Jan de Leeuw

Estimation in latent Class Analysis

RB 001 - 73

0:

In this paper we introduce several different methods for estimating the parameters of the discrete latent class model for manifest dichotomies.

1:

The model is meant for n different m-dimensional binary variates $\underset{\sim}{y}_i$. We assume

LC1: $BI_{(m)}(\underset{\sim}{y}_i)$ for all $i=1,\dots,n$.

LC2: $IID(\underset{\sim}{y}_1,\dots,\underset{\sim}{y}_n)$.

LC3: $\text{prob}(\underset{\sim}{y}_{ij} = 1 \mid \forall j \in J) = \sum_{s=1}^{p} \theta_s \prod_{j \in J} \pi_j^s$

for all $J \subset \mathcal{J}_m$ and for all $i=1,\dots,m$.

Here the θ_s are called latent class frequencies. They must satisfy

RQ1: $\sum_{s=1}^{p} \theta_s = 1$.

RQ2: $0 \leq \theta_s \leq 1$ for all $s=1,\dots,p$.

The π_j^s are called latent marginals, and must satisfy

RQ3: $0 \leq \pi_j^s \leq 1$ for all $j=1,\dots,m$; $s=1,\dots,p$.

It follows from LC1-LC3 that

$$\text{prob}(\underset{\sim}{y}_{i1} = y_{i1}, \dots, \underset{\sim}{y}_{im} = y_{im}) = \sum_{s=1}^{p} \theta_s \prod_{j=1}^{m} (\pi_j^s)^{y_{ij}} (1 - \pi_j^s)^{1-y_{ij}}$$

for all binary vectors y_{ij}.

2:

Let $J \subset \mathcal{J}_m$ and

$$\underset{\sim}{s}(J) = \frac{1}{n} \sum_{i=1}^{n} \left(\prod_{j \in J} \underset{\sim}{y}_{ij} \right).$$

It follows from LC1-LC3 that

$$\underset{\sim}{s}(J) \xrightarrow{\mathcal{L}} \eta\left(p(J), \frac{1}{n} p(J)(1 - p(J))\right),$$

with

$$p(J) = \text{prob}(y_{ij} = 1 \mid \forall j \in J).$$

We set

$p(\emptyset) = 1$.

For $J \subset \mathcal{J}_m$ and $L \subset \mathcal{J}_m$ we find

$$\begin{bmatrix} \underset{\sim}{s}(J) \\ \underset{\sim}{s}(L) \end{bmatrix} \xrightarrow{\mathcal{L}} \eta\left\{ \begin{pmatrix} p(J) \\ p(L) \end{pmatrix}, \frac{1}{n} \begin{pmatrix} p(J)(1 - p(J)) & p(J \cup L) - p(J)p(L) \\ p(J \cup L) - p(J)p(L) & p(L)(1 - p(L)) \end{pmatrix} \right\}$$

Obviously the $p(J)$ can be estimated consistently by

$$\hat{p}(J) = \frac{1}{n} \sum_{i=1}^{n} \left(\prod_{j \in J} y_{ij} \right).$$

Let $J_1, \dots, J_r \subset \mathcal{J}_m$ be chosen in such a way that the matrix V with elements

$v_{ab} = \text{cov}(\underset{\sim}{s}(J_a), \underset{\sim}{s}(J_b))$

is nonsingular with probability tending to one. Suppose moreover

$mp + p - 1 \leqslant r \leqslant 2^m.$

By using the $\hat{p}(J)$ we can estimate $\{v_{ab}\}$ and its inverse $\{v^{ab}\}$. The statistic

$$\mathcal{J} = \sum_{a=1}^{r} \sum_{b=1}^{r} \hat{v}^{ab}(p(J_a) - \hat{p}(J_a))(p(J_b) - \hat{p}(J_b))$$

can be used for estimation and testing purposes. The estimates of θ_s and π_j^s will be efficient in the submodel

$$p(J_a) = \sum_{s=1}^{p} \theta_s \prod_{j \in J_a} \pi_j^s$$

for all $a=1,\ldots,r$, but in general they do not use all the information in the data and they are not efficient in the complete model LC1-LC3. If we use the restriction RQ1 only, and we forget about RQ2 and RQ3, then

$$\min \mathcal{J} \xrightarrow{\mathcal{L}} \chi^2(r - mp - p + 1).$$

3:

By changing the latent class frequencies according to

$$\hat{\theta}_s = \theta_s + \varepsilon_s,$$

we find

$$\hat{\mathcal{J}} = \mathcal{J} + 2\varepsilon' h + \varepsilon' G \varepsilon,$$

with

$$h_s = \sum_{a=1}^{r} \sum_{b=1}^{r} \hat{v}^{ab}\, p_s(J_a)\,(p(J_b) - \hat{p}(J_b)),$$

$$g_{st} = \sum_{a=1}^{r} \sum_{b=1}^{r} \hat{v}^{ab}\, p_s(J_a)\, p_t(J_b),$$

and

$$p_s(J) = \prod_{j \in J} \pi_j^s.$$

If we change the latent marginals of item 1 according to

$$\tilde{\pi}_j^s = \pi_j^s + \delta^{j1} \lambda_s,$$

we find

$$\tilde{\mathcal{J}} = \mathcal{J} + 2\lambda' u + \lambda' W \lambda,$$

with

$$u_s = \theta_s \sum_{a=1}^{r} \sum_{b=1}^{r} \frac{p_s(J_a)}{\pi_1^s} (p(J_b) - \hat{p}(J_b))\, \phi_a(\ell) \hat{v}^{ab},$$

$$w_{st} = \theta_s \theta_t \sum_{a=1}^{r} \sum_{b=1}^{r} \hat{v}_{ab} \frac{p_s(J_a)}{\pi_1^s} \frac{p_t(J_b)}{\pi_1^t} \phi_a(1) \phi_b(1) ,$$

with

$$\phi_a(1) = \begin{cases} 1 & \text{if } 1 \in J_a \\ 0 & \text{otherwise} \end{cases} .$$

The computational procedure associated with these approximations is to minimize $\mathcal{J}$ for given latent marginals π_j^s over the θ_s. If θ_s satisfies the restrictions RQ1 and RQ2, then we must require

$$\sum_{s=1}^{p} \varepsilon_s = 0,$$

$$-\theta_s \leq \varepsilon_s \leq 1 - \theta_s.$$

If

$$\hat{\varepsilon}_s = \frac{e'G^{-1}h}{e'G^{-1}e} G^{-1}e - G^{-1}h$$

satisfies the inequality restrictions, then it is the optimal solution. Otherwise we must use quadratic programming.

Other subproblems minimize $\mathcal{J}$ over π_1^s with all π_j^s with $j \neq 1$ and all θ_s fixed. We use the restrictions RQ3.

$$-\pi_1^s \leq \lambda_s \leq 1 - \pi_1^s.$$

If

$$\hat{\lambda}_s = -W^{-1}u$$

satisfies these restrictions it is the optimal solution. Otherwise we use quadratic programming.

β:

The developments in the previous section makes it desirable to have a fast and reliable method for solving the bounded variables QP problem

$$\mathcal{J} = x'Ax + 2b'x + c \text{ min!}$$

subject to

$$y_i \leq x_i \leq z_i$$

and, possibly,

$$\sum x_i = 0.$$

Assume, without loss of generality, that

$y_i \neq z_i$

for all i. Start with a feasible x^0, set k = 0. Let

$S_-(k) = \{ i \mid x_i^k = y_i \}$,

$S_+(k) = \{ i \mid x_i^k = z_i \}$,

$T(k) = \{ i \mid y_i < x_i^k < z_i \}$,

and solve

$$\mathcal{J} = x'Ax + 2b'x + c \quad \text{min!}$$

$$x_i = y_i \qquad (i \in S_-(k)),$$

$$x_i = z_i \qquad (i \in S_+(k)),$$

$$(\sum x_i = 0).$$

This is equivalent to

$$\mathcal{J} = \sum_{i \in T(k)} \sum_{j \in T(k)} a_{ij} x_i x_j + 2 \sum_{i \in T(k)} x_i \beta_i^k + \gamma^k \qquad \text{min!}$$

$$(\sum_{i \in T(k)} x_i = \delta^k),$$

with

$$\beta_i^k = b_i + \sum_{j \in S_-(k)} a_{ij} y_j + \sum_{j \in S_+(k)} a_{ij} z_j,$$

$$\gamma^k = c + \sum_{i \in S_-(k)} \sum_{j \in S_-(k)} a_{ij} y_i y_j + \sum_{i \in S_+(k)} \sum_{j \in S_+(k)} a_{ij} z_i z_j + 2 \sum_{i \in S_-(k)} \sum_{j \in S_+(k)} a_{ij} y_i z_j + 2 \sum_{i \in S_-(k)} b_i y_i + 2 \sum_{i \in S_+(k)} b_i z_i,$$

$$\delta^k = - \sum_{i \in S_-(k)} y_i - \sum_{i \in S_+(k)} z_i.$$

Let A_k be the part of A formed by all indices $i \in T(k)$. Then the solution to the unrestricted subproblem is

$$\tilde{x}^k = - A_k^{-1} \beta^k,$$

and the solution to the restricted subproblem is

$$\hat{x}^k = - A_k^{-1} \beta^k + \frac{\delta^k + e' A_k^{-1} \beta^k}{e' A_k^{-1} e} A_k^{-1} e .$$

If $\tilde{x}^k$ is feasible we compute

$$t^k = A\tilde{x}^k + b.$$

Choose j, l, m such that

$$t_j^k = \min (t_i^k \mid i \in S_-^k),$$

$$t_l^k = \min (-t_i^k \mid i \in S_+^k),$$

$$t_m^k = \min (t_j^k, t_l^k).$$

If $t_m^k \geqslant 0$ then stop, x^k is optimal. If $t_m^k < 0$ then

$$x^{k+1} = x^k.$$

If $m \in S_+^k$ then $S_+^{k+1} = S_+^k - \{m\}$, if $m \in S_-^k$ then $S_-^{k+1} = S_-^k - \{m\}$. Always $T^{k+1} = T^k \cup \{m\}$. If $\tilde{x}^k$ is not feasible, then we solve

$$\lambda \quad \min!$$

$$0 \leqslant \lambda \leqslant 1,$$

$$y_i \leqslant \lambda x_i^k + (1 - \lambda)\tilde{x}_i^k \leqslant z_i.$$

Let

$$S_-^k = \{ i \mid \tilde{x}_i^k < y_i \} ,$$

$$S_+^k = \{ i \mid \tilde{x}_i^k > z_i \} ,$$

$$\lambda_-^k = \min \{ (y_i - x_i^k)/(x_i^k - x_i^k) \mid i \in S_-^k \} ,$$

$$\lambda_+^k = \min \{ (z_i - x_i^k)/(x_i^k - x_i^k) \mid i \in S_+^k \} ;$$

then the solution is

$$\lambda^k = \min (\lambda_+^k, \lambda_-^k).$$

We set

$$x^{k+1} = \lambda^k x^k + (1 - \lambda^k)\, \tilde{x}^k,$$

$$S_-^{k+1} = S_-(x^{k+1}),$$

$$S_+^{k+1} = S_+(x^{k+1}),$$

$$T^{k+1} = T(x^{k+1}).$$

This procedure ends in a finite number of steps (W.I. Zangwill: Nonlinear Programming, section 8.3, Englewood Cliffs, Prentice Hall, 1969). Observe that the major computational work is the inversion of A_k. If $T_{k+1} = T_k \cup \{ m\}$, then the step from A_k^{-1} to A_{k+1}^{-1} is simple. In more complicated cases we can still use the Gauss-Jordan method efficiently (i.e. we can pivot in a simplex-like tableau).

5:

Take $k \in \mathcal{J}_m$ arbitrary and define $\mathcal{J}_k = \mathcal{J}_m - \{k\}$. Take $J \subset \mathcal{J}_k$ and $L \subset \mathcal{J}_k$ in such a way that $J \cap L = \emptyset$. LC implies, with obvious notation,

$$P_{JL} = A_J B A_L',$$
$$P_{JL}^k = A_J B C_k A_L' = A_J C_k B A_L'.$$

Assume that A_J and A_L are of rank s. Define

$$\underline{A}_J = A_J (A_J' A_J)^{-1},$$
$$\underline{A}_L = A_L (A_L' A_L)^{-1}.$$

Then

$$P_{JL}^k = A_J B A_L' \underline{A}_L C_k A_L' = A_J C_k \underline{A}_J' A_J B A_L' = P_{JL} \underline{A}_L C_k A_L' = A_J C_k \underline{A}_J' P_{JL},$$

or

$$P_{JL}^k \underline{A}_L = P_{JL} \underline{A}_L C_k,$$
$$P_{LJ}^k \underline{A}_J = P_{LJ} \underline{A}_J C_k,$$

or

$$P_{LJ} P_{JL}^k \underline{A}_L = P_{LJ} P_{JL} \underline{A}_L C_k,$$
$$P_{LJ}^k P_{JL} \underline{A}_L = P_{LJ}^k P_{JL}^k \underline{A}_L C_k^{-1},$$

and

$$P_{JL} P_{LJ}^k \underline{A}_J = P_{JL} P_{LJ} \underline{A}_J C_k,$$
$$P_{JL}^k P_{LJ} \underline{A}_J = P_{JL}^k P_{LJ}^k \underline{A}_J C_k^{-1}.$$

These are simple asymmetric eigenproblems which can be solved for C_k (and for $\underline{A}_J$ and $\underline{A}_L$). The estimates in this section are (Fisher) consistent, and also asymptotically normal. They generalize the 'basic solutions' given by Lazarsfeld and Henry (1968, p 52 and further). Other consistent (but not necessarily asymptotically normal) estimates are given by Mooyaart (1973).

6:

We wrote an APL program LCA for the procedure of section 3 (without incorporating the inequality constraints). The example used was the two-class four-stimuli example from Lazarsfeld & Henry (1968):

.8	.4
.9	.1
.6	.2
.7	.5
.5	.5

We generated random data on the basis of this model. The results of the runs are given below. First column: we used either 10 sets (all sets of one and two stimuli) or 14 sets (all sets of one, two, or three stimuli), or 15 sets (all subsets). Second column: n, the size of the random sample. Fifth column: number of cycles (a cycle is a single step which changes θ and m steps that change the π_j^s). We always used the configuration on the basis of which the data were generated as a starting point. The precision (stop criterion) is the maximum change of any one of the parameters in a cycle. It is given in column 8. In column 7 the number of parameters that converged to values not in [0,1] is given, and the CPU time used is presented in column 6 (this CPU time includ the time used to construct the random sample).

As a preliminary conclusion it seems that a batch version of the program in PL/I or Fortran could be practical for larger sets of stimuli. The unrestricte versions can be used if the sample is large enough. A very good starting configuration is needed (some runs with random starts proved this). If the number of subsets is relatively large compared with the number of sample elements we shall have to use the restricted version. In this case the asympto tic distributional theory has to be modified (formula's for asymptotic disper- sions of the estimates in section 5 and in the restricted and unrestricted versions of section 3 will be presented in another paper).

10	25	X^2 = 1.272	dfr = 1	NCYC = 89	CPU = ?	IMPR = 2	EPS = 1E-3
10	50	X^2 = 7.238	dfr = 1	NCYC = 13	CPU = ?	IMPR = 0	EPS = 1E-4
10	100	X^2 = .461	dfr = 1	NCYC = 5	CPU = ?	IMPR = 1	EPS = 1E-3
10	100	X^2 = .500	dfr = 1	NCYC = 5	CPU = ?	IMPR = 0	EPS = 1E-3
10	250	X^2 = 1.645	dfr = 1	NCYC = 7	CPU = 36	IMPR = 0	EPS = 1E-4
10	1000	X^2 = 2.516	dfr = 1	NCYC = 8	CPU = 114	IMPR = 0	EPS = 1E-4
14	50	X^2 = 10.431	dfr = 5	NCYC = ?	CPU = ?	IMPR = 3	EPS = 1E-4
14	100	X^2 = 8.848	dfr = 5	NCYC = 37	CPU = ?	IMPR = 1	EPS = 1E-4
14	100	X^2 = 2.760	dfr = 5	NCYC = 47	CPU = 108	IMPR = 1	EPS = 1E-4
14	250	X^2 = 4.165	dfr = 5	NCYC = 23	CPU = 87	IMPR = 0	EPS = 1E-4
15	100	X^2 = 2.219	dfr = 6	NCYC = 24	CPU = ?	IMPR = 1	EPS = 1E-3
15	250	X^2 = 13.290	dfr = 6	NCYC = 32	CPU = 113	IMPR = 0	EPS = 1E-4
15	250	X^2 = 4.760	dfr = 6	NCYC = 18	CPU = ?	IMPR = 0	EPS = 1E-4