Stat 8053, Fall 2013: Latent Class Analysis

Generalities

j	level of the latent factor, $j = 1, \ldots, K$
i	the ith manifest variable, $i = 1, \ldots, p$, also a factor with 2 levels in this handout
h	subject number, $h = 1, \ldots, n$
γ_j	fraction of population in level j of the latent factor
π_{ij}	probability of correct response on manifest variable i for an individual in latent category j
$\mathbf{x}_h$	$p \times 1$ response vectors for the p manifest variables for subject h

1. We observe p *manifest variables* $x_1, \ldots, x_p$. Each of the manifest variables is categorical, such as the response to a question, the presence/absence of a state, and so on. Suppose the j-th variable has m_j levels. *Example:* The manifest variables are p exam questions. Question j has $m_j = 2$ possible responses, of which one is correct and the other is incorrect.

2. We hypothesize the existence of a latent categorical variable y with $K \geq 2$ levels. The big idea is: *Each level of y defines a population of subjects. Conditioning on level of the latent variable, the manifest variables are independent*: $(x_1 \perp\!\!\!\perp x_2 \cdots \perp\!\!\!\perp x_p)|y$. This replaces the possibly complex association among the manifest variables with a simple classification. *Example*: Exam takers are of 2 types: those who have mastered the material and those who have not mastered the material. Students who have mastered the material may still get some of the questions wrong, but their responses are independent. Similarly, for the other class we have complete independence. We would further expect that the probability of correct response on a question depends on both the difficulty of the question and on class membership, so the students who have mastered the material would have higher probabilities than those who have not mastered it.

This approach is certainly not appropriate for understanding the structure in all contingency tables, depending on the x_j. This model seems most appropriate when the manifest variables x_i are measuring the aspects of the same thing. As a different example, people surveyed in a questionnaire may answer a set of p questions about their attitudes toward about various government programs, and we could hypothesize that subjects can be classified into a number of levels of a latent variable so that given their class membership responses are independent.

The Set Up

1. The latent variable y has K classes, and suppose for an individual in the population the prior probability of class membership is $P(y = j) = \gamma_j$, with $\sum \gamma_j = 1$.

2. For simplicity here, suppose that all the manifest variables are binary. The extension to manifest variables with more categories requires only a little more notation. Let π_{ij} be the probability that a person in latent category j gives a correct response on manifest variable i. If $x_i = 1$ for a correct response and $x_i = 0$ for an incorrect response, then according to the model

$$f(\mathbf{x}|y = j) = \prod_{i=1}^{p} \pi_{ij}^{x_i}(1 - \pi_{ij})^{1-x_i} \tag{1}$$

3. The unconditional distribution is

$$f(\mathbf{x}) = \sum_{j=1}^{K} \left\{ \gamma_j \prod_{i=1}^{p} \pi_{ij}^{x_i}(1 - \pi_{ij})^{1-x_i} \right\} \tag{2}$$

Suppose we sample at random from a population, which implies that on average the fraction of subjects in class j estimates $\gamma)j$. With this crucial assumption, all we need is (2). If we have n observations, then the log-likelihood is

$$\text{logL}(\theta) = \log \left(\prod_{h=1}^{n} f(\mathbf{x}_h) \right) \tag{3}$$

where θ consists of the γs ($K-1$ free parameters) and the $\{\pi_{ij}\}$ (Kp parameters). From this we can find estimates using an appropriate maximizing method like Newton-Raphson, or the EM procedure to be outlined shortly.

We may wish to assign observations to the latent classes. Let h index the current subject to be assigned. For $j = 1, \ldots, K$, by Bayes' theorem

$$\begin{aligned} h(j|\mathbf{x}_h) &= \frac{f(\mathbf{x}_h|y = j)}{f(\mathbf{x}_h)} \\ &= \frac{\gamma_j \left[\prod \pi_{ij}^{x_{hi}}(1 - \pi_{ij})^{1-x_{hi}} \right]}{f(\mathbf{x}_h)} \end{aligned} \tag{4}$$

To make a "hard" assignment, assign i to level j if $j = \arg\max [h(j|\mathbf{x}_i)]$, with estimates replacing parameters.

ML estimation

The log-likelihood (3) can be maximized directly using Newton-Raphson subject the the constraint that $\sum \hat{\gamma}_j = 1$. Because this is a special case of more complex models, an EM algorithm is used. This is an alternative algorithm with repeated applications of an Expectation step and a Maximization step; a student will present later in the course a talk on this algorthim. We differentiate logL with respect to the parameters and solve to get estimating

equations. We skip the details, and get the following.

$$\hat{\gamma}_j = \sum_{h=1}^{n} h(j|\mathbf{x}_h)/n, i = 1, \ldots, K \quad (5)$$

$$\hat{\pi}_{ij} = \frac{1}{n\hat{\gamma}_j} \sum_{h=1}^{n} x_{ih} h(j|\mathbf{x}_h), i = 1, \ldots, K, j = 1, \ldots, p \quad (6)$$

Equation (5) is the estimated fraction of observations allocated to latent class j, while (6) is the fraction of correct answers to manifest variable i in latent class j.

Here is the iterative procedure:

1. Choose starting values for the conditional probabilities (4).
2. Use (5) and (6) to update parameter estimates. This is the M step; in this case there is a closed-form maximization so this is really fast.
3. Substitute the estimates into (4) to get improved estimates of the conditional probabilities. This is imputing values for the unobserved allocation of individuals to levels of the latent variable, and is the E step.
4. Repeat steps 2–3 until convergence.

Tidbits

1. The EM algorithm generally has linear convergence, meaning that many, hundreds or even thousands, of iterations can be required. In this example the iterations all have closed form and are fast.
2. logL need not be unimodal, and so local maxima are possible. Several random starts are recommended.
3. SEs are from the inverse of the negative Fisher Information, by evaluating minus the second derivatives of the logL evaluated at the mle and inverting.
4. Likelihood ratio tests are available for the number of classes. If a model fits the likelihood ratio test is almost the same as testing $(x_1 \perp\!\!\!\perp \cdots \perp\!\!\!\perp x_p)|\hat{\mathbf{y}}$, where $\hat{\mathbf{y}}$ is the "hard allocation of units to levels of the latent variable. Presumably other model selection criteria exist/could be developed.
5. All this easily extends to items with more than 2 categories for the manifest variables.
6. **Extends to regression**, in which $\gamma_j = \gamma_j(\mathbf{z}_h)$ for some vector of covariates $\mathbf{z}_h$.

Example 1

To get started, I'll work with simulated data and set

- $K = 2$, $\boldsymbol{\gamma} = (0.7, .03)'$, so 70% of subjects on average are in class 1.
- Set $p = 4$ manifest variables.
- All the $\pi_{i1} = 0.9$ and all the $\pi_{i2} = 0.3$, so for subjects in class 1 the total number of correct answers is $\text{Bn}(4, 0.9)$ and in the other class the number of correct answers is $\text{Bn}(4, 0.3)$.

```
gam <- c(0.7, 0.3)
pr <- c(0.9, 0.3)
p <- 4
```

Next, set $n = 400$, and generate the "data":

```
n <- 400
set.seed(567)
class <- ifelse( runif(n) > gam[1], 2, 1)   #  assign rows to classes
data <- matrix(runif(n*p), nrow=n, ncol=p) # initialize
data <- t(apply(cbind(class, data), 1, function(x){  # by roww
             prob <- pr[x[1]]
             ifelse(x[-1] > prob, 2, 1)
             }))
data <- as.data.frame(data)
data$class <- class
(data.table <- as.data.frame(xtabs( ~ V1 +V2 +V3 +V4, data)))
```

```
   V1 V2 V3 V4 Freq
1   1  1  1  1  197
2   2  1  1  1   19
3   1  2  1  1   22
4   2  2  1  1    9
5   1  1  2  1   22
6   2  1  2  1    6
7   1  2  2  1   11
8   2  2  2  1   10
9   1  1  1  2   20
10  2  1  1  2    8
11  1  2  1  2   11
12  2  2  1  2   12
13  1  1  2  2    3
14  2  1  2  2    5
15  1  2  2  2   11
16  2  2  2  2   34
```

Fit the model of complete independence to the cell frequencies:

```
p1 <- glm(Freq ~ ., family=poisson, data=data.table)
data.frame(deviance=p1$deviance, df=p1$df.residual,
    pval=1-pchisq(p1$deviance,p1$df.residual))
```

```
  deviance df pval
1    270.6 11    0
```

```
library(car)
Anova(update(p1, ~ (.)^2))
```

```
Analysis of Deviance Table (Type II tests)

Response: Freq
      LR Chisq Df Pr(>Chisq)
V1         98.2  1    < 2e-16
V2         65.8  1    4.9e-16
V3        100.3  1    < 2e-16
V4         96.1  1    < 2e-16
V1:V2      15.1  1    0.00010
V1:V3      11.6  1    0.00067
V1:V4      19.3  1    1.1e-05
V2:V3      26.9  1    2.1e-07
V2:V4      27.1  1    1.9e-07
V3:V4       4.6  1    0.03253
```

At least all the 2 fi's are needed. If we look at the (unobservable) 5-way table,

```
big.data.table <- as.data.frame(xtabs( ~ V1 +V2 +V3 +V4 + class, data))
m1 <- glm(Freq ~ V1 + V2 + V3 + V4, poisson, big.data.table,
           subset=class==1) # class 1
m2 <- update(m1, subset=class==2) # class 2
m3 <- glm(Freq ~ (V1 + V2 + V3 + V4)*class, poisson,
           big.data.table) # conditional indep
data.frame(deviance=d <- c(m1$deviance, m2$deviance, m3$deviance),
            df=f <- c(m1$df.residual, m2$df.residual, m3$df.residual),
            pval = 1-pchisq(d, f)
            )
```

```
  deviance df    pval
1    13.11 11 0.28641
2    17.81 11 0.08614
3    30.91 22 0.09787
```

Next, we use the `poLCA` package to do LCA (Linzer and Lewis, 2011). Although the sufficient statistics are the cell counts, the `poLCA` software requires one line per observation, not sufficient statistics. For this example the data.frame `data` is in this format, but if you only have the contingency table you can convert it to the correct form with

```
freq.to.long <- function(x, freq){x[rep(1:length(freq), freq), ]}
data1 <- freq.to.long(data.table, data.table$Freq)
```

We first fit the one-class model, which is the same as the model `p1` fit above. You may need to install the `poLCA` package.

```
library(poLCA)
m1 <- poLCA(cbind(V1, V2, V3, V4) ~ 1, data=data, nclass=1)

Conditional item response (column) probabilities,
 by outcome variable, for each class (row)

$V1
           Pr(1)  Pr(2)
class 1:  0.7425 0.2575

$V2
          Pr(1) Pr(2)
class 1:    0.7   0.3

$V3
          Pr(1) Pr(2)
class 1:  0.745 0.255

$V4
          Pr(1) Pr(2)
class 1:   0.74  0.26

Estimated class population shares
 1

Predicted class memberships (by modal posterior prob.)
 1

=========================================================
Fit for 1 latent classes:
=========================================================
number of observations: 400
number of estimated parameters: 4
```

```
residual degrees of freedom: 11
maximum log-likelihood: -928.8

AIC(1): 1866
BIC(1): 1882
G^2(1): 270.6 (Likelihood ratio/deviance statistic)
X^2(1): 637.8 (Chi-square goodness of fit)
```

The left-side of the formula is a list of all the manifest variables. The right-side is not yet used, so the specification `~ 1` is appropriate. The `nclass` argument specifies K. Unlike most R functions, this one produces printed output unless you suppress it with `verbose=FALSE`. The G^2 reported is identical to the G^2 for complete independence computed in `p1` above.

Fit the more interesting 2 class model. The function has its own algorithm for choosing starting values, and we will use the default method. To avoid finding a local maximum, the argument `nrep=5` tells the program to repeat `nrep` times, each time with different starting values, and keep the fit with the highest likelihood.

```
m2 <- poLCA(cbind(V1, V2, V3, V4) ~ 1, data=data, nclass=2, nrep=5)
```

```
Model 1: llik = -797.2 ... best llik = -797.2
Model 2: llik = -797.2 ... best llik = -797.2
Model 3: llik = -797.2 ... best llik = -797.2
Model 4: llik = -797.2 ... best llik = -797.2
Model 5: llik = -797.2 ... best llik = -797.2
Conditional item response (column) probabilities,
 by outcome variable, for each class (row)

$V1
           Pr(1)  Pr(2)
class 1:  0.3124 0.6876
class 2:  0.9095 0.0905

$V2
           Pr(1)  Pr(2)
class 1:  0.1727 0.8273
class 2:  0.9048 0.0952

$V3
           Pr(1)  Pr(2)
class 1:  0.3424 0.6576
class 2:  0.9014 0.0986

$V4
           Pr(1)  Pr(2)
```

```
class 1:  0.3023 0.6977
class 2:  0.9100 0.0900

Estimated class population shares
 0.2797 0.7203

Predicted class memberships (by modal posterior prob.)
 0.3 0.7

=========================================================
Fit for 2 latent classes:
=========================================================
number of observations: 400
number of estimated parameters: 9
residual degrees of freedom: 6
maximum log-likelihood: -797.2

AIC(2): 1612
BIC(2): 1648
G^2(2): 7.355 (Likelihood ratio/deviance statistic)
X^2(2): 7.533 (Chi-square goodness of fit)
```

The estimate $\hat{\gamma}_1 = 0.28$ nearly reproduces the population value from the simulation of $\gamma_1 = 0.3$. The input values for the probabilities in the simulation were $(.3, .7)$ for one class and $(.1, .9)$ for the other, nearly matching the estimates for most of the manifest variables. This fitted model is almost equivalent to fitting a 2^5 table with additional dimension determined by the predicted class, and fitting the model of conditional independence given the predicted class $\hat{\mathbf{y}}$:

```
data.table1 <- as.data.frame(xtabs( ~ m2$predclass + V1 +V2 +V3 +V4, data))
p2 <- glm(Freq ~ m2.predclass*(V1 + V2 + V3 + V4),
          family=poisson, data=data.table1, subset=Freq > 0)
c(deviance=p2$deviance, df=p2$df.residual)
```

```
deviance       df
   11.11     6.00
```

Let's see how well this the predicted class matched the actual class (this is a simulation, after all):

```
xtabs(~ class + m2$predclass)
```

```
     m2$predclass
class   1   2
    1  14 271
    2 106   9
```

The order of the predicted classes is reversed from the input. Finally, here is a three class solution:

```
 m3 <- poLCA(cbind(V1, V2, V3, V4) ~ 1, data=data, nclass=3, nrep=5)

Model 1: llik = -795.8 ... best llik = -795.8
Model 2: llik = -794.8 ... best llik = -794.8
Model 3: llik = -794.8 ... best llik = -794.8
Model 4: llik = -795.8 ... best llik = -794.8
Model 5: llik = -795.8 ... best llik = -794.8
Conditional item response (column) probabilities,
 by outcome variable, for each class (row)

$V1
           Pr(1)  Pr(2)
class 1:  0.5706 0.4294
class 2:  0.9403 0.0597
class 3:  0.1601 0.8399

$V2
           Pr(1)  Pr(2)
class 1:  0.4361 0.5639
class 2:  0.9522 0.0478
class 3:  0.0526 0.9474

$V3
           Pr(1)  Pr(2)
class 1:  0.6277 0.3723
class 2:  0.9221 0.0779
class 3:  0.1456 0.8544

$V4
           Pr(1)  Pr(2)
class 1:  0.5828 0.4172
class 2:  0.9372 0.0628
class 3:  0.1294 0.8706

Estimated class population shares
 0.2682 0.6054 0.1264

Predicted class memberships (by modal posterior prob.)
 0.2425 0.645 0.1125

=========================================================
```

```
Fit for 3 latent classes:
=========================================================
number of observations: 400
number of estimated parameters: 14
residual degrees of freedom: 1
maximum log-likelihood: -794.8

AIC(3): 1618
BIC(3): 1674
G^2(3): 2.547 (Likelihood ratio/deviance statistic)
X^2(3): 2.353 (Chi-square goodness of fit)
```

```
c(dev_change=m2$Gsq-m3$Gsq, df=m2$resid.df-m3$resid.df,
      pval=pchisq(m2$Gsq-m3$Gsq, m2$resid.df-m3$resid.df))
```

```
dev_change         df       pval
    4.8085     5.0000     0.5603
```

As expected, the 3 class model offers no improvement over the two class model.

Student Cheating

This example has 319 observations on 4 manifest variables relating to student cheating, either no (= 1) or yes (= 2). A fifth variable in the data file is `GPA`, the student's overall grade point average.

```
data(cheating)
str(cheating)
```

```
'data.frame':        319 obs. of  5 variables:
 $ LIEEXAM : num  1 1 1 1 1 1 1 1 1 1 ...
 $ LIEPAPER: num  1 1 1 1 1 1 1 1 1 1 ...
 $ FRAUD   : num  1 1 1 1 1 1 1 1 1 1 ...
 $ COPYEXAM: num  1 1 1 1 1 1 1 1 1 1 ...
 $ GPA     : int  NA NA NA NA 1 1 1 1 1 1 ...
```

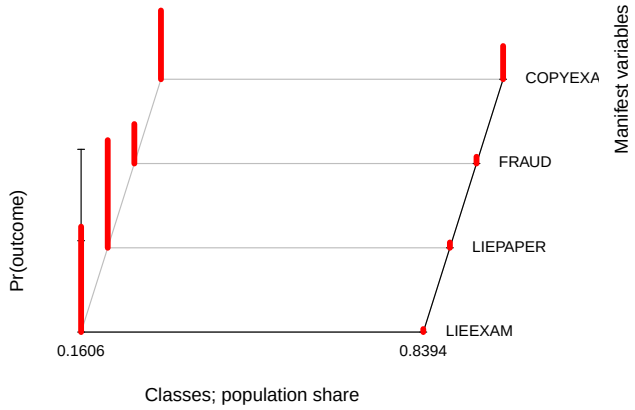
```
f <- cbind(LIEEXAM, LIEPAPER, FRAUD, COPYEXAM) ~ 1
ch2 <- poLCA(f, cheating, nclass=2, nrep=5)
```

```
Model 1: llik = -440 ... best llik = -440
Model 2: llik = -440 ... best llik = -440
Model 3: llik = -440 ... best llik = -440
Model 4: llik = -440 ... best llik = -440
Model 5: llik = -440 ... best llik = -440
```

```
Conditional item response (column) probabilities,
 by outcome variable, for each class (row)

$LIEEXAM
           Pr(1)  Pr(2)
class 1:  0.4231 0.5769
class 2:  0.9834 0.0166

$LIEPAPER
           Pr(1)  Pr(2)
class 1:  0.4109 0.5891
class 2:  0.9708 0.0292

$FRAUD
           Pr(1)  Pr(2)
class 1:  0.7840 0.2160
class 2:  0.9629 0.0371

$COPYEXAM
           Pr(1)  Pr(2)
class 1:  0.6236 0.3764
class 2:  0.8181 0.1819

Estimated class population shares
 0.1606 0.8394

Predicted class memberships (by modal posterior prob.)
 0.1693 0.8307

=========================================================
Fit for 2 latent classes:
=========================================================
number of observations: 319
number of estimated parameters: 9
residual degrees of freedom: 6
maximum log-likelihood: -440

AIC(2): 898.1
BIC(2): 931.9
G^2(2): 7.764 (Likelihood ratio/deviance statistic)
X^2(2): 8.323 (Chi-square goodness of fit)
```

Comparing G^2 to its df, the two-class model seems to fit adequately.

```
plot(ch2)
```



The two classes are called "cheaters" and "not-cheaters".

```
xtabs(~ ch2$predclass + rowSums(cheating[, -5]))
```

```
             rowSums(cheating[, -5])
ch2$predclass   4   5   6   7   8
            1   0  23  20   9   2
            2 207  53   5   0   0
```

The classes are not the same as the sum!

Regression

Regression can allow γ_j to depend on predictors, in this case `GPA`. We first do logistic regression as if classes were observed not estimated.

```
summary(g1 <- glm(I(ch2$predclass -1) ~ GPA, binomial, cheating))$coef
```

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.6052     0.3170   1.909 0.056245
GPA           0.4628     0.1472   3.144 0.001668
```

Alternatively estimate latent classes and coefficients simultaneously. When $K = 2$, we can use a logistic model,

$$\log(\gamma_2/\gamma_1) = \alpha_0 + \boldsymbol{\alpha}'\mathbf{z}$$

When $K > 2$, poLCA uses the baseline category multinomial logistic model,

$$\log(\gamma_i/\gamma_1) = \alpha_{0i} + \boldsymbol{\alpha}_i'\mathbf{z},\ i = 2, \ldots, K$$

```
f2 <- cbind(LIEEXAM, LIEPAPER, FRAUD, COPYEXAM) ~ GPA
ch2c <- poLCA(f2, cheating, nclass=2, nrep=5)
```

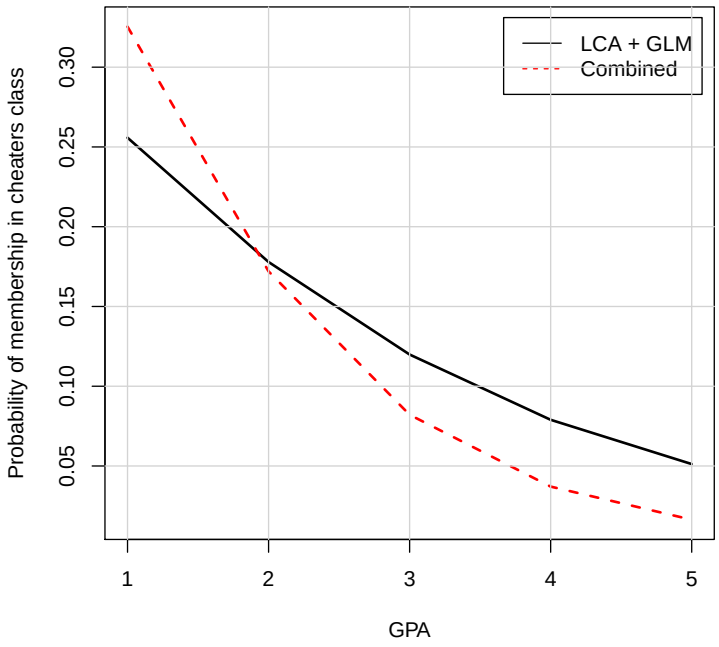
```
Model 1: llik = -429.6 ... best llik = -429.6
Model 2: llik = -429.6 ... best llik = -429.6
Model 3: llik = -429.6 ... best llik = -429.6
Model 4: llik = -429.6 ... best llik = -429.6
Model 5: llik = -429.6 ... best llik = -429.6
Conditional item response (column) probabilities,
 by outcome variable, for each class (row)

$LIEEXAM
           Pr(1)  Pr(2)
class 1:  0.9903 0.0097
class 2:  0.4389 0.5611

$LIEPAPER
           Pr(1)  Pr(2)
class 1:  0.9647 0.0353
class 2:  0.4858 0.5142

$FRAUD
           Pr(1)  Pr(2)
class 1:  0.9655 0.0345
class 2:  0.7850 0.2150

$COPYEXAM
           Pr(1)  Pr(2)
class 1:  0.8257 0.1743
class 2:  0.5925 0.4075

Estimated class population shares
 0.8219 0.1781

Predicted class memberships (by modal posterior prob.)
 0.8508 0.1492

=========================================================
Fit for 2 latent classes:
=========================================================
```

```
2 / 1
            Coefficient  Std. error  t value  Pr(>|t|)
(Intercept)      0.1134      0.5099    0.222     0.833
GPA             -0.8425      0.2813   -2.995     0.030
=========================================================
number of observations: 315
number of estimated parameters: 10
residual degrees of freedom: 5
maximum log-likelihood: -429.6

AIC(2): 879.3
BIC(2): 916.8
X^2(2): 8.642 (Chi-square goodness of fit)

ALERT: estimation algorithm automatically restarted with new initial values
```

See the help page for `poLCA.reorder` on how to get `poLCA` to reorder the latent classes.

For these data, you cannot fit with $K > 2$ because the number of parameters would then exceed the number of cells with data.

```
table(cheating$GPA)

  1   2   3   4   5
100 104  48  34  29

GPAmat <- cbind(1, c(1:5))
exb <- exp(GPAmat %*% ch2c$coeff)
matplot(c(1:5),
   cbind(1-predict(g1, newdata=data.frame(GPA=1:5), type="response") , exb/(1+exb)),
   xlab="GPA", type="l", lwd=2, lty=1:2, col=1:2,
   ylab="Probability of membership in cheaters class")
legend("topright", c("LCA + GLM", "Combined"), lty=1:2, col=1:2,inset=0.02)
grid(lty=1)
```



Effects Plots

Effects plots for the covariates like `GPA` above will be available in the next version of the `efffects` package that you can get using

```
install.packages("effects", repos="http://r-forge.r-project.org")
```

References

1. Bartholomew, D., Knott, M. and Moustaki, I. (2011). *Latent Variable Models and Factor Analysis*, 3rd edition, Wiley. More mathematics, and very wide coverage.

2. Collins, L. and Lanza, S. (2010). *Latent Class and Latent Transition Analysis*, Wiley. Less mathematics, more application.

3. Linzer, D. and Lewis, J. (2011). `poLCA`: An R Package for Polytomous Variable Latent Class Analysis, *Journal of Statistical Software, 42(10)*, 1-29. `http://www.jstatsoft.org/v42/i10/`.