017

Introduction

"Latent class analysis" (LCA) comprises a set of techniques used to model situations where there are different subgroups of individuals, and group memebership is not directly observed, for example:.

- Social sciences: a population where different subgroups have different motivations to drink.
- Medical sciences: using available data to identify subgroups of risk for diabetes.
- Survival analysis: subgroups that are vulnerable to different types of risks (competing risks).
- Education: identifying groups of students with different learning skills.
- Market research: identifying different kinds of consumers.

The scope of the term “latent class analysis” varies widely from source to source.

Collin and Lanza (2010) discuss some of the models that are usually considered LCA. Also, they point out: “ In this book, when we refer to latent class models we mean models in which the latent variable is categorical and the indicators are treated as categorical”.

In Stata, we use " LCA" to refer to a wide array of models where there are two or more unobserved classes

- Dependent variables might follow any of the distributions supported by **gsem**, as logistic, Gaussian, Poisson, multinomial, negative binomial, Weibull, etc.(**help gsem family and link options**)
- There might be covariates (categorical or continuos) to explain the dependent variables
- There might be covariates to explain class membership

Stata adopts a model-based approach to LCA. In this context, we can see LCA as group analysis where the groups are unknown.

Let's see an example, first with groups and then with classes:

Below we use **group()** option fit regressions to the childweight data, weight vs age, different regressions per sex:

```
. gsem (weight <- age), group(girl) ginvariant(none) ///
>   vsquish nodvheader noheader nolog

Group           : boy                     Number of obs     =        100

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
weight       |
         age |   3.481124   .1987508    17.52   0.000      3.09158    3.870669
       _cons |   5.438747   .2646575    20.55   0.000     4.920028    5.957466
-------------+----------------------------------------------------------------
var(e.weight)|     2.4316   .3438802                      1.842952    3.208265
------------------------------------------------------------------------------

Group           : girl                    Number of obs     =         98

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
weight       |
         age |   3.250378   .1606456    20.23   0.000     2.935518    3.565237
       _cons |   4.955374   .2152251    23.02   0.000     4.533541    5.377207
-------------+----------------------------------------------------------------
var(e.weight)|   1.560709   .2229585                      1.179565     2.06501
------------------------------------------------------------------------------
```

Group analysis allows us to make comparisons between these equations, and easily set some common. (**help gsem group options**)

Now let's assume that we have the same data, and we don't have variable **girl**. We suspect that there are two groups that behave different.

```
. gsem (weight <- age),  lclass(C 2)  lcinvariant(none) ///
>   vsquish nodvheader noheader nolog
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
1.C	(base outcome)					
2.C						
_cons	.5070054	.2725872	1.86	0.063	-.0272557	1.041267

Class : 1

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
weight						
age	5.938576	.2172374	27.34	0.000	5.512798	6.364353
_cons	3.8304	.2198091	17.43	0.000	3.399582	4.261218
var(e.weight)	.6766618	.1817454			.3997112	1.145505

Class : 2

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
weight						
age	2.90492	.2375441	12.23	0.000	2.439342	3.370498
_cons	5.551337	.4567506	12.15	0.000	4.656122	6.446551
var(e.weight)	1.52708	.2679605			1.082678	2.153893

The second table on the LCA model same structure as the output from the group model.

In addition, the LCA output starts with a table corresponding to the class estimation. This is a binary (**logit**) model used to find the two classes.

In the latent class model all the equations are estimated jointly and all parameters affect each other, even when we estimate different parameters per class.

How do we interpret these classes? We need to analyze our classes and see how they relate to other variables in the data. Also, we might interpret our classes in terms of a previous theory, provided that our analysis is in agreement with the theory. We will see post-estimation commands that implement the usual tools used for this task.

Latent class analysis in Stata is an extension of the classic latent class analysis.

Stata documentation and formulas refer to the general model, and don't match the notation and approach you will see on the classic LCA literature (though results match).

We'll introduce the classic approach to LCA and discuss how Stata approach generalizes it.

Example: Role conflict dataset

```
. use gsem_lca1
(Latent class analysis)

. notes in 1/4

_dta:
  1.  Data from Samuel A. Stouffer and Jackson Toby, March 1951, "Role conflict
      and personality", _The American Journal of Sociology_, vol. 56 no. 5,
      395-406.
  2.  Variables represent responses of students from Harvard and Radcliffe who
      were asked how they would respond to four situations. Respondents
      selected either a particularistic response (based on obligations to a
      friend) or universalistic response (based on obligations to society).
  3.  Each variable is coded with 0 indicating a particularistic response and 1
      indicating a universalistic response.
  4.  For a full description of the questions, type "notes in 5/8".
```

```
. describe

Contains data from gsem_lca1.dta
  obs:           216                          Latent class analysis
 vars:             4                          10 Oct 2017 12:46
 size:           864                          (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
accident        byte    %9.0g                 would testify against friend in
                                                accident case
play            byte    %9.0g                 would give negative review of
                                                friend´s play
insurance       byte    %9.0g                 would disclose health concerns to
                                                friend´s insurance company
stock           byte    %9.0g                 would keep company secret from
                                                friend
-------------------------------------------------------------------------------
Sorted by: accident  play  insurance  stock
```

```
. list in 120/121
```

	accident	play	insura~e	stock
120.	1	0	1	1
121.	1	1	0	0

For each observation, we have a vector of responses
$\mathbf{Y} = (Y_1, Y_2, Y_2, Y_4)$ (I am omitting an observation index)

Classic approach

Let's assume that we have two classes, $C1$ and $C2$.
The probabilty of Y taking a value y can be expressed as:

$$P(Y = y|C1) * P(C1) + P(Y = y|C2) * P(C2)$$

Which, under the assumption of conditional independence, is:

$$\prod_{j=1}^{4} P(Y_j = y_j|C1) \times P(C1) + \prod_{j=1}^{4} P(Y_j = y_j|C2) \times P(C2)$$

In short, the likelihood contribution for an observation would be:

$$L = \sum_{k=1,2} \prod_{j=1}^{4} P(Y_j = 1|Ck)^{y_j} \times (1 - P(Y_j = 1|Ck))^{1-y_j} \times P(Ck)$$

Maximizing the sum of the log-likelihood contributions from all observations, we obtain the values $P(Y_j = rj|Ck)$ and $P(Ck)$. In the literature, you will see generalizations of this formula, like

$$L = \sum_{k=1,\ldots m} \prod_{j=1}^{4} \prod_{rj=1}^{Rj} P(Y_j = rj|Ck)^{(I(y_j=rj))} \times P(Ck)$$

where $rj, j = 1 \ldots Rj$ are the possible values for variable Y_j.

Stata (Model-based) approach

The description before corresponds to a non-parametric estimation. We estimate the probabilities directly, not through a parameterization.
Now, how do we do it in Stata?

```
.gsem (accident play insurance stock <- ), logit lclass(C 2)
```

We are fitting a logit model for each class, with no covariates. Because there are no covariates, estimating the constant is equivalent to estimating the probability: $p = F(constant)$, where F is the inverse logit function.

The model-based approach can be represented as a mixed model:

$$L = f(y;\Theta_1) \times P(C1) + f(y;\Theta_2) \times P(C2)$$

Where

$$f(y;\Theta_k) = \prod_{i=1}^{4} p_{jk}^{y_i} \times (1 - p_{jk})^{1-y_i}$$

and p_{jk} is expressed as $exp(cons_{jk})/(1 + exp(cons_{jk})$
gsem also represents class probabilities $P(Ck)$ with a logit model.

By default, we are fitting the non-parametric model, but this flexibility allows us to include covariates to model the class membership probabilities, the conditional probabilities, or both.

Now, let's fit the model.

```
. gsem(accident play insurance stock <- ),logit lclass(C 2) ///
>     vsquish nodvheader noheader nolog
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
1.C	(base outcome)					
2.C						
_cons	-.9482041	.2886333	-3.29	0.001	-1.513915	-.3824933

Class : 1

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
accident						
_cons	.9128742	.1974695	4.62	0.000	.5258411	1.299907
play						
_cons	-.7099072	.2249096	-3.16	0.002	-1.150722	-.2690926
insurance						
_cons	-.6014307	.2123096	-2.83	0.005	-1.01755	-.1853115
stock						
_cons	-1.880142	.3337665	-5.63	0.000	-2.534312	-1.225972

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Class : 2						

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
accident						
_cons	4.983017	3.745987	1.33	0.183	-2.358982	12.32502
play						
_cons	2.747366	1.165853	2.36	0.018	.4623372	5.032395
insurance						
_cons	2.534582	.9644841	2.63	0.009	.6442279	4.424936
stock						
_cons	1.203416	.5361735	2.24	0.025	.1525356	2.254297

After our estimation, the **predict** command allows us to obtain many predictions:

Probabilities of positive outcome, conditional on class	
$P(Y_1 = 1 \| C2)$	`predict pr1c, mu outcome(accident) class(2)`
$P(Yj = 1 \| C2) \forall j$	`predict prc*, mu class(2)`
$P(Y_1 = 1 \| Ck) \forall k$	`predict prc*, mu outcome(accident)`
$P(Yj = 1 \| Ck) \forall j, k$	`predict prc*, mu`
Probabilities of positive outcome, marginal on class	
$P(Y1 = 1)$	`predict p1, mu outcome(1) pmarginal`
$P(Yj = 1) \forall j$	`predict p*, mu pmarginal`
Prior probability of class membership, $P(Ck)$	
$P(\mathbf{Y} \in Ck)$	`predict classpr*, classpr`
Posterior probability of class membership, (Bayes formula)	
$P(\mathbf{Y} \in C_k \| \mathbf{Y} = \mathbf{y})$	`predict classpostpr*, classposteriorpr`

To interpret the classes, we could compare the mean of the (counter-factual) conditional probabilities for each answer on each class; (the ones we get with **predict** by default) **estat lcmean** will do that.

```
. estat lcmean

Latent class marginal means                     Number of obs     =        216
```

	Margin	Delta-method Std. Err.	[95% Conf.	Interval]
1				
accident	.7135879	.0403588	.6285126	.7858194
play	.3296193	.0496984	.2403572	.4331299
insurance	.3540164	.0485528	.2655049	.4538042
stock	.1323726	.0383331	.0734875	.2268872
2				
accident	.9931933	.0253243	.0863544	.9999956
play	.9397644	.0659957	.6135685	.9935191
insurance	.9265309	.0656538	.6557086	.9881667
stock	.769132	.0952072	.5380601	.9050206

“marginal means” on the title refers to means averaged over the observations, but they are conditional on the class.

The probability of giving an universalistic response for each question is higher in group 2 than in group 1.

Also, we compute the predicted probabilities for each class.

Prior probabilities are the ones predicted by the logistic model for the latent class, which (with no covariates) will have no variations across the data.

```
. predict classpr*, classpr

. summ classpr*
```

Variable	Obs	Mean	Std. Dev.	Min	Max
classpr1	216	.7207538	0	.7207538	.7207538
classpr2	216	.2792462	0	.2792462	.2792462

This is an estimator of the population expected means for these variables. These estimates, and their confidence intervals can be obtained with **estat lcprob**.

```
. estat lcprob

Latent class marginal probabilities             Number of obs     =        216
```

	Margin	Delta-method Std. Err.	[95% Conf.	Interval]
C				
1	.7207539	.0580926	.5944743	.8196407
2	.2792461	.0580926	.1803593	.4055257

Stata provides some tools to evaluate goodness of fit:

```
. estat lcgof
```

Fit statistic	Value	Description
Likelihood ratio		
chi2_ms(6)	2.720	model vs. saturated
p > chi2	0.843	
Information criteria		
AIC	1026.935	Akaike´s information criterion
BIC	1057.313	Bayesian information criterion

Model with covariates: Geometry dataset [1]

Variables **pyit1** and **pyit2** contains binary responses for two Pythagorean test; **alg** is a score for a test on algebra. We fit three different models.

```
. use algebra, clear

. list in 1/5
```

	alg_sc~e	pyit1	pyit2	freq
1.	0	0	0	61
2.	0	0	1	24
3.	0	1	0	9
4.	0	1	1	6
5.	1	0	0	92

```
. expand freq
(1,213 observations created)
```

[1] (see Hagenaars and McCutcheon, 2002)

Model 1: two classes are determined by the binary variables **pyit1** and **pyit2**

```
. gsem (pyit1 pyit2 <-, logit), lclass(C 2) )
```

Model 2: two classes are determined by the binary variables **pyit1** and **pyit2**, and variable **alg** might contain helpful information to identify those groups

```
. gsem (pyit1 pyit2 <-, logit) (C <- alg), lclass(C 2)
```

Model 3: two classes are determined by the regressions of **pyit1** and **pyit2**, on variable **alg**; We are accounting not only for variations on the response among groups, but also on how this reponse relates to the covariate.

```
. gsem (pyit1 pyit2 <- alg, logit) , lclass(C 2) )
```

```
gsem (pyit1 pyit2 <-, logit), lclass(C 2) startvalues(randomid, draws(5)
seed(23))

. estat lcmean, vsquish
```

Latent class marginal means Number of obs = 1,241

	Margin	Delta-method Std. Err.	[95% Conf.	Interval]
1				
pyit1	.7707281	142.2577	0	1
pyit2	.8156159	247.4665	0	1
2				
pyit1	.1721594	253.6474	0	1
pyit2	.2158945	146.3729	0	1

```
. estat lcprob,vsquish
```

Latent class marginal probabilities Number of obs = 1,241

	Margin	Delta-method Std. Err.	[95% Conf.	Interval]
C				
1	.506648	241.258	0	1
2	.493352	241.258	0	1

```
gsem (pyit1 pyit2 <-, logit) (C <- alg), lclass(C 2)

. estat lcmean

Latent class marginal means                     Number of obs     =      1,241
```

	Margin	Delta-method Std. Err.	[95% Conf.	Interval]
1				
pyit1	.1985894	.0236409	.1562666	.2489921
pyit2	.3404315	.0202552	.3019188	.3811744
2				
pyit1	.9923852	.0292546	.0619459	.9999961
pyit2	.8545888	.0270487	.7932187	.9000403

```
. estat lcprob

Latent class marginal probabilities             Number of obs     =      1,241
```

	Margin	Delta-method Std. Err.	[95% Conf.	Interval]
C				
1	.6512534	.0237176	.6034547	.6961911
2	.3487466	.0237176	.3038089	.3965453

```
gsem (pyit1 pyit2 <- alg, logit) , lclass(C 2) startvalues(randomid,
draws(5) seed(15))
```

```
. estat lcmean
```

Latent class marginal means   Number of obs   =   1,241

	Margin	Delta-method Std. Err.	[95% Conf.	Interval]
1				
pyit1	.5846306	.0193834	.5462094	.6220497
pyit2	.6409796	.0220191	.596784	.682905
2				
pyit1	.0633972	.0363614	.0199756	.1835298
pyit2	.0618345	.036141	.0190673	.1826642

```
. estat lcprob
```

Latent class marginal probabilities   Number of obs   =   1,241

	Margin	Delta-method Std. Err.	[95% Conf.	Interval]
C				
1	.7922178	.0294795	.728562	.844139
2	.2077822	.0294795	.155861	.271438

From model 2, we see that variable **alg** helps us to identify groups with different scores; The identification of the 'high' and 'low' score groups doesn't improve when accounting for their dependence on **alg**, suggesting there might be a different interpretation for the last model.

Additional remarks:

- LCA order might vary when we vary the starting values.
- Fit the model repeateadly with different starting values to avoid local maxima.
- The conditional independence assumption might not be true; a way to account for dependence is to incorporate more discrete latent variables. Another way, for categorical responses, is to generate new categories with combinations of the correlated variables.
- The conditional independence is not necessary for Gaussian variables, we can include correlations among them.

Concluding remarks:

- **gsem** offers a framework where we can fit models accounting for latent classes.
- Responses might take one or more of the distributions supported by **gsem**.
- We can fit non-parametric models by using only binary or categorical responses. We can also parameterize the responses and the probabilities of class membership by introducing covariates.
- Discrete latent variables might have more than two groups, and more than one latent variable also might be included.
- Some latent class models are a special case of finite mixture models. The **fmm** prefix allows us to fit finite mixture models for a variety of distributions.