36-720: Latent Class Models

Brian Junker

October 17, 2007

- Latent Class Models and Simpson's Paradox
- Latent Class Model for a Table of Counts
- An Intuitive E-M Algorithm for Fitting Latent Class Models
- Deriving the E-M Algorithm
- Example
- Issues and Warnings

For more reading, see:

Bartholomew, D. J. & Knott, M. M. (1999). *Latent Variable Models and Factor Analysis.* Oxford Univ Press.

Hagenaars, J. A. & McCutcheon, A. L. (2003). *Applied Latent Class Analysis.* Cambridge Univ Press.

McLachlan, G. & Peel, D. (2000). *Finite Mixture Models.* NY: Wiley.

Latent Class Models and Simpson's Paradox

Simpson's Paradox: *If you collapse a table along one dimension, you may introduce new dependencies in the data.*

$$
\begin{array}{cc|cc}
 & & \multicolumn{2}{c}{Y_2} \\
 & & 0 & 1 \\
\hline
 & 0 & n_{000} & n_{010} \\
Y_1 & 1 & n_{100} & n_{110}
\end{array}
\quad + \quad
\begin{array}{cc|cc}
 & & \multicolumn{2}{c}{Y_2} \\
 & & 0 & 1 \\
\hline
 & 0 & n_{001} & n_{011} \\
Y_1 & 1 & n_{101} & n_{111}
\end{array}
\quad \Rightarrow \quad
\begin{array}{cc|cc}
 & & \multicolumn{2}{c}{Y_2} \\
 & & 0 & 1 \\
\hline
 & 0 & n_{00+} & n_{01+} \\
Y_1 & 1 & n_{10+} & n_{11+}
\end{array}
$$

$$(OR \mid Z = 0) = 1 \qquad (OR \mid Z = 1) = 1 \qquad OR > 1$$

Latent Class Model: *Given a table of counts exhibiting dependence, find a dimension along which you can separate observations in the table to make the dependence go away!* (Simpson in reverse…)

$$
\begin{array}{cc|cc}
 & & \multicolumn{2}{c}{Y_2} \\
 & & 0 & 1 \\
\hline
 & 0 & n_{00} & n_{01} \\
Y_1 & 1 & n_{10} & n_{11}
\end{array}
\quad \Rightarrow \quad
\begin{array}{cc|cc}
 & & \multicolumn{2}{c}{Y_2} \\
 & & 0 & 1 \\
\hline
 & 0 & n_{000} & n_{010} \\
Y_1 & 1 & n_{100} & n_{110}
\end{array}
\quad + \quad
\begin{array}{cc|cc}
 & & \multicolumn{2}{c}{Y_2} \\
 & & 0 & 1 \\
\hline
 & 0 & n_{001} & n_{011} \\
Y_1 & 1 & n_{101} & n_{111}
\end{array}
$$

$$OR > 1 \qquad (OR \mid Z = 0) = 1 \qquad (OR \mid Z = 1) = 1$$

The Many Siblings of Latent Class Analysis (LCA)

- To the extent that we are "assigning" some observations to the group "$Z = 0$" and some to the group "$Z = 1$", LCA is a kind of *model-based cluster analysis* and/or an *unsupervised classifier* for multivariate discrete data.

- We will see below that LCA basically models the distribution of $\{n_{00}, n_{01}, n_{10}, n_{11}\}$ as the mixture of two log-linear models (one for $Z = 0$ and one for $Z = 1$). Therefore LCA can also be viewed as a kind of *finite mixture modeling*.
  - Finite mixture models for continuous response data are perhaps more familiar: *mixture-of-normals density estimators, kernel density estimators*, etc.
  - Other finite mixtures for discrete data, such as *zero-inflated Poisson models*, are also closely related to LCA.

- *Bayes Network models* for discrete data with discrete hidden nodes may also be viewed as a kind of constrained LCA (each latent class is determined by a unique set of values for the hidden nodes).
  - These models are basic choices in AI/expert systems, computer-based tutoring, cognitive diagnosis, etc.

Latent Class Model for a Table of Counts

Latent class models are often used with item response data (like the Rasch model) so we will consider that case as well. Once again let

$$y_{ij} = \begin{cases} 1, & \text{if student } i \text{ got question } j \text{ correct} \\ 0, & \text{else} \end{cases}$$

say, for $i = 1, \ldots, N$ students and $j = 1, \ldots, J$ questions.

We begin by considering the 2^J table

$$\{n_{\underline{y}} : \text{ as } \underline{y} \text{ ranges over all } 2^J \text{ possible patterns } (y_1, \ldots, y_J)\}$$

directly.

The latent class model says that there are W layers, or latent classes, that we can split this table into:

$$
\begin{aligned}
n_{\underline{y}}^{(1)} &= t_{\underline{y}}^{(1)} n_{\underline{y}} \\
n_{\underline{y}}^{(2)} &= t_{\underline{y}}^{(2)} n_{\underline{y}} \\
&\vdots \\
n_{\underline{y}}^{(W)} &= t_{\underline{y}}^{(W)} n_{\underline{y}}
\end{aligned}
$$

where the $t_{\underline{y}}^{(w)}$ is simply the proportion of observations in cell $\underline{y}$ that get assigned to layer, or latent class, w (so $\sum_w t_{\underline{y}}^{(w)} = 1$, for each $\underline{y}$; and $t_{\underline{y}}^{(w)}$ may be different for different $\underline{y}$).

Within each latent class we assume the model of independence holds:

$$
\begin{aligned}
n_{\underline{y}}^{(w)} &\sim Multinom(p_{\underline{y}}^{(w)}, N_w) \\
\log m_{\underline{y}}^{(w)} &= \alpha^{(w)} + \beta_1^{(w)} y_1 + \cdots + \beta_J^{(w)} y_J
\end{aligned}
$$

where $N_w = \sum_{\underline{y}} n_{\underline{y}}^{(w)}$, and $m_{\underline{y}}^{(w)} = N_w p_{\underline{y}}^{(w)}$.

An Intuitive E-M Algorithm for Fitting Latent Class Models

One way to fit a latent class model is with a kind of E-M algorithm:

Initialization: Make an initial guess $\tilde{n}_{\underline{y}}^{(w)}$ for the counts in the w^{th} layer.

M step: Fit the log-linear model of independence to each table $\tilde{n}_{\underline{y}}^{(w)}$, using `glm()`, obtaining fitted values $m_{\underline{y}}^{(w)}$;

E step: Recalculate

$$\tilde{t}_{\underline{y}}^{(w)} = \frac{m_{\underline{y}}^{(w)}}{m_{\underline{y}}^{(1)} + m_{\underline{y}}^{(2)} + \cdots + m_{\underline{y}}^{(W)}} \quad , \quad \tilde{n}_{\underline{y}}^{(w)} = \tilde{t}_{\underline{y}}^{(w)} n_{\underline{y}}$$

Convergence check: If this iteration does not produce much change from the previous iteration, then stop. Otherwise start over, with the "M" step.

Note that all of the data $n_{\underline{y}}^{(w)}$ is missing, in every layer of the table along W.

- Clearly the "E" step above is imputing this missing data by first calculating the fitted probabilties $\tilde{t}_{\underline{y}}^{(w)}$ that $\underline{y}$ will be classified into class w, and then computing the expected number $n_{\underline{y}}^{(w)} = \tilde{t}_{\underline{y}}^{(w)} n_y$.
- The "M" step is just fitting the complete-data model (log-linear model of independence) to the imputed tables $n_{\underline{y}}^{(w)}$.
- Finally, any reasonable measure of fit can be used in the "Convergence check". One interesting possibility is to calculate expected cell counts $m_{\underline{y}} = \sum_w m_{\underline{y}}^{(w)}$ and then keep track of $G^2 = 2\sum_{\underline{y}} n_{\underline{y}} \log(n_{\underline{y}}/m_{\underline{y}})$ for the original table. When G^2 stops changing, we can quit the E-M algorithm.

Deriving the E-M Algorithm

Imitating our work for the Rasch model, we can see that the likelihood for examinee i in latent class w is

$$P[y_{i1}, \ldots, y_{iJ} \mid w, p_{i1}, \ldots, p_{iJ}] = \prod_{j=1}^{J} p_{wj}^{y_{ij}} (1 - p_{wj})^{y_{ij}}$$

If we treat w as a random effect (like we treated θ in the Rasch model) we get the mixture model

$$P[y_{i1}, \ldots, y_{iJ} \mid p_{i1}, \ldots, p_{iJ}, \lambda_1, \ldots, \lambda_W] = \sum_{w} \lambda_w \prod_{j=1}^{J} p_{wj}^{y_{ij}} (1 - p_{wj})^{y_{ij}}$$

where λ_w is the (prior) probability that an examinee belongs to class w. This equation is analogous to equation

$$P[y_{i1}, \ldots, y_{iJ} \mid \beta_1, \ldots, \beta_J; \sigma] = \int_{-\infty}^{\infty} \prod_{j=1}^{J} \frac{\exp\{y_{ij}(\theta_i - \beta_j)\}}{1 + \exp\{\theta_i - \beta_j\}} f(\theta_i \mid \sigma)\, d\theta_i$$

for the Rasch model.

The joint model for all examinees

$$P[\mathcal{Y} \mid p_{ij},\ \text{all } i, j; \lambda_1, \ldots, \lambda_W] = \prod_{i=1}^{N} \sum_{w} \lambda_w \prod_{j=1}^{J} p_{wj}^{y_{ij}} (1 - p_{wj})^{y_{ij}}$$

is kind of unwieldy but it simplifies if we introduce a *data augmentation variable* $z_{iw} = 1$ if examinee i is in latent class w and $z_{iw} = 0$ otherwise. With this the complete data likelihood becomes

$$P[\mathcal{Y}, \mathcal{Z} | \text{ all } p_{ij},\ \text{all} \lambda_w] = \prod_{i} \prod_{w} \left\{ \lambda_w \prod_{j} p_{wj}^{y_{ij}} (1 - p_{wj})^{1-y_{ij}} \right\}^{z_{iw}}$$

Now if we apply the “standard” E-M algorithm with

$$Q(\varphi, \varphi^{(k)}) = E[\log P(\mathcal{Y}, \mathcal{Z} | \varphi) \mid \varphi^{(k)}, \mathcal{Y}]$$

[where now $\varphi = (\text{all } p_{ij},\ \text{all } \lambda_w)$] we will eventually obtain the steps above, with $\tilde{t}_{\underline{y}_i}^{(w)} = P[z_{iw} = 1 \ \mid \underline{y} = \underline{y}_i]$.

Example

We previously analyzed the Stouffer-Toby data using graphical log-linear models and found the decomposable model

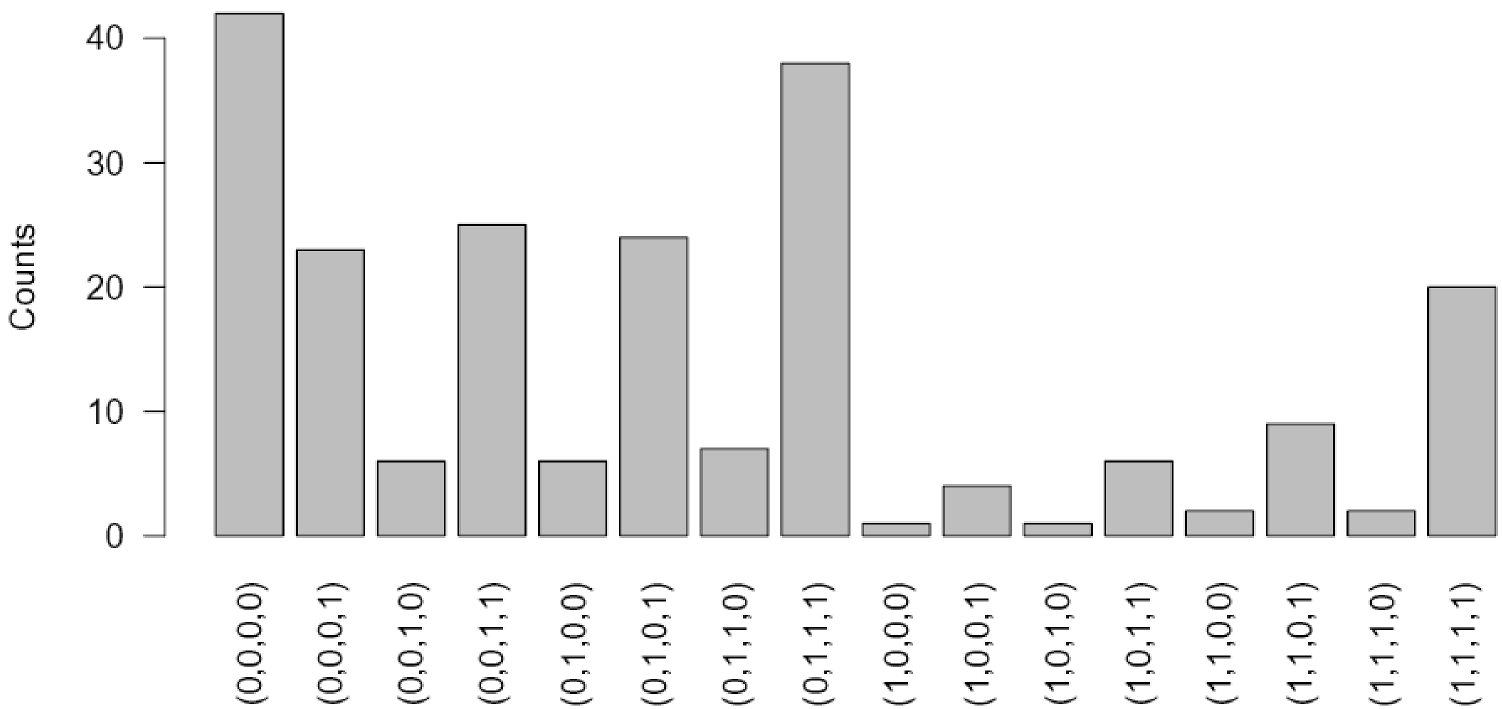
```
1 -- 2 -- 3
       \   |
        \ |
         4
```

Recall that the data consist of role-conflict responses of 216 respondents to four situations, with response “0” being “universalistic” and “1” being “particularistic”. Goodman (2002, Chapter 1 of *Applied Latent Class Analysis*, Hagenaars & McCutcheon, eds, NY: Cambridge Univ Press) considers a latent class analysis of the same data.

We will reproduce, with some modifications, Goodman’s analysis.

A pdf of Goodman’s chapter is included with these lecture notes.

We begin with a barplot of the actual counts in each cell:



- There is clearly dependence in the data: for example $P[y_2, y_3, y_4 | y_1 = 0] \neq P[y_2, y_3, y_4 | y_1 = 1]$.
- This also suggests perhaps some latent class structure, driven somehow by response to item 1.

An E-M algorithm to fit the LC model can be coded easily in R:

```
lcm <- function(data,tstar=runif(dim(data)[1]),
            fla1=n~., fla2=n~., tol=1, reps=100) {

  cells <- dim(data)[1]

  if (length(tstar)!=cells) {
    tstar <- rep(abs(tstar[1]),cells)
  }
  tstar <- tstar/sum(tstar)

  d1 <- data
  d2 <- data

  G1 <- 1e6
  G2 <- 1e6

  K <- 0
  err <- 1e6

  while ((err > tol)&&(K<reps)) {

    K <- K+1

    G1.old <- G1
    G2.old <- G2

    d1$n <- tstar*data$n
    d2$n <- (1-tstar)*data$n

    fit1 <- glm(fla1,data=d1,family=poisson)
    fit2 <- glm(fla2,data=d2,family=poisson)

    m1 <- fitted(fit1)
    m2 <- fitted(fit2)

    tstar <- m1/(m1+m2)

    G1 <- summary(fit1)$deviance
    G2 <- summary(fit2)$deviance

    err <- abs((G1+G2)-(G1.old+G2.old))

  }

  return(list(K=K,err=err,tstar=tstar,
         fit1=fit1,fit2=fit2,d1=d1,d2=d2))

}
```

Fitting in R we get

```
> result <- lcm(stouffer,tol=0.00001)
There were 50 or more warnings
(use warnings() to see the first 50)
> warnings()
Warning messages:
1: non-integer x = 0.450371
2: non-integer x = 1.485916
[...]
50: non-integer x = 21.600592
```

- The non-integer cell counts are a consequence of the E-M algorithm; you can verify that maximizing the Poisson or multinomial likelihood still makes sense mathematically.
- We can look at barplots in each of the two latent classes to see how the LC model split up the data [see next slide]...

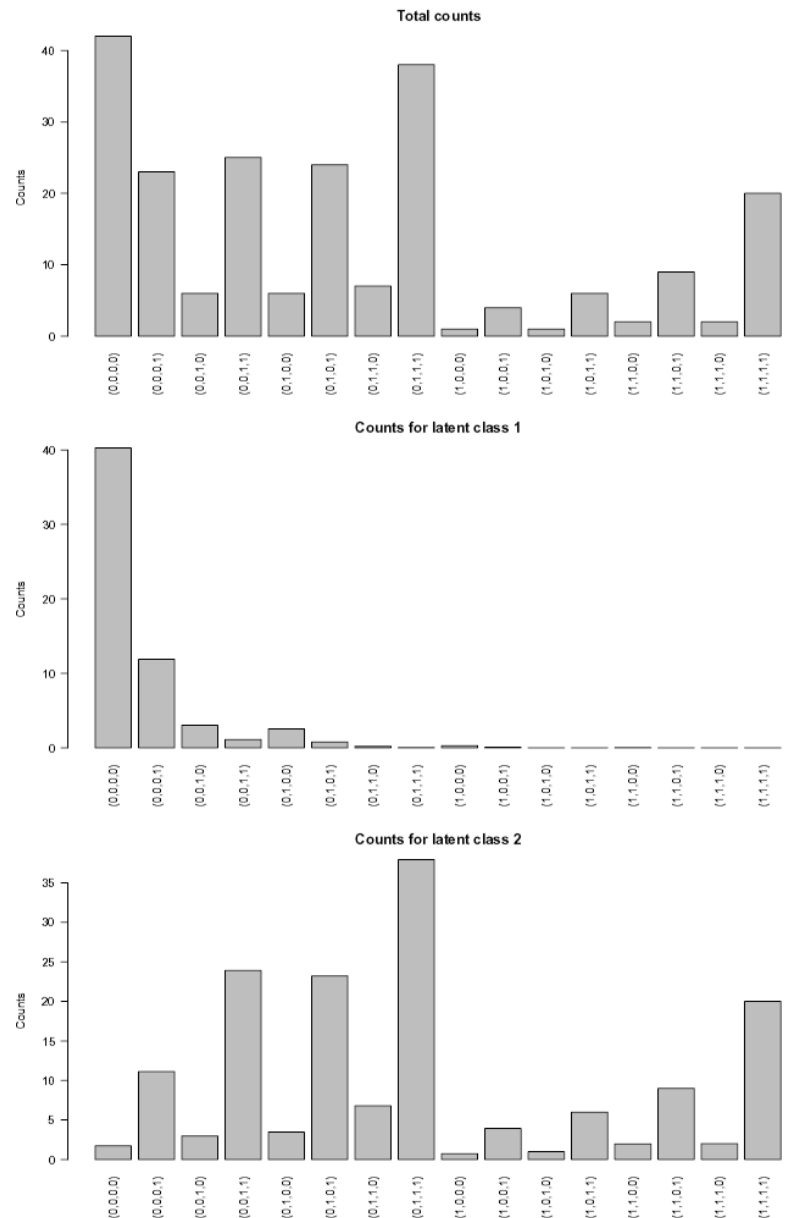
- LC #1: individuals excessively likely to respond 0 to most questions. . .

- LC #2: $y_1 = 0$ frequencies are roughly proportional to $y_1 = 1$ frequencies, allowing cond'l indep.

- Note that the deviances in each class add up to the overall deviance:

```
> m <- fitted(result$fit1) + fitted(result$fit2)
> 2*sum(n*log(n/m))
[1] 2.719957
> result$fit1$deviance + result$fit2$deviance
[1] 2.719960
```

You should be able to prove that this must happen!



Issues and Warnings

Many other issues remain for latent class analysis. . .

- How many latent classes? We chose two. . .
  - The best approaches to choosing the number of latent classes tend to involve Bayes Factor (or BIC) comparisons of models with different numbers of latent classes.
- Reliable MLE optimization. . .
  - LC models tend to have one or more secondary maxima, so that it is difficult to see that you are at the global maximum without a lot of trial and error with starting values.
- Label-switching issues
  - "The" global maximum will actually occur at $W!$ different locations on the likelihood surface, due to equivalent re-labellings of the latent classes.
  - This is an annoyance for careful inference using MLE methods. . .

- For MCMC methods it can be a disaster, since the Markov Chain will visit different ones of these modes and average over them, producing mashed potatoes for parameter estimates...

- Interpretation...
  - LC models for prediction have all of the above problems.
  - In Social Science (and other) applications, we ***also*** often want to interpret the latent classes in terms of the underlying science.
  - This can be very useful, but it can also be dangerous—reifying apparent structure in the latent classes that is't scientifically justifiable.
  - Similar problems occur with mixture-of-normal models, factor analysis models, etc.

- Etc., etc., ...

Details and Difficulties

Marin, J.M., Mengersen, K. & Robert, C.P. (2004). Bayesian modelling and inference on mixtures of distributions. *Handbook of Statistics 25*, D. Dey and C.R. Rao (eds). Elsevier-Sciences (to appear).

only deals with continuous-data mixtures, but it addresses most of the computational and inferential issues found in current discussions of mixture models by statisticians; the same issues apply to LCA. *A pdf is included with these lecture notes.*

Fienberg, S. E., Hersh, P., Rinaldo, A., & Zhou, Y. (2007). Maximum Likelihood Estimation in Latent Class Models For Contingency Table Data. Document arXiv:0709.3535v1, available at http://arxiv.org/abs/0709.3535

provides an account of LC models from the point of view of algebraic statistics and gives new examples suggesting that ML estimation of LC models can be quite delicate. *A pdf is also included with these lecture notes.*