

5-1-2010

JMASM30 PI-LCA: A SAS Program Computing the Two-point Mixture Index of Fit for Two-class LCA Models with Dichotomous Variables (SAS)


Dongquan Zhang

DMS International, dq.zhang@dmsinetwork.com

C. Mitchell Dayton

University of Maryland, cdayton@umd.edu

Follow this and additional works at: <http://digitalcommons.wayne.edu/jmasm>

 Part of the [Applied Statistics Commons](#), [Social and Behavioral Sciences Commons](#), and the [Statistical Theory Commons](#)

Recommended Citation

Zhang, Dongquan and Dayton, C. Mitchell (2010) "*JMASM30* PI-LCA: A SAS Program Computing the Two-point Mixture Index of Fit for Two-class LCA Models with Dichotomous Variables (SAS)," *Journal of Modern Applied Statistical Methods*: Vol. 9 : Iss. 1 , Article 32.

DOI: 10.22237/jmasm/1272688260

Available at: <http://digitalcommons.wayne.edu/jmasm/vol9/iss1/32>

This Algorithms and Code is brought to you for free and open access by the Open Access Journals at DigitalCommons@WayneState. It has been accepted for inclusion in Journal of Modern Applied Statistical Methods by an authorized editor of DigitalCommons@WayneState.

JMASM ALGORITHMS AND CODE

JMASM30 PI-LCA: A SAS Program Computing the Two-point Mixture Index of Fit for Two-class LCA Models with Dichotomous Variables (SAS)

Dongquan Zhang
DMS International

C. Mitchell Dayton
University of Maryland
College Park

The two-point mixture index of fit enjoys some desirable features in model fit assessment and model selection, however, a need exists for efficient computational strategies. Applying an NLP algorithm, a program using the SAS matrix language is presented to estimate the two-point index of fit for two-class LCA models with dichotomous response variables. The program offers a tool to compute π^* for two-class models and it also provides an alternative program for conducting latent class analysis with SAS. This study builds a foundation for further research on computational approaches for M-class models.

Key words: Pi-star, two-class LCA models, SAS.

Introduction

The two-point mixture index of fit, π^* , was introduced to address the issue of model fit for frequency data in two-way contingency tables (Rudas, et al., 1994; Xi, 1994; Clogg, et al., 1995; Xi & Lindsay, 1996). This index has been extended to a variety of other theoretical models. For example, Rudas & Zwick (1997) discussed the use of π^* in differential item functioning, Rudas (1999) studied applications of π^* with regression models involving continuous variables and Dayton (1999; 2003) extended the application of π^* to latent class models.

For a two-point mixture, $P = (1 - \pi)\Phi + \pi\Psi$, let Φ denote the probability distribution of some hypothesized frequency model, H , let Ψ represent an unspecified probability distribution, and let π indicate the proportion of the population that is

not intrinsically described by model H . Then, the mathematical model for π^* can be written as (Rudas, et al., 1994):

$$\pi^* = \inf\{\pi \mid P = (1 - \pi)\Phi + \pi\Psi, \Phi \in H\} \quad (1)$$

In effect, π^* is defined as the smallest value of π for which P remains true for model H and can be viewed as “a measure of the proportion of the population measured with error” (Rudas, et al., 1994, p. 628) or as a measure of lack of fit (Rudas, et al., 1994; Xi, 1994; Xi & Lindsay, 1996). In practice, the minimum proportion of cases that must be removed from the frequency table is compared to the remaining cases in order to provide perfect fit for H (Dayton, 2003).

As opposed to conventional approaches, such as the G^2 likelihood ratio test and various information criteria such as AIC, π^* represents a new perspective with respect to model-fit assessment and provides an easy-to-interpret alternative basis for model comparison and selection. Rudas, et al. (1994) summarized the desirable properties of this new index as: (1) unique; (2) defined on the 0, 1 interval; (3) decreasing in magnitude for increasingly more complex models when comparing nested models; and (4) invariant to multiplicative transformation of the frequency data. This latter

Dongquan Zhang is an Operations Research Analyst in DMS International. Email: dq.zhang@dmsinetwork.com. C. Mitchell Dayton is a Professor Emeritus and past Chair in the Department of Measurement & Statistics at the University of Maryland. Email: cdayton@umd.edu.

property is particularly interesting because it means that the magnitude of π^* is not dependent on sample (although its sampling error is).

Application of π^* to Latent Class Models

A latent class model with T classes is, from a mathematical point of view, a finite mixture of product-multinomial probability functions. Considering a four-variable model as an example, the unconditional probability for the response vector, $Y = \{i j k l\}$, can be defined as:

$$P(Y) = \sum_{t=1}^T \tau_t P(Y | t) \tag{2}$$

where τ_t is the proportion in latent class t , and $P(Y | t)$ is the product of the conditional response probabilities for the four variables corresponding to the response pattern $\{i j k l\}$, given membership in latent class t . The latent class model is subject to the restrictions that: (1) the latent class proportions sum to 1; (2) the conditional response probabilities, given latent class membership, sum to 1 for each variable; (3) the variables are conditionally independent within any given class (Lazarsfeld & Henry, 1968; Goodman, 1974; Dayton, 1999; among others).

In latent class analysis, Chi-square goodness-of fit tests and information criteria are widely applied procedures for assessing model fit and for model selection. These methods are open to the criticisms that: (1) with small sample size or sparse data, the statistics do not asymptotically follow appropriate χ^2 distributions; and (2) with large sample size, it is highly likely that the null hypothesis will be rejected for relatively trivial effects. Therefore, Chi-square tests may not be appropriate for model selection under those circumstances. For information criteria, such as AIC , it is not clear how much the effect of sample size persists when the penalty term is applied. In addition, information criteria cannot be used to assess model fit in an absolute sense inasmuch as interpretation of magnitudes of information criteria per se is difficult (Rudas, et al., 1994).

For the s^{th} response vector, the latent class model can be incorporated into the two-point mixture model as follows (Dayton, 2003):

$$P_s = (1 - \pi)P(y_s) + \pi\Psi_s \tag{3}$$

where $P(y_s)$ represents the probability distribution for the s^{th} response vector or response pattern. π^* is obtained as the minimum value of π when the model holds true across all response vectors (Dayton, 2003). The definition of π^* circumvents the drawbacks of Chi-square statistics, thus, the index enjoys some unique advantages in model selection.

Methodology

Computational Approach

Programs for LCA such as LEM or SAS PROC LCA (Lanza, et al., 2007) do not provide options for computing π^* . However, π^* can be estimated using the iterative procedures proposed by Rudas, et al. (1994) and with MLE or nonlinear programming (NLP) algorithms (Xi, 1994; Xi & Lindsay, 1996). Dayton (2003) discusses computational strategies for the fit index applied to latent class and IRT (Rasch) models and presents examples using Microsoft Excel Solver, a program that is based on a NLP algorithm. For latent class models, Dayton (2003) detailed a computational strategy in two stages: in the first stage, the NLP parameters are defined as $a_{it}, b_{jt}, c_{kt}, d_{lt}$, etc. such that $\hat{n}_s = a_{it} \times b_{jt} \times c_{kt} \times d_{lt} \times \dots$. Given the nonlinear constraint that the total expected frequency is equal to the total observed frequency, conventional MLEs of the parameters for an unrestricted latent class model can be estimated by minimizing G^2 as the objective function.

In the second stage, more nonlinear constraints, which specify the relationship between the expected frequency and the observed frequency for each response vector, are applied in NLP. The objective function is then redefined as maximizing the total expected frequency (or, equivalently, minimizing π , which is a function of the expected frequencies). After convergence to some preset criterion, an estimate of π^* is obtained (Dayton, 2003).

Technically, simply applying the second stage alone generates an estimate of π^* . However, an associated problem, which is increasingly crucial when the number of parameters increases, is the selection of start values because good start values are critical to computational efficiency and accuracy. With inappropriate start values, the optimization procedure may fail to converge, may converge at a local optimum, or may encounter other unexpected difficulties.

Although it is possible to provide different sets of start values and to examine the results in a single stage, a more efficient approach is to first conduct a conventional unrestricted LCA analysis and then start from the resultant parameter estimates which are, in general, closer to the final NLP estimates than arbitrarily selected start values. Although start values still need to be selected for first stage optimization, one benefit of the two-stage approach is that the closer estimates of the parameters are secured with only one (not multiple) constraints, no matter how many parameters are in the model. Hence, in the second stage, computational efficiency is achieved with faster convergence since the number of NLP function calls is greatly reduced.

Two SAS NLP subroutines, NLPNMS and NLPQN, are available to implement nonlinear constraints. The NLPQN subroutine applies quasi-Newton optimization technique that involves computing first-order partial derivatives in the gradient vector or the Jacobian matrix. It is suitable for medium to moderately large problems (NLPQN, SAS 9.1 Documentation, 2007) that contain relatively large numbers of parameters; NLPNMS is suitable for smaller problems. For nonlinearly constrained optimization, the NLPQN subroutine applies a modification of Powell's (1978, 1982) Variable Metric Constrained WatchDog algorithm (NLPQN call, SAS 9.1 Documentation, 2007). PI-LCA implements the NLPQN subroutine for optimization.

SAS Program Description

As the computation involves relatively complex matrix operations, the current version of the procedure is restricted to two-class LCA

models with dichotomous response variables. The SAS program, PI-LCA, is designed to compute π^* for models for varying numbers of variables. However, for large numbers of variables computational time may become excessive. Factors influencing the number of function calls include selection of start values, number of parameters, and data structure, such as the number of zero-frequency vectors.

The SAS program PI-LCA has four sections:

1. Macro variables. Specifically, the following quantities are labeled and input as macro variables:
 - a. Number of dichotomous variables;
 - b. Number of latent classes (set at 2 in current version of program);
 - c. Observed sample size;
 - d. Start values for the first stage optimization;
 - e. Input data file name and location.

In this area of the program, the user must make adjustments in accordance with the data under consideration.

2. Data input for computing the expected frequencies. The data file can be any format (such as ASCII) that is acceptable to SAS. As the NLP procedure involves nonlinear constraints with regard to each response vector, aggregated data by the response pattern must be used as input. Assume that the number of items is numvar (as suggested previously), there should be numvar+1 fields in the dataset, with the first numvar fields representing the response patterns (e.g., 1 1 1 1 for 4 items) - the last field being the observed frequency. For ASCII data input, such as the text data generated by Microsoft Notepad, the fields should be space delimited; for example: 1 1 1 1 freq. For each observation (response pattern), the first numvar fields can either be 1's and 2's or 0's and 1's (see Table 1).
3. The first stage of the optimization procedure. This stage computes

conventional, unrestricted two-class LCA parameters using the NLPQN algorithm. The objective function that is minimized is G^2 , given the constraint that the total observed frequency and the total expected frequency are equal. In addition, boundary constraints are applied to ensure that all parameter estimates are non-negative. Because start values are randomly selected in this stage, detailed NLP options (items 4-8 in the option vector for NLPQN, which may vary from case to case) are specified to obtain accurate estimates. The options may increase the number of function calls and make the convergence slower, especially when there are large numbers of parameters. When the procedure converges, the start values for the second stage are obtained. It is suggested that distinct sets of start values for the first stage should be tried to ensure that a global optimum has been obtained.

4. The second stage of the optimization procedure. In this stage, both the objective function and nonlinear constraints are redefined. In most cases, the convergence is relatively fast as the start values are close to optimum. In general, items 4-8 in the option vector for NLPQN do not need to be changed from default values. At convergence, the estimate of π^* is obtained.

Results: Exemplary Data

Example 1: Academic Cheating Data (Four Items)

Dayton (2003) used Microsoft ExcelSolver to compute π^* for a two-class LCA model with frequency data for four dichotomous (2 = yes, have engaged in this cheating behavior, and 1 = no, have not engaged in this cheating behavior) items from a survey concerned with academic cheating behavior by college students (see Table 1).

Table 1: Academic Cheating Data

Item				Frequency
A	B	C	D	
1	1	1	1	207
1	1	1	2	46
1	1	2	1	7
1	1	2	2	5
1	2	1	1	13
1	2	1	2	4
1	2	2	1	1
1	2	2	2	2
2	1	1	1	10
2	1	1	2	3
2	1	2	1	1
2	1	2	2	2
2	2	1	1	11
2	2	1	2	4
2	2	2	1	1
2	2	2	2	2
				319

Input to Section (1) of the SAS Program

```
*****
PI-LCA: A SAS PROGRAM COMPUTING THE TWO-POINT MIXTURE INDEX OF FIT FOR
TWO-CLASS LCA MODELS WITH DICHOTOMOUS VARIABLES
*****;

* SECTION 1: PROVIDE VALUES FOR FOLLOWING 5 MACRO VARIABLES;

%let numvar=4;          * NUMBER OF ITEMS (MANIFEST VARIABLES);
%let numcl=2;          * NUMBER OF CLASSES;
%let numsap=319;       * NUMBER OF SAMPLE SIZE;
%let start=1;          * START VALUES FOR THE FIRST STAGE OPTIMIZATION;
%let datafile = "c:\cheat4.txt"; * LOCATION OF THE INPUT DATA FILE;
*****;
```

PI-LCA SAS PROGRAM FOR TWO-CLASS LCA DICHOTOMOUS VARIABLE MODELS

Selected Output: SAS output 1- Call NLPQN Subroutine in the First Stage to Conduct Latent Class Analysis

Computing Pi-star, the Two-Point Mixture Fit Index
 The First Stage: Latent Class Analysis
 The Objective Function Is to Minimize G-square

Optimization Results			
Iterations	98	Function Calls	107
Gradient Calls	100	Active Constraints	1
Objective Function	3.8821212398	Maximum Constraint Violation	1.4590216E-7
Maximum Projected Gradient	0.0009804183	Value Lagrange Function	3.8821210939
Maximum Gradient of the Lagran Func	0.0008997954	Slope of Search Direction	-3.02492E-7

Parameter Estimates			
N Parameter	Estimate	Gradient Objective Function	Gradient Lagrange Function
1 X1	8.023430	-32.820114	0.000030386
2 X2	4.030813	-64.491314	0.000059927
3 X3	2.585951	-99.709946	0.000146
4 X4	2.407726	-90.980625	0.000146
5 X5	0.135555	-32.819815	0.000329
6 X6	0.121356	-64.490988	0.000386
7 X7	0.099626	-99.710992	-0.000900
8 X8	0.535523	-90.980890	-0.000119
9 X9	1.361351	-15.918088	0.000058771
10 X10	1.203668	-17.486176	0.000071031
11 X11	1.237312	-32.453413	0.000193
12 X12	2.147240	-14.876575	0.000191
13 X13	1.856461	-15.917976	0.000171
14 X14	1.725583	-17.486067	0.000181
15 X15	0.340991	-32.453140	0.000466
16 X16	1.295821	-14.876776	-0.000010208

Value of Objective Function = 3.8821212398
 Value of Lagrange Function = 3.8821210939

Latent Class Analysis
 Observed Frequency Expected Frequency

207	205.71667
46	47.414163
7	8.9574477
5	2.4494936
13	12.303603
4	5.1148321
1	1.9535633
2	1.0899392
10	9.3388155
3	4.3394233
1	1.7671869
2	1.0165214
11	8.6134428
4	5.1590278
1	2.3494963
2	1.4163751

Total Expected Frequency
 319

LC1 Proportion LC2 Proportion
 0.839431 0.160569

Conditional Probabilities

CP Positive Response (1)	1	0.9833858	0.4230674
	2	0.970773	0.4109132
	3	0.9629034	0.783951
	4	0.8180504	0.6236428
CP Negative Response (2)	1	0.0166142	0.5769326
	2	0.029227	0.5890868
	3	0.0370966	0.216049
	4	0.1819496	0.3763572

ZHANG & DAYTON

SAS Output 2: Call NLPQN Subroutine in the Second Stage to Compute π^*

Computing Pi-star, the Two-Point Mixture Fit Index

The Second Stage: Pi Optimization

The Objective Function Is to Maximize the Total Expected Frequency

Optimization Results			
Iterations	6	Function Calls	8
Gradient Calls	8	Active Constraints	10
Objective Function	310.01091238	Maximum Constraint Violation	9.3996391E-8
Maximum Projected Gradient	3.9912676E-7	Value Lagrange Function	-310.0109122
Maximum Gradient of the Lagran Func	3.0088341E-7	Slope of Search Direction	-4.425832E-7

Parameter Estimates			
N Parameter	Estimate	Gradient Objective Function	Gradient Lagrange Function
1 X1	8.033189	32.624665	2.9195017E-9
2 X2	4.037358	64.709589	-0.000000232
3 X3	2.592207	100.059213	-0.000000175
4 X4	2.414567	90.971362	0.000000301
5 X5	0.176867	32.624664	6.428263E-11
6 X6	0.101910	64.709588	-5.847028E-9
7 X7	0.084711	100.059213	-5.726554E-9
8 X8	0.529770	90.971362	6.6015519E-8
9 X9	1.324498	13.361448	-4.779081E-8
10 X10	0.937841	15.143763	7.451175E-8
11 X11	1.350402	28.603220	-8.034978E-8
12 X12	2.385143	12.948459	3.8478315E-8
13 X13	1.830894	13.361447	-6.606271E-8
14 X14	1.846183	15.143764	0.000000147
15 X15	0.123579	28.603222	-7.353005E-9
16 X16	0.870889	12.948459	1.4049627E-8

Value of Objective Function = 310.01091238

Value of Lagrange Function = 310.01091224

Pi-Star Results		
Observed Frequency	Expected Frequency	
207	207	
46	46	
7	7	
5	1.5891942	
13	13	
4	4	
1	0.8881988	
2	0.2999071	
10	10	
3	3	
1	0.6521739	
2	0.2168445	
11	11	
4	4	
1	1	
2	0.3645937	

Total Expected Frequency: 310.01091

Pi-Star: 0.028179

LC1 Proportion	LC2 Proportion
0.8640029	0.1359971

Conditional Probabilities		
CP Positive Response (1)	1	0.9784573 0.419757
	2	0.9753798 0.3368651
	3	0.9683548 0.9161599
	4	0.8200717 0.7325305
CP Negative Response (2)	1	0.0215427 0.580243
	2	0.0246202 0.6631349
	3	0.0316452 0.0838401
	4	0.1799283 0.2674695

PI-LCA SAS PROGRAM FOR TWO-CLASS LCA DICHOTOMOUS VARIABLE MODELS

In this example, the start values for all the parameters are set equal to 1. In general, distinct sets of start values should be employed to ensure a global maximum. In this stage, there are 98 iterations and 107 function calls. The maximum constraint violation is in the range of 1E-6, which is acceptable. The objective function (G^2) is minimized at 3.88. With the NLP parameters, the latent class proportions and the conditional probabilities (CP) for the LCA model are computed.

The start values are imported from the first stage output. The objective function is redefined as maximizing the total expected frequency, which converges at 310.01 (in contrast to the total observed frequency of 319). There are only 6 iterations and 8 function calls prior to convergence (compared to 98 and 107 in

the first stage). The estimated value of π^* converges at 0.028. Thus, only 2.8% of the cases in the population are estimated as not described by the two-class model; this suggests adequate model-data fit. (See SAS Output 2.)

Example 2: Drug Use Data (Five Items)

Five dichotomous (2 = yes, have used this drug and 1 = no, have not used this drug) items in the drug use data set with a large number of zero frequencies (see Table 2). Following the approach of Clogg, et al. (1991) in applying flattening constants to deal with the sparse data that do not support conventional maximum likelihood analysis, zero frequencies are replaced with 0.5, which enables the NLP optimization to converge. This increased the total frequency from 7,224 to 7,233.

Table 2: Drug Use Data

Item					Frequency	
A	B	C	D	E	Original	Replaced
1	1	1	1	1	710	710
1	1	1	1	2	0	0.5
1	1	1	2	1	0	0.5
1	1	1	2	2	0	0.5
1	1	2	1	1	4	4
1	1	2	1	2	0	0.5
1	1	2	2	1	0	0.5
1	1	2	2	2	0	0.5
1	2	1	1	1	263	263
1	2	1	1	2	0	0.5
1	2	1	2	1	0	0.5
1	2	1	2	2	0	0.5
1	2	2	1	1	21	21
1	2	2	1	2	0	0.5
1	2	2	2	1	0	0.5
1	2	2	2	2	0	0.5
Totals					7224	7233

Data continues in next table

ZHANG & DAYTON

Input to Section (1)

```
*****
PI-LCA: A SAS PROGRAM COMPUTING THE TWO-POINT MIXTURE INDEX OF FIT FOR
TWO-CLASS LCA MODELS WITH DICHOTOMOUS VARIABLES
*****;

* SECTION 1: PROVIDE VALUES FOR FOLLOWING 5 MACRO VARIABLES;

%let numvar=5;                * NUMBER OF ITEMS (MANIFEST VARIABLES);
%let numcl=2;                * NUMBER OF CLASSES;
%let numsap=7224;           * NUMBER OF SAMPLE SIZE;
%let start=1.2;             * START VALUES FOR THE FIRST STAGE OPTIMIZATION;
%let datafile = "c:\druguse.txt"; * LOCATION OF THE INPUT DATA FILE;
*****
```

SAS output 1- Call NLPQN Subroutine in the First Stage to Conduct Latent Class Analysis

Computing Pi-star, the Two-Point Mixture Fit Index
The First Stage: Latent Class Analysis
The Objective Function Is to Minimize G-square

Optimization Results

Iterations	267	Function Calls	326
Gradient Calls	269	Active Constraints	1
Objective Function	469.21431307	Maximum Constraint Violation	4.3116415E-8
Maximum Projected Gradient	3.7700252447	Value Lagrange Function	469.21431303
Maximum Gradient of the Lagran Func	2.8297208894	Slope of Search Direction	-9.946578E-8

Latent Class Analysis

Total Expected Frequency
7233

LC1 Proportion	LC2 Proportion
0.6394273	0.3605727

Conditional Probabilities

CP Positive Response (1)	1	0.2155591	0.0027008
	2	0.3561529	0.0612726
	3	0.9700971	0.0074087
	4	0.9981559	0.6917313
	5	0.9994005	0.9550512
CP Negative Response (2)	1	0.7844409	0.9972992
	2	0.6438471	0.9387274
	3	0.0299029	0.9925913
	4	0.0018441	0.3082687
	5	0.0005995	0.0449488

PI-LCA SAS PROGRAM FOR TWO-CLASS LCA DICHOTOMOUS VARIABLE MODELS

SAS output 2: Call NLPQN subroutine in the second stage to compute π^*

Computing Pi-star, the Two-Point Mixture Fit Index
 The Second Stage: Pi Optimization
 The Objective Function Is to Maximize the Total Expected Frequency

Optimization Results

Iterations	8	Function Calls	10
Gradient Calls	10	Active Constraints	12
Objective Function	0.1078697651	Maximum Constraint Violation	8.7764806E-6
Maximum Projected Gradient	9.600992E-10	Value Lagrange Function	0.1078697675
Maximum Gradient of the Lagran Func	1.07744E-9	Slope of Search Direction	-5.240666E-9

Total Expected Frequency
6444.7488

Pi-Star
0.1078698

LC1 Proportion LC2 Proportion
0.6285462 0.3714538

Conditional Probabilities

CP Positive Response (1)	1	0.0907209	0.0007148
	2	0.2507106	0.0469887
	3	0.9571076	6.2406E-8
	4	0.9981068	0.7052601
	5	0.9998104	0.9937875
CP Negative Response (2)	1	0.9092791	0.9992852
	2	0.7492894	0.9530113
	3	0.0428924	0.9999999
	4	0.0018932	0.2947399
	5	0.0001896	0.0062125

A vector of start values equal to 1.2 provides better start values than 1's as used in the first example, although the NLP call required comparatively more iterations before convergence. In the first stage, the objective function converges at 469.21. In the second stage, the value of the maximized objective function is 6,444.75 (total expected frequency), which corresponds to a π^* value of 0.108. The result suggests that in order to provide perfect fit for the two-class model, about 11% of the cases in the population are not described by the model H .

Example 3: Abortion Data (Six Items)

The 6-item General Social Survey (GSS) abortion attitude data set (1=Yes, approve abortion for this reason, and 2=No, do not approve abortion for this reason), was collected between 1972 and 1998 and analyzed by Dayton (2006). As shown in Table 3, the total sample size is 27,151. Because there is a zero frequency for the response vector {212121}, it is replaced with .5 as was done in Example 2. The matrix combining the parameters and response patterns is 64x12, which requires relatively a long computational time.

Table 3: Abortion Data

Item						Frequency	Item						Frequency
A	B	C	D	E	F		A	B	C	D	E	F	
1	1	1	1	1	1	10728	2	1	1	1	1	1	61
1	1	1	1	1	2	732	2	1	1	1	1	2	24
1	1	1	1	2	1	12	2	1	1	1	2	1	2
1	1	1	1	2	2	24	2	1	1	1	2	2	6
1	1	1	2	1	1	413	2	1	1	2	1	1	7
1	1	1	2	1	2	503	2	1	1	2	1	2	25
1	1	1	2	2	1	7	2	1	1	2	2	1	5
1	1	1	2	2	2	53	2	1	1	2	2	2	11
1	1	2	1	1	1	29	2	1	2	1	1	1	15
1	1	2	1	1	2	11	2	1	2	1	1	2	7
1	1	2	1	2	1	1	2	1	2	1	2	1	0
1	1	2	1	2	2	1	2	1	2	1	2	2	9
1	1	2	2	1	1	7	2	1	2	2	1	1	6
1	1	2	2	1	2	9	2	1	2	2	1	2	7
1	1	2	2	2	1	4	2	1	2	2	2	1	2
1	1	2	2	2	2	3	2	1	2	2	2	2	12
1	2	1	1	1	1	774	2	2	1	1	1	1	48
1	2	1	1	1	2	1059	2	2	1	1	1	2	91
1	2	1	1	2	1	18	2	2	1	1	2	1	4
1	2	1	1	2	2	60	2	2	1	1	2	2	34
1	2	1	2	1	1	641	2	2	1	2	1	1	46
1	2	1	2	1	2	5643	2	2	1	2	1	2	1100
1	2	1	2	2	1	21	2	2	1	2	2	1	3
1	2	1	2	2	2	1181	2	2	1	2	2	2	1040
1	2	2	1	1	1	7	2	2	2	1	1	1	6
1	2	2	1	1	2	14	2	2	2	1	1	2	8
1	2	2	1	2	1	1	2	2	2	1	2	1	3
1	2	2	1	2	2	3	2	2	2	1	2	2	6
1	2	2	2	1	1	10	2	2	2	2	1	1	8
1	2	2	2	1	2	153	2	2	2	2	1	2	264
1	2	2	2	2	1	2	2	2	2	2	2	1	1
1	2	2	2	2	2	121	2	2	2	2	2	2	2045

Total: 27,151

PI-LCA SAS PROGRAM FOR TWO-CLASS LCA DICHOTOMOUS VARIABLE MODELS

Input to Section (1)

```
*****
PI-LCA: A SAS PROGRAM COMPUTING THE TWO-POINT MIXTURE INDEX OF FIT FOR TWO-
CLASS LCA MODELS WITH DICHOTOMOUS VARIABLES
*****;

* SECTION 1: PROVIDE VALUES FOR FOLLOWING 5 MACRO VARIABLES;

%let numvar=6;                * NUMBER OF ITEMS (MANIFEST VARIABLES);
%let numcl=2;                * NUMBER OF CLASSES;
%let numsap=27151;          * NUMBER OF SAMPLE SIZE;
%let start=2.5;             * START VALUES FOR THE FIRST STAGE OPTIMIZATION;
%let datafile = "c:\abortion6.txt"; * LOCATION OF THE INPUT DATA FILE;
*****
```

SAS output 1- Call NLPQN Subroutine in the First Stage to Conduct Latent Class Analysis

Computing Pi-star, the Two-Point Mixture Fit Index
The First Stage: Latent Class Analysis
The Objective Function Is to Minimize G-square
Optimization Results

Iterations	67	Function Calls	91
Gradient Calls	69	Active Constraints	1
Objective Function	5356.558615	Maximum Constraint Violation	0.0000207942
Maximum Projected Gradient	8.5964821829	Value Lagrange Function	5356.5585942
Maximum Gradient of the Lagran Func	8.7492309364	Slope of Search Direction	-0.000050218

Latent Class Analysis

Total Expected Frequency
27203.5

LC1 Proportion	LC2 Proportion
0.4834715	0.5165285

Conditional Probabilities

CP Positive Response (1)	1	0.9923362	0.6579907
	2	0.9209919	0.0480761
	3	0.9962743	0.8059632
	4	0.9546303	0.0921695
	5	0.9987107	0.6670409
	6	0.9257254	0.0547449
CP Negative Response (2)	1	0.0076638	0.3420093
	2	0.0790081	0.9519239
	3	0.0037257	0.1940368
	4	0.0453697	0.9078305
	5	0.0012893	0.3329591
	6	0.0742746	0.9452551

ZHANG & DAYTON

SAS output 2: Call NLPQN Subroutine in the Second Stage to Compute π^*

Computing Pi-star, the Two-Point Mixture Fit Index
 The Second Stage: Pi Optimization
 The Objective Function Is to Maximize the Total Expected

Optimization Results

Iterations	6	Function Calls	8
Gradient Calls	8	Active Constraints	14
Objective Function	22033.637653	Maximum Constraint Violation	1.4665943E-8
Maximum Projected Gradient	0.0000261874	Value Lagrange Function	-22033.63765
Maximum Gradient of the Lagran Func	0.0000238896	Slope of Search Direction	-2.511921E-6

Pi-Star Results

Total Expected Frequency
22033.638

Pi-Star
0.1884779

LC1 Proportion LC2 Proportion
0.5863454 0.4136546

Conditional Probabilities

CP Positive Response (1)	1	0.9943747	0.8368214
	2	0.933311	0.0220863
	3	0.9973172	0.9735944
	4	0.9632445	0.048304
	5	0.9988897	0.8268844
	6	0.93694	0.0128586
CP Negative Response (2)	1	0.0056253	0.1631786
	2	0.066689	0.9779137
	3	0.0026828	0.0264056
	4	0.0367555	0.951696
	5	0.0011103	0.1731156
	6	0.06306	0.9871414

A vector of values equal to 2.5 was selected as start values. While the latent class proportions are 58% and 42%, respectively, the value of π^* is near 0.188, indicating that in order to provide perfect fit, around 19% of the cases in the population are not taken into account. This suggests that the two-class model does not provide adequate fit; Dayton (2006) considered more complex models for these data.

References

Clogg, C. C., Rudas, T., & Xi, L. (1995). A new index of structure for the analysis of models for mobility tables and other cross-classification. In P. Marsden (Ed.), *Sociological Methodology*, 197-222. Oxford: Blackwell.

Clogg, C. C., Rubin, D. B., Schenker, N., Schultz, B., & Weidman, L. (1991). Multiple imputation of industry and occupation codes in census public-use samples using Bayesian logistic regression. *Journal of the American Statistical Association*, 86(413), 68-78.

Dayton, C. M. (1999). *Latent class scaling analysis*. Thousand Oaks, CA: Sage.

Dayton, C. M. (2003). Applications and Computational Strategies for the two-point mixture index of fit. *British Journal of Mathematical and Statistical Psychology*, 56, 1-13.

Dayton, C. M. (2006). Latent structure of attitudes toward abortion. In *Real Data Analysis*, S. S. Sawilowsky (Ed.), AERA SIG/ES, 293-298.

PI-LCA SAS PROGRAM FOR TWO-CLASS LCA DICHOTOMOUS VARIABLE MODELS

Goodman, L. A. (1974). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*, 61, 215-231.

Lanza, S. T., Lemmon, D. R., Schafer, J. L., & Collins, L. M. (2007). *PROC LCA & PROC LTA user's guide Version 1.13 Beta*. The Methodology Center, Pennsylvania State University.

Lazarsfeld, P. F., & Henry, N. W. (1968). *Latent structure analysis*. Boston: Houghton Mifflin Company.

Rudas, T. (1999). The mixture index of fit and minimax regression. *Metrika*, 50, 163-172.

Rudas, T., Clogg, C. C., & Lindsay, B. G. (1994). A new index of fit based on mixture methods for the analysis of contingency tables. *Journal of the Royal Statistical Society, Series B*, 56, 623-639.

Rudas, T., & Zwick, R. (1997). Estimating the importance of differential item functioning. *Journal of Educational and Behavioral Statistics*, 22, 31-45.

SAS. (2007). SAS OnlineDoc 9.1.3. SAS Institute, Inc. Retrieved June 10, 2007, from <http://support.sas.com/onlinedoc/913/docMainpage.jsp>.

Xi, L. (1994). *The mixture index of fit for the independence model in contingency tables*. Master of Arts paper, Department of Statistics, Pennsylvania State University.

Xi, L., & Lindsay, B. G. (1996). A note on calculating the π^* index of fit for the analysis of contingency tables. *Sociological Methods & Research*, 25, 248-259.

Appendix: The SAS Program to Compute Pi-star with the Cheat4 Data

```
options nodate pageno=1 linesize=80 pagesize=60;

*****
A SAS PROGRAM COMPUTING THE TWO-POINT MIXTURE INDEX OF FIT FOR THE
TWO-CLASS LCA MODELS
*****

* SECTION 1: PROVIDE VALUES FOR FOLLOWING 5 MACRO VARIABLES;

%let numvar=4;      * NUMBER OF ITEMS (MANIFEST VARIABLES);
%let numcl=2;      * NUMBER OF CLASSES;
%let numsap=319;   * NUMBER OF SAMPLE SIZE;
%let start=1;      * START VALUES FOR THE FIRST STAGE OPTIMIZATION;
%let datafile = "c:\cheat4.txt"; * LOCATION OF THE INPUT DATA FILE;

*****

* SECTION 2: PREPARE DATA TO COMPUTE EXPECTED FREQUENCY;

* READ IN DATA FILE;

data lca;
infile &datafile;
input x1-x&numvar count;
run;

*NLP MACRO;

%macro Twoclasspistar;
```

Appendix: The SAS Program to Compute Pi-star with the Cheat4 Data (continued)

```

* CREATE A DATA SET WITH BINARY (0/1) DATA, EACH OBSERVATION
CORRESPONDING TO THE CONDITIONAL JOINT DISTRIBUTION PATTERN OF ONE
RESPONSE VECTOR;

data bin1 (drop=i j );
  do i= &numvar to 1 by -1;
    do j= (2**i) to 1 by -1;
      binary1 = putn((j-1), "binary&numvar.");
      if i=&numvar then
        output;
      end;
    end;
  end;
run;
data bin2 (drop= i j);
  do i=1 to &numvar;
    do j=1 to (2**i);
      binary2 = putn((j-1), "binary&numvar.");
      if i=&numvar then
        output;
      end;
    end;
  end;
run;

data bin (drop=i j binary1 binary2);
merge bin1 bin2;
  array x[&numvar] x1-x&numvar;
  do i=1 to &numvar;
    x[i]=substr(binary1,i,1);
  end;
  array y[&numvar] y1-y&numvar;
  do j=1 to &numvar;
    y[j]=substr(binary2,j,1);
  end;
run;

* CALL SAS PROC IML;

proc iml;

* CONVERT SAS DATAFILES INTO PROC IML MATRICES;

  *WRITE THE BINARY DATA INTO THE MATRIX A;

  use bin;
read all into a;

  * WRITE THE COUNTS OF THE RESPONSE VECTORS INTO THE MATRIX OBSF;

  use lca;
read all var {count} into obsf;

* CREATE A MACRO TO COMPUTE THE EXPECTED FREQUENCY FOR EACH CLASS;

```

Appendix: The SAS Program to Compute Pi-star with the Cheat4 Data (continued)

```

%macro expf;
  b=x[,1:2*&numvar];
      b= b`;
  c=x[,2*&numvar+1:4*&numvar];
      c= c`;
  p=j(2**&numvar,2*&numvar,0);
  q=j(2**&numvar,2*&numvar,0);
  do i=1 to &numcl**&numvar;
      do j=1 to 2*&numvar;
          p[i,j]=a[i,j]*b[j,];
              if p[i,j]=0 then
                  p[i,j]=1;
          else p[i,j]=p[i,j];
          q[i,j]=a[i,j]*c[j,];
              if q[i,j]=0 then
                  q[i,j]=1;
          else q[i,j]=q[i,j];
      end;
  end;
  pjoint=p[,#]; *EXPECTED FREQUENCY FOR EACH RESPONSE VECTOR IN LC 1;
  qjoint=q[,#]; *EXPECTED FREQUENCY FOR EACH RESPONSE VECTOR IN LC 2;
%mend;

*****
*SECTION 3: THE FIRST STAGE - CONVENTIONAL LATENT CLASS ANALYSIS;
*DEFINE THE BLOCK OF PARAMETER BOUNDS;

bounds=j(2,2*&numvar*&numcl,.);

* SPECIFY POSITIVE BOUNDS;

bounds[1,1:2*&numvar*&numcl]=1.e-6;

* DEFINE THE SUBROUTINE OF THE OBJECTIVE FUNCTION;

start F_objective(x) global (a, obsf, pjoint, qjoint);

  %expf;
  expf=pjoint+qjoint;
  ins1= obsf/expf;
  ins2=log(ins1);
  g=obsf#ins2;
  gsquare=g[+,];
  return(gsquare); * DEFINE THE OBJECTIVE FUNCTION AS G-SQUARE/2;

finish F_objective;

start C_nlin(x) global(a, obsf,pjoint,qjoint);
  %expf;
  expf=pjoint+qjoint;
  Tot_expf=expf[+,]; * AGGREGATE THE EXPECTED FREQUENCY;
  Tot_obsf=obsf[+,]; * AGGREGATE THE OBSERVED FREQUENCY;
  c=Tot_obsf-Tot_expf; * THE TOTAL EXPECTED FREQUENCY IS
EQUAL TO THE TOTAL OBSERVED FREQUENCY;
  return (c); * APPLY NONLINEAR CONSTRAINTS;
finish C_nlin;

```


Appendix: The SAS Program to Compute Pi-star with the Cheat4 Data (continued)

```

* NLP PROCEDURE;

x=j(1,2*&numvar*&numcl,&start);      * EXTRACT START VALUES;
optn= j(1,11,.);                    * DEFINE THE VECTOR OF NLP OPTIONS;
optn[1]=0;                          * SPECIFY A MINIMIZATION FOR THE OBJECTIVE FUNCTION;
optn[2]=2; * SPECIFY THE AMOUNT OF OUTPUT PRINTED BY THE SUBROUTINES;

* WHEN OPTIONS 4-8 ARE SPECIFIED, MORE FUNCTION CALLS MAY BE REQUIRED TO
OBTAIN ACCURATE ESTIMATES;

optn[4]=3; * DEFINE THE UPDATE TECHNIQUE FOR (DUAL) QUASI-NEWTON AND
CONJUGATE GRADIENT TECHNIQUES;
optn[5]=7; * DEFINE THE LINE-SEARCH TECHNIQUE FOR THE NLPQN SUBROUTINE;
optn[6]=1; * DEFINE THE VERSION OF THE ALGORITHM USED TO UPDATE THE
VECTOR OF THE LAGRANGE MULTIPLIERS;
optn[7]=1; * DEFINE THE TYPE OF START MATRIX, G(0),USED FOR THE HESSIAN
APPROXIMATION;
optn[8]=21; * DEFINE THE TYPE OF FINITE DIFFERENCE APPROXIMATION;

* NUMBER OF NONLINEAR CONSTRAINTS;

optn[10]=1;      * SPECIFY TOTAL NUMBER OF NONLINEAR CONSTRAINTS;
optn[11]=1;     * SPECIFY NUMBER OF EQUALITY CONSTRAINTS;

* MAXIMUM NUMBER OF ITERATIONS AND FUNCTION CALLS;

tc=j(1,10,.);
tc[1]=800;
tc[2]=1000;

* ADD TITLES FOR THE LATENT CLASS ANALYSIS;

title 'Computing Pi-star, the Two-Point Mixture Fit Index';
title2 'The First Stage: Latent Class Analysis';
title3 'The Objective Function Is to Minimize G-square';

* CALL NLPQN;

call nlpqn(rc, xr, "F_objective",x,optn,bounds) nlc="C_nlin" tc=tc;

* AGGREGATE THE TOTAL EXPECTED FREQUENCY AND COMPUTE THE LC PROPORTIONS;

%macro tef;
expf=pjoint+qjoint;
Tot_expf=expf[+,];
ppjoint=pjoint[+,];
qqjoint=qjoint[+,];
prop1=ppjoint/Tot_expf;
prop2=qqjoint/Tot_expf;
pistar=1-Tot_expf/&numsap;
%mend;

* RUN THE MARCO;

%tef;

```

PI-LCA SAS PROGRAM FOR TWO-CLASS LCA DICHOTOMOUS VARIABLE MODELS

Appendix: The SAS Program to Compute Pi-star with the Cheat4 Data (continued)

```

* CREATE A MARCO TO COMPUTE CONDITIONAL PROBABILITIES;

%macro cp;
xr=xr`;
x1=xr[1:&numvar,]; x2=xr[&numvar+1:2*&numvar,];
x3=xr[2*&numvar+1:3*&numvar,]; x4=xr[3*&numvar+1:4*&numvar,];
p1=x1/(x1+x2); p2=x2/(x1+x2);
p3=x3/(x3+x4); p4=x4/(x3+x4);
cp=(p1//p2) || (p3//p4);
nlp_par=xr;
cn=1:&numvar;
cn=cn`;
cp1=cn || cp[1:&numvar,]; cp2=cn || cp[&numvar+1:2*&numvar,];
%mend;

* RUN THE MARCO;

%cp;

* PRINT OUTPUT;

Print 'Latent Class Analysis';
print obsf [label='Observed Frequency'] expf [label='Expected
Frequency'];
Print Tot_expf [label='Total Expected Frequency'];
print prop1 [label='LC1 Proportion'] prop2 [label='LC2 Proportion'];
print cp1 [label='Conditional Probabilities' rowname='CP Positive
Response (1)'];
print cp2 [label=' ' rowname='CP Negative Response (2)'];

*****
* SECTION 4: THE SECOND STAGE - COMPUTE PISTAR;

* REDEFINE THE SUBROUTINE OF THE OBJECTIVE FUNCTION TO MAXIMIZE THE
EXPECTED FREQUENCY;

start F_objective(x) global (a, pjoint,qjoint);
  %expf;
  expf=pjoint+qjoint;
  Tot_expf=expf[+,]; * AGGREGATE THE EXPECTED FREQUENCY;
  return(Tot_expf); * REDEFINE THE OBJECTIVE FUNCTION AS TOTAL
EXPECTED FREQUENCY;
finish F_objective;

* REDEFINE THE SUBROUTINE OF NONLINEAR CONSTRAINTS;

start C_nlin(x) global(a,obsf,pjoint,qjoint);
  %expf;
  expf=pjoint+qjoint;
  c=obsf-expf; *FOR EACH RESPONSE VECTOR, THE EXPECTED
FREQUENCY IS EQUAL TO OR SMALLER THAN THE OBSERVED FREQUENCY;
  return (c); * APPLY NONLINEAR CONSTRAINTS;
finish C_nlin;

```

Appendix: The SAS Program to Compute Pi-star with the Cheat4 Data (continued)

```

* CALL NLP PROCEDURE;

x=xr;          * EXTRACT START VALUES;
optn= j(1,11,.); * DEFINE THE VECTOR OF NLP OPTIONS;
optn[1]=1;     * SPECIFY A MAXIMIZATION FOR THE OBJECTIVE FUNCTION;
optn[2]=2;     * SPECIFY THE AMOUNT OF OUTPUT PRINTED BY THE SUBROUTINES;
optn[10]=2**&numvar; * SPECIFY TOTAL NUMBER OF NONLINEAR CONSTRAINTS;
optn[11]=0;    * SPECIFY NUMBER OF EQUALITY CONSTRAINTS;

* ADD TITLES FOR THE PI-STAR COMPUTATION;

title 'Computing Pi-star, the Two-Point Mixture Fit Index';
title2 'The Second Stage: Pi Optimization';
title3 'The Objective Function Is to Maximize the Total Expected
Frequency';

* CALL NLPQN;

call nlpqn(rc, xr, "F_objective",x,optn,bounds) nlc="C_nlin";

* RUN THE MACRO TO AGGREGATE THE TOTAL EXPECTED FREQUENCY;

%tef;

* RUN THE MACRO TO COMPUTE THE CONDITIONAL PROBABILITIES;

%cp;

* PRINT OUTPUT;

Print " Pi-Star Results";
print obsf [label='Observed Frequency'] expf [label='Expected
Frequency'];
Print Tot_expf [label='Total Expected Frequency'];
print Pistar[label='Pi-Star'];
print prop1 [label='LC1 Proportion'] prop2 [label='LC2 Proportion'];
print cp1 [label='Conditional Probabilities' rowname='CP Positive
Response (1)'];
print cp2 [label=' ' rowname='CP Negative Response (2)'];

* EXIT SAS IML;

quit;

* CLOSE NLP MACRO;

%Mend;

* RUN MACRO;

%TwoclassPistar;

run;

```