DEPARTMENT OF STATISTICS
The Pennsylvania State University
University Park, PA 16802 U.S.A.

TECHNICAL REPORTS AND PREPRINTS

Number 12-03 : September 2012

LCCA Package for R, Version 1 (Beta)

Joseph L. Schafer[*] and Joseph Kang[**]

*Center for Statistical Research & Methodology, U. S. Census Bureau, Washington, DC

**Joseph Kang, Department of Preventive Medicine, Feinberg School of Medicine, Northwestern University, Chicago, IL

LCCA Package for R, Version 1 (Beta)

Joseph L. Schafer
Center for Statistical Research & Methodology
U.S. Census Bureau
4600 Silver Hill Rd
Washington, DC 20233
Joseph.L.Schafer@census.gov

Joseph Kang
Department of Preventive Medicine
Feinberg School of Medicine
Northwestern University
Chicago, IL 60611
joseph-kang@northwestern.edu

September 19, 2010

This package performs latent-class causal analysis (LCCA) as described by Kang and Schafer (submitted). LCCA estimates population average treatent effects under an extended version of the Rubin causal model, where the "treatment" is a polytomous latent variable measured by items assumed to be independent within classes. This package also has functions for conventional latent-class analysis with and without covariates. The modeling functions will accept data from surveys collected under the general class of with-replacement (WR) designs, which encompasses simple random samples, stratified samples, cluster samples and multistage designs with equal or unequal probabilities of selection at any stage. Pseudo-maximum likelihood (PML) estimates are computed by an EM algorithm, and standard errors are obtained by a Taylor linearization (sandwich) formula.

This work was supported by National Institute of Child Health and Human Development (NICHD) 1-R03-HD060659, and by National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) 1-R21-DK082858.

1 Overview

1.1 Description

This R package provides functions for latent-class analysis and latent-class causal analysis.

Latent-class analysis (LCA) describes relationships among a set of categorical variables by assuming that they are conditionally independent given an unobserved categorical variable. The LCA model is often attributed to Lazarsfeld and Henry (1968), and procedures for maximum-likelihood (ML) estimation were described by Goodman (1974) and Clogg and Goodman (1984). Properties and applications of LCA have been described by many authors; a good example is the recent textbook by Collins and Lanza (2010). LCA was previously implemented in another R package called `poLCA` (Linzer & Lewis, 2007). Our implementation of LCA differs from `poLCA`'s in the following respects.

- Most computations are performed in native Fortran.
- Standard errors for parameters are computed by several different methods.
- The functions for latent-class modeling support multi-group analyses, which are useful for examining questions of measurement invariance.
- The modeling functions accept survey weights, computing pseudo-maximum likelihood (PML) estimates for model parameters.
- The modeling functions accept identifiers for sampling strata and primary clusters under the general class of with-replacement (WR) survey designs. Standard errors are computed using a linearization (sandwich) method.

Latent-class causal analysis (LCCA) is a new procedure for estimating average treatment effects from observational (non-randomized) studies in which the treatment variable is imperfectly measured (Kang & Schafer, submitted). LCCA combines aspects of latent-class analysis with Rubin's causal model (Rubin, 1974; 2005).

The LCCA package includes three major modeling functions:

`lca`: Fit a conventional latent-class model.

`lcacov`: Fit a latent-class model with covariates.

`lcca`: Fit a latent-class causal model.

Results from these functions are returned as objects of class `"lca"`, `"lcacov"` and `"lcca"`, respectively. These objects contain parameter estimates, standard errors and fit statistics. Nicely formatted printed summaries can be displayed by calling the generic method `summary`. Other important functions include:

`permute.class`: Reorder the latent classes in a result from `lca`, `lcacov` or `lcca`.

`compare.fit`: Compare the fit of nested models from `lca`, `lcacov` or `lcca`.

Three functions are provided for data generation, which are useful for performing simulation studies.

`lca.datasim`: Simulate random data from a latent-class model.

`lcacov.datasim`: Simulate random data from a latent-class model with covariates.

`lcca.datasim`: Simulate random data from a latent-class causal model.

The LCCA package also includes four example datasets:

`hivtest`: diagnostic accuracy of tests for HIV infection

`abortion`: attitudes on legalized abortion from the 2006 General Social Survey

`NHsmoking`: recent cigarette use from NHANES 2005–2006

`diet`: simulated study of the effects of dieting on emotional distress

1.2 Limitations and use

A limitation of the LCCA package is that *it currently works only with R for Windows.* This limitation arises because the native computational routines are written in Fortran 95. Unfortunately, the tools for building R packages in the standard way still require Fortran source code to be written in old-fashioned Fortran 77. Until the R package-building machinery can accommodate newer features of Fortran, we cannot submit this software to CRAN as a platform-independent R package. Rather, we are distributing it ourselves as a precompiled binary package for Windows versions of R.

This software is provided in good faith to researchers free of charge and may be used by anyone if proper credit is given. It is distributed in the hope that it will be useful, but without any warranty, without even the implied warranty of merchantability or fitness for a particular purpose. The recommended citation is:

> Schafer, J.L. and Kang, J. (2012) LCCA: Latent-class causal analysis. Software package for R. University Park, PA: Department of Statistics, The Pennsylvania State University.

1.3 Installing the package

The LCCA package is distributed in precompiled form for Windows as a single compressed archive (`.zip` file). For version 1.0.0, the name of the file is `lcca_1.0.0.zip`. The easiest way to install the package is to begin a Windows R session and select "Install package(s) from local zip files..." from the "Packages" menu. A file selection box will appear from which you can browse your computer. When you select the `.zip` file, the package is installed automatically.

A note to users of Windows Vista. If you are running Windows Vista, you may encounter difficulty when trying to install LCCA for the following reason. R itself has probably been installed in a subdirectory of `C:\Program Files` or `C:\Program Files (x86)`. By default, packages are installed in subdirectory `library` of the R directory. As a Vista user, you may not have sufficient privileges to create new subdirectories and install files there. If you have trouble installing the package under Vista, try running R with Administrator privileges. To do this, *right-click* an R shortcut and select "Run as administrator". If you start R in this way, you should be able to install packages without any trouble. Once the package has been installed, you do not need Administrator privileges to use it. So you may quit that R session and start another one in the usual way, not running as Administrator.

1.4 Loading the package

Before you can use any of LCCA's functions or datasets, you will have to load it into your current session by issuing the command

```
> library(lcca)
```

from the R console. Alternatively, you can go to the "Packages" menu and select "Load packages..." A small window will appear that lists in alphabetical order all of the packages that have been installed on your computer. Select `lcca` from that list, and the package will load.

1.5 Documentation and data examples

Once the library has been loaded, you can view its documentation files in the usual way, as in the following examples:

```
> help(lcacov)            # documentation for the function lcacov
> ?lcacov                 # same thing as above
> help("lcca-package")    # overview of the package
```

If you issue the command

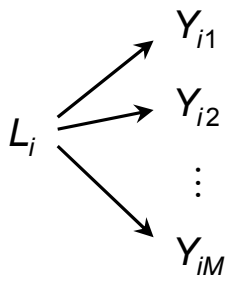

Figure 1: *Standard latent-class analysis.*

```
> data(package="lcca")
```

then a list of the datasets distributed with LCCA will appear. Each dataset has its own documentation; for example,

```
> help(hivtest)
```

will display the page for the `hivtest` dataset. To gain access to these datasets, use the `data` function with the name of the dataset as its argument. For example, if you type

```
> data(hivtest)
```

then a copy of the `hivtest` dataset will be loaded into your current workspace as a data frame.

2 Latent class analysis (LCA)

2.1 Notation and assumptions

Variables. Let $\boldsymbol{Y}_i = (Y_{ii}, \ldots, Y_{iM})$ denote a vector of manifest categorical items for individual i. We will assume that Y_{im} takes possible values $1, \ldots, r_m$, which are called the response categories. The realized value for $\boldsymbol{Y}_i$ is denoted by $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{iM})$. In the standard C-class model, we assume that $Y_{i1}, \ldots, Y_{iM}$ are conditionally independent given a latent categorical variable L_i, which takes possible values (classes) $1, \ldots, C$. A graphical representation of the LCA model is shown in Figure 1.

Measurement parameters. Our LCA implementation allows for multigroup analyses in which individuals are classified into groups $g = 1, \ldots, G$ across which the measurement parameters and/or class-membership probabilities may vary (Clogg & Goodman, 1984). Let g_i denote the group to which individual i belongs. The measurement parameters are

$$\rho_{mr|cg} = \Pr(Y_{im} = r \mid L_i = c, g_i = g).$$

These parameters satisfy the obvious constraints $\sum_{r=1}^{r_m} \rho_{mr|cg} = 1$ for each combination of $m = 1, \ldots, M$, $c = 1, \ldots, C$ and $g = 1, \ldots, G$. If the ρ's are assumed to be equal across groups, a condition known as measurement invariance, then the number of free measurement parameters is $C \sum_{m=1}^{M}(r_m - 1)$. If the ρ's are not assumed to be equal across groups, then the number of free measurement parameters is $GC \sum_{m=1}^{M}(r_m - 1)$.

Class prevalences. The prevalences or class-membership probabilities are

$$\gamma_{c|g} = \Pr(L_i = c \,|\, g_i = g).$$

If the γ's are assumed to be equal across groups, then the number of free γ-parameters is $(C - 1)$. If we do not assume equality across groups, it becomes $G(C - 1)$.

Distributions. The model represented by Figure 1 can be written as

$$\Pr(\boldsymbol{Y}_i = \boldsymbol{y}_i, L_i = c \,|\, g_i = g) = \gamma_{c|g} \prod_{m=1}^{M} \prod_{r=1}^{r_m} \rho_{mr|cg}^{I(y_{im}=r)},$$

or as

$$\Pr(\boldsymbol{Y}_i = \boldsymbol{y}_i, L_i \,|\, g_i = g) = \prod_{c=1}^{C} \gamma_{c|g}^{I(L_i=c)} \prod_{c=1}^{C} \prod_{m=1}^{M} \prod_{r=1}^{r_m} \rho_{mr|cg}^{I(L_i=c)\, I(y_{im}=r)}.$$

Marginalizing over the latent variable gives

$$\Pr(\boldsymbol{Y}_i = \boldsymbol{y}_i \,|\, g_i = g) = \sum_{c=1}^{C} \gamma_{c|g} \prod_{m=1}^{M} \prod_{r=1}^{r_m} \rho_{mr|cg}^{I(y_{im}=r)}.$$

Missing items. Suppose that some of the items in $\boldsymbol{Y}_i$ are missing. Let $\boldsymbol{Y}_{i,obs}$ denote the observed part of $\boldsymbol{Y}_i$, and let $\boldsymbol{y}_{i,obs}$ denote its realized value. The marginal distribution of $\boldsymbol{Y}_{i,obs}$ can be written as

$$\Pr(\boldsymbol{Y}_{i,obs} = \boldsymbol{y}_{i,obs} \,|\, g_i = g) = \sum_{c=1}^{C} \gamma_{c|g} \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|cg}^{I(y_{im}=r)},$$

where obs_i denotes the subset of $\{1, \ldots, M\}$ corresponding to the items that are observed for individual i (Little & Rubin, 2002).

Frequencies for aggregated data. If multiple individuals in the sample have identical values for $\boldsymbol{Y}_{i,obs}$, these individuals may be aggregated into a single case (line) in the data file, with with a frequency f_i indicating how many individuals are represented by that case. Then the total sample size is not n but $\sum_{i=1}^{n} f_i$. For non-aggregated data, define $f_i = 1$ for $i = 1, \ldots, n$.

2.2 Estimation procedure

Loglikelihood. Let $\boldsymbol{\theta}$ denote the free parameters (i.e. the non-redundant ρ's and γ's) to be estimated. If we assume that any missing elements of $\boldsymbol{Y}_i$ are missing at random, then the maximum-likelihood (ML) estimator of $\boldsymbol{\theta}$ will maximize the loglikelihood function that

ignores the missing-data mechanism (Little & Rubin, 2002), which we call the observed-data loglikelihood. Let $l_i(\boldsymbol{\theta})$ denote the contribution of individual or case i to the observed-data loglikelihood. The observed-data loglikelihood is

$$\begin{aligned} l(\boldsymbol{\theta}) &= \sum_{i=1}^{n} l_i(\boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} \log \Pr(\boldsymbol{Y}_{i,obs} = \boldsymbol{y}_{i,obs} \,|\, g_i = g)^{f_i} \\ &= \sum_{g=1}^{G} \sum_{i \in g} f_i \log \left(\sum_{c=1}^{C} \gamma_{c|g} \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|cg}^{I(y_{im}=r)} \right). \end{aligned} \tag{1}$$

The function (1) is difficult to maximize directly, so we accomplish it indirectly by an EM algorithm. At each cycle of EM, we maximize the expected value of the loglikelihood function that augments the observed data with the latent variable L_i,

$$\begin{aligned} l^*(\boldsymbol{\theta}) &= \sum_{i=1}^{n} l_i^*(\boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} \log \Pr(\boldsymbol{Y}_{i,obs} = \boldsymbol{y}_{i,obs}, L_i \,|\, g_i = g)^{f_i} \\ &= \sum_{g=1}^{G} \sum_{i \in g} f_i \log \left(\prod_{c=1}^{C} \gamma_{c|g}^{I(L_i=c)} \prod_{c=1}^{C} \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|cg}^{I(L_i=c)\, I(y_{im}=r)} \right) \\ &= \sum_{g=1}^{G} \sum_{i \in g} \sum_{c=1}^{C} f_i I(L_i = c) \log \gamma_{ic} \\ &\quad \times \sum_{g=1}^{G} \sum_{i \in g} \sum_{c=1}^{C} \sum_{m \in obs_i} \sum_{r=1}^{r_m} f_i I(L_i = c)\, I(y_{im} = r) \log \rho_{mr|cg}. \end{aligned} \tag{2}$$

This is done with an Expectation or E-step followed by a Maximization or M-step.

E-step. In the E-step, we compute the expectation of (2) with respect to the distribution of the missing data given the observed data, fixing the unknown parameters at their current estimates. Because (2) is a linear function of the indicators $I(L_i = c)$, we replace these indicators by their expectations, which are the posterior probabilities of class membership given the observed items. The posterior probabilities are

$$\begin{aligned} \eta_{ic} &= \Pr(L_i = c \,|\, \boldsymbol{Y}_{i,obs} = \boldsymbol{y}_{i,obs}, g_i = g) \\ &= \frac{\gamma_{c|g} \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|cg}^{I(y_{im}=r)}}{\sum_{c'=1}^{C} \gamma_{c'|g} \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|c'g}^{I(y_{im}=r)}}. \end{aligned} \tag{3}$$

The E-step consists of computing the vector of posterior probabilities $(\eta_{i1}, \ldots, \eta_{iC})$ for $i = 1, \ldots, n$ based on the current estimates of the γ's and ρ's.

M-step. For the M-step, note that the expectation of (2) is the sum of two terms, one depending on the ρ's, the other depending on the γ's. The overall maximum with respect to

$\boldsymbol{\theta}$ is achieved by maximizing the two terms separately. If we do not assume equality of ρ's across groups, then the maximizer with respect to the ρ's is

$$\begin{aligned}\hat{\rho}_{mr|cg} &= \frac{\sum_{i\in obs_{(g,m)}} f_i\,\eta_{ic}\,I(y_{im}=r)}{\sum_{r'=1}^{r_m}\sum_{i\in obs_{(g,m)}} f_i\,\eta_{ic}\,I(y_{im}=r')} \\ &= \frac{\sum_{i\in obs_{(g,m)}} f_i\,\eta_{ic}\,I(y_{im}=r)}{\sum_{i\in obs_{(g,m)}} f_i\,\eta_{ic}}, \quad r=1,\ldots,r_m,\end{aligned}$$

for each combination of group g, class c and item m, where $obs_{(g,m)}$ denotes the subset of cases within group g for which Y_{im} is non-missing. If we do assume equality of ρ's across groups,

$$\rho_{mr|c1} = \rho_{mr|c2} = \cdots, \rho_{mr|cG} = \rho_{mr|c},$$

then the maximum for the ρ's occurs at

$$\begin{aligned}\hat{\rho}_{mr|c} &= \frac{\sum_{i\in obs_{(m)}} f_i\,\eta_{ic}\,I(y_{im}=r)}{\sum_{r'=1}^{r_m}\sum_{i\in obs_{(m)}} f_i\,\eta_{ic}\,I(y_{im}=r')} \\ &= \frac{\sum_{i\in obs_{(m)}} f_i\,\eta_{ic}\,I(y_{im}=r)}{\sum_{i\in obs_{(m)}} f_i\,\eta_{ic}}, \quad r=1,\ldots,r_m\end{aligned}$$

for each combination of class c and item m, where $obs_{(m)}$ denotes the subset of cases for which Y_{im} is non-missing. If we do not assume equality of the γ's across groups, then the maximum with respect to the γ's occurs at

$$\hat{\gamma}_{c|g} = \frac{\sum_{i\in g} f_i\eta_{ic}}{\sum_{c'=1}^{C}\sum_{i\in g} f_i\eta_{ic'}}$$

for $c = 1,\ldots,C$ and $g = 1,\ldots,G$. If the γ's are constrained to be equal across groups, it becomes

$$\hat{\gamma}_{c|g} = \hat{\gamma}_c = \frac{\sum_{i=1}^{n} f_i\eta_{ic}}{\sum_{c'=1}^{C}\sum_{i=1}^{n} f_i\eta_{ic'}}.$$

Starting values. The loglikelihood function for this model is invariant to reordering of the classes. This means that, depending on the starting values, EM may converge to any one of $C!$ equivalent modes in which the class labels $1, 2, \ldots, C$ have been permuted.

Our implementation gives users the option of providing starting values for the ρ's and/or γ's. If none are given, starting values are randomly generated from uniform distributions subject to the usual sum-to-one constraints. In multigroup analyses, identical random starting values are applied to each group.

For some datasets and models, EM may converge to a local minor mode. Users are advised to repeat the estimation procedure from a variety of random starting values and compare the loglikelihoods at the solutions to determine if minor modes are present.

Boundary solutions. In applications of LCA, it is not unusual for estimates of some γ or ρ-parameters to approach zero, which puts them on a boundary of the parameter space. A

zero value for a γ, which corresponds to an empty class, may suggest that the number of classes should be reduced. Empty-class solutions may also be local minor modes resulting from "bad" (i.e., very implausible) random starting values. A zero value for a ρ-parameter indicates that no individuals within a class provide the given response to an item.

Flattening constants. A solution at or near a boundary can make it impossible to compute standard errors in the usual fashion. When the standard error procedures fail, we recommend the use of flattening constants for the ρ's and/or γ's. A flattening constant introduces information about each set of probabilities that sums to one. The constant k adds information equivalent to k prior observations (individuals), spread equally across the categories. A small positive value such as $k = 1$ is often sufficient to nudge the solution away from the boundary, allowing standard errors to be computed. When $k > 0$, the EM algorithm maximizes a function equal to the loglikelihood plus a penalty term. The penalty may be regarded as a log-prior density function, and the resulting estimate may be regarded as a Bayesian posterior mode. The default value of $k = 0$, which corresponds to no flattening, is equivalent to a uniform prior distribution and leads to ML estimates.

2.3 Standard errors

Loglikelihood derivatives. Suppose we collect the nonredundant free parameters of the LCA model into a single parameter vector $\boldsymbol{\theta}$. The contribution of case i in group g to the observed-data loglikelihood function $l(\boldsymbol{\theta})$ is

$$l_i(\boldsymbol{\theta}) \;=\; f_i \log \left(\sum_{c=1}^{C} \gamma_{c|g} \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|cg}^{I(y_{im}=r)} \right).$$

Denote the vector of first derivatives of l_i by

$$l_i'(\boldsymbol{\theta}) \;=\; \frac{\partial}{\partial \boldsymbol{\theta}}\, l_i(\boldsymbol{\theta}) \;=\; \boldsymbol{\psi}_i(\boldsymbol{\theta});$$

this is also called the *score function* for case i. Denote the matrix of second derivatives by

$$l_i''(\boldsymbol{\theta}) \;=\; \frac{\partial^2}{\partial \boldsymbol{\theta}\, \partial \boldsymbol{\theta}^T}\, l_i(\boldsymbol{\theta}) \;=\; \boldsymbol{\psi}_i'(\boldsymbol{\theta}) \;=\; \frac{\partial}{\partial \boldsymbol{\theta}^T}\, \boldsymbol{\psi}_i(\boldsymbol{\theta}).$$

Denote the total score vector by

$$\boldsymbol{\psi}(\boldsymbol{\theta}) \;=\; \sum_{i=1}^{n} \boldsymbol{\psi}_i(\boldsymbol{\theta}) \;=\; \sum_{g=1}^{G} \sum_{i \in g} \boldsymbol{\psi}_i(\boldsymbol{\theta}).$$

If a solution $\hat{\boldsymbol{\theta}}$ lies in the interior of the parameter space, then it should satisfy $\boldsymbol{\psi}(\hat{\boldsymbol{\theta}}) = 0$. But if the estimate lies on a boundary, then some elements of $\boldsymbol{\psi}(\hat{\boldsymbol{\theta}})$ may be nonzero.

Our implementation computes standard errors by three different methods. These procedures will fail if $\hat{\boldsymbol{\theta}}$ lies on or near a boundary or if the model is under-identified.

Standard method. The default method for computing standard errors, which we call "standard," approximates the variance of $\hat{\boldsymbol{\theta}}$ by

$$V(\hat{\boldsymbol{\theta}}) \approx \left(-\sum_{i=1}^{n} \boldsymbol{\psi}_i'(\hat{\boldsymbol{\theta}})\right)^{-1} = \left(-\sum_{i=1}^{n} \hat{\boldsymbol{\psi}}_i'\right)^{-1}, \tag{4}$$

where $\hat{\boldsymbol{\psi}}_i'$ is shorthand for $\boldsymbol{\psi}_i'(\hat{\boldsymbol{\theta}})$. If flattening constants are used, the second derivatives of the penalty function $q(\boldsymbol{\theta})$ are included before the inverse is taken,

$$V(\hat{\boldsymbol{\theta}}) \approx \left(-\sum_{i=1}^{n} \boldsymbol{\psi}_i'(\hat{\boldsymbol{\theta}})\right)^{-1} = \left(-\sum_{i=1}^{n} \hat{\boldsymbol{\psi}}_i' - q''(\hat{\boldsymbol{\theta}})\right)^{-1}.$$

Fast method. The "fast" method, which uses only the first derivatives, approximates the variance of $\hat{\boldsymbol{\theta}}$ by

$$V(\hat{\boldsymbol{\theta}}) \approx \left(\sum_{i=1}^{n} f_i^{-1} \boldsymbol{\psi}_i(\hat{\boldsymbol{\theta}})\, \boldsymbol{\psi}_i(\hat{\boldsymbol{\theta}})^T\right)^{-1} = \left(\sum_{i=1}^{n} f_i^{-1} \hat{\boldsymbol{\psi}}_i \hat{\boldsymbol{\psi}}_i^T\right)^{-1}, \tag{5}$$

where $\hat{\boldsymbol{\psi}}_i$ is shorthand for $\boldsymbol{\psi}_i(\hat{\boldsymbol{\theta}})$. The factor f_i^{-1} inside of the sum is necessary to ensure that results are the same whether data are aggregated or not. This form is consistent with an asymptotic sequence in which the individuals are exchangeable and the total number of individuals (rather than the total number of cases, if the data are aggregated) goes to infinity. Flattening constants, if present, are not used in the computation of these standard errors; the penalty function is ignored.

Sandwich method. The "sandwich" method approximates the variance of $\hat{\boldsymbol{\theta}}$ by

$$V(\hat{\boldsymbol{\theta}}) \approx \left(-\sum_{i=1}^{n} \hat{\boldsymbol{\psi}}_i'\right)^{-1} \left(\sum_{i=1}^{n} f_i^{-1} \hat{\boldsymbol{\psi}}_i \hat{\boldsymbol{\psi}}_i^T\right) \left(-\sum_{i=1}^{n} \hat{\boldsymbol{\psi}}_i'\right)^{-1}. \tag{6}$$

In other types of statistical models (e.g., regression analyses), standard errors computed by the sandwich method are sometimes called "empirical" or "robust." If a penalty function is present, the sandwich estimator becomes

$$V(\hat{\boldsymbol{\theta}}) \approx \left(-\sum_{i=1}^{n} \hat{\boldsymbol{\psi}}_i' - q''(\hat{\boldsymbol{\theta}})\right)^{-1} \left(\sum_{i=1}^{n} f_i^{-1} \hat{\boldsymbol{\psi}}_i \hat{\boldsymbol{\psi}}_i^T\right) \left(-\sum_{i=1}^{n} \hat{\boldsymbol{\psi}}_i' - q''(\hat{\boldsymbol{\theta}})\right)^{-1}.$$

3 Example: HIV immunoassay

3.1 Fitting a latent-class model

Yang and Becker (1997) described a study to assess the accuracy of diagnostic tests for human immunodeficiency virus (HIV) infection. Four tests were applied to each of 428 high-risk patients. The data, which have been provided in the dataset `hivtest`, are in aggregated form:

```
> data(hivtest)
> hivtest
  A B C D COUNT
1 1 1 1 1   170
2 1 1 1 2    15
3 1 2 1 1     6
4 2 1 1 1     4
5 2 1 1 2    17
6 2 1 2 2    83
7 2 2 1 1     1
8 2 2 1 2     4
9 2 2 2 2   128
```

The four tests are labeled `A`, `B`, `C`, and `D`. For each test, `1` indicates a positive result and 2 indicates a negative result. None of these tests can be considered a gold standard, and the authors applied latent-class analysis to estimate the sensitivity and specificity of each test. There are presumably two latent classes corresponding to the true positives and true negatives.

3.2 Using the lca function

The syntax of the `lca` function is shown below.

```
lca( formula, data, freq, groups, nclass = 2,
   constrain.rhos = F, constrain.gammas = F, iseeds = NULL,
   iter.max = 5000, tol = 1e-06, starting.values = NULL,
   flatten.rhos = 0, flatten.gammas = 0, se.method = "STANDARD",
   weights, clusters, strata, subpop)
```

The only required argument is `formula`. This is an R object of class `"formula"` which determines the model to be fit. It is similar to the formulas used in the familiar R regression procedures `lm` and `glm`. Unlike standard regression models, however, an LCA model is a multivariate model with multiple response variables. The response is actually a matrix whose columns are polytomous items. The two-class model may be fit like this:

```
> set.seed(123)  # to reproduce these results
> fit <- lca( cbind(A,B,C,D)~1, freq=COUNT, nclass=2, data=hivtest)
```

Some details of this syntax are explained below.

- In the model formula `cbind(A,B,C,D)~1`, The expression term on the left-hand side of `~` must be a matrix (not a data frame) of variables to be used as responses. Each response variable should consist of integer codes `1,2,...`. Response variables may also be factors, in which case they will be automatically converted to integer codes (as in the function `unclass`), and the levels of the factors will be ignored. The right-hand side of formula should be `1`, indicating that the only predictor is a constant; any other predictors in the model formula will be ignored.

- Missing values in response variables are allowed and should be conveyed by the R missing value code `NA`. Cases with missing responses are retained in the fitting procedure, and the missing values are assumed to be ignorably missing or missing at random.
- The number of latent classes to be fit is determined by `nclass`, with `nclass=1` indicating that the response variables are jointly independent.
- By default, each case (row) of data or the model environment is assumed to represent one observational unit or individual. Data may also be aggegated, with individuals bearing identical responses to all variables (including `NA`'s, if present) collapsed into a single case, with frequencies conveyed through the numeric variable `freq`.
- By default, `lca` generates random starting values for the model parameters. The function uses its own internal random number generator which is seeded by a pair of integers through the optional argument `seeds` (for example, `seeds=c(123,456)`), which allows results to be reproduced in the future. If `seeds` is not provided, then the function will seed itself with two random integers from R. Therefore, the results can also be made reproducible by calling the R function `set.seed` beforehand.

3.3 Displaying the results

The result from `lca` is an object of class `"lca"` which contains a large number of components. To see a nicely formatted set of results, apply the `summary` command:

```
> summary(fit)

        Summary of Latent-Class Analysis

==================================================
Fit statistics
==================================================

The EM algorithm CONVERGED in: 13 iterations

Standard errors computed successfully.
Standard-error method:   STANDARD

Number of free parameters estimated:      9.0000000
Loglikelihood:                         -629.8826889
Loglikelihood + penalty:               -629.8826889
-2 * Loglikelihood:                    1259.7653778
AIC (smaller is better):               1277.7653778
BIC (smaller is better):               1314.2974866

==================================================
Parameter estimates
==================================================
```

```
Class prevalences (gammas):
Class:       1       2
        0.4599  0.5401

Item-response probabilities (rhos):
   Response category 1
Class:       1       2
   A    0.9703  0.0000
   B    0.9644  0.4290
   C    1.0000  0.0871
   D    0.9195  0.0000

   Response category 2
Class:       1       2
   A    0.0297  1.0000
   B    0.0356  0.5710
   C    0.0000  0.9129
   D    0.0805  1.0000
```

It is estimated that 46.0% of the individuals fall into Class 1, and 54.0% fall into Class 2. From the estimated measurement parameters (the ρ's), we see that Class 1 contains individuals who are likely to test positive on every test (the true positives), whereas Class 2 contains those who are likely to test negative on every test (the true negatives). The estimated sensitivities for tests A, B, C, and D are 97.0%, 96,4%, 100% and 91.9%, respectively, and the estimated specificities are 100%, 57.1%, 91.3% and 100%.

Different starting values for the parameters may lead to solutions in which the classes have different orderings. You can permute the the classes using the function `permute.class`. For example, this code will display the same information, but with the order of the classes reversed.

```
> fit <- permute.class(fit, c(2,1) )  # reverse the order of the classes
> summary(fit)
```

3.4 Standard errors and boundary solutions

By default, `summary` does not display standard errors for the parameters in an lca object. To see the standard errors, supply the argument `show.all=T`:

```
> summary(fit, show.all=T)

        Summary of Latent-Class Analysis

==================================================
Data and model information
==================================================

Number of cases:   9
```

```
Total frequency for all cases:   428

Number of measurement items:     4
Number of categories per item:   2 2 2 2
Number of latent classes:        2

Starting values for rhos:        randomly generated
Random seed 1:                   288
Random seed 2:                   788
Starting values for gammas:      uniform values

Max. number of EM iterations:    5000
Convergence criterion:           0.000001

==================================================
Fit statistics
==================================================

The EM algorithm CONVERGED in: 13 iterations

Standard errors computed successfully.
Standard-error method:   STANDARD

Number of free parameters estimated:      9.0000000
Loglikelihood:                         -629.8826889
Loglikelihood + penalty:               -629.8826889
-2 * Loglikelihood:                    1259.7653778
AIC (smaller is better):               1277.7653778
BIC (smaller is better):               1314.2974866

==================================================
Parameter estimates
==================================================

Class prevalences (gammas):
Class:       1       2
        0.5401  0.4599

Item-response probabilities (rhos):
   Response category 1
Class:       1       2
   A    0.0000  0.9703
   B    0.4290  0.9644
   C    0.0871  1.0000
   D    0.0000  0.9195

   Response category 2
Class:       1       2
   A    1.0000  0.0297
   B    0.5710  0.0356
   C    0.9129  0.0000
   D    1.0000  0.0805
```

```
Standard errors for class prevalences (gammas):
           Est. Std.Err
Class 1 0.54006 0.02458
Class 2 0.45994 0.02458

Standard errors for item-response probabilities (rhos):
                          Est. Std.Err
Class 1, A, Response 1 0.00000 0.06647
Class 1, A, Response 2 1.00000 0.06647
Class 1, B, Response 1 0.42895 0.03271
Class 1, B, Response 2 0.57105 0.03271
Class 1, C, Response 1 0.08715 0.02029
Class 1, C, Response 2 0.91285 0.02029
Class 1, D, Response 1 0.00000 0.02155
Class 1, D, Response 2 1.00000 0.02155
Class 2, A, Response 1 0.97025 0.01567
Class 2, A, Response 2 0.02975 0.01567
Class 2, B, Response 1 0.96441 0.01463
Class 2, B, Response 2 0.03559 0.01463
Class 2, C, Response 1 1.00000 0.07195
Class 2, C, Response 2 0.00000 0.07195
Class 2, D, Response 1 0.91945 0.02018
Class 2, D, Response 2 0.08055 0.02018
```

To see more options for `summary`, view the help file for the method `summary.lca`.

In this example, the γ parameters appear to have reasonable standard errors, but some of the ρ's do not, because their estimated values are zero and one. When the solution lies on a boundary of the parameter space, the derivatives of the loglikelihood function at the solution are not all zero, and the procedures used to compute standard errors are not trustworthy. You can examine the derivatives yourself, because they are stored in the lca object in a component named `score`.

```
> round(fit$score, 3)
rho[1,1,2,1] rho[2,1,2,1] rho[3,1,2,1] rho[4,1,2,1] rho[1,1,1,1] rho[2,1,1,1]
      -0.001        0.000      196.421        0.000     -222.408        0.000
rho[3,1,1,1] rho[4,1,1,1]   gamma[2,1]
       0.000     -164.457        0.000
```

3.5 Flattening constants

Boundary solutions are extremely common in latent-class analysis. To draw parameter estimates away from the boundary, you may supply flattening constants to `lca` through the arguments `flatten.gammas` and `flatten.rhos`. For small datasets like this one, constants of 1.0 should be sufficient. In the call to `lca` shown below, we use the current parameter estimates as the starting values for a new run of the EM algorithm, this time with flattening constants for γ and ρ.

```
> fit2 <- lca( cbind(A,B,C,D)~1, freq=COUNT, nclass=2, data=hivtest,
+   starting.values=fit$param, flatten.gammas=1, flatten.rhos=1)

> round(fit2$score,3)
rho[1,1,1,1] rho[2,1,1,1] rho[3,1,1,1] rho[4,1,1,1] rho[1,1,2,1] rho[2,1,2,1]
       0.000        0.000        0.000        0.005        0.001        0.000
rho[3,1,2,1] rho[4,1,2,1]   gamma[1,1]
       0.000        0.000        0.000
```

The scores, which are now the derivatives of the loglikelihood plus a penalty function (i.e., a log-posterior density) are nearly zero at this new solution. (Setting a tighter convergence criterion by reducing the size of the argument `tol` would bring them even closer to zero.) With these flattening constants, the parameter estimates have changed very little, but the standard errors are now a bit more reasonable:

```
Standard errors for class prevalences (gammas):
           Est. Std.Err
Class 1 0.54023 0.02422
Class 2 0.45977 0.02422

Standard errors for item-response probabilities (rhos):
                          Est. Std.Err
Class 1, A, Response 1 0.00224 0.00316
Class 1, A, Response 2 0.99776 0.00316
Class 1, B, Response 1 0.42897 0.03259
Class 1, B, Response 2 0.57103 0.03259
Class 1, C, Response 1 0.08939 0.01929
Class 1, C, Response 2 0.91061 0.01929
Class 1, D, Response 1 0.00281 0.00392
Class 1, D, Response 2 0.99719 0.00392
Class 2, A, Response 1 0.96834 0.01373
Class 2, A, Response 2 0.03166 0.01373
Class 2, B, Response 1 0.96272 0.01361
Class 2, B, Response 2 0.03728 0.01361
Class 2, C, Response 1 0.99747 0.00358
Class 2, C, Response 2 0.00253 0.00358
Class 2, D, Response 1 0.91710 0.02023
Class 2, D, Response 2 0.08290 0.02023
```

These are still not entirely believable, because approximate 95% confidence intervals (estimate plus or minus two standard errors) for some of these ρ's still stray outside of the parameter space. When a probability parameter lies close to zero or one, a symmetric interval on the probability scale will have poor repeated-sampling properties, and a symmetric interval on an open-ended scale (e.g., the log odds) will tend to perform better. Computing standard errors for the log-odds is not difficult, but we have chosen not to do this because, in our experience, high-performance confidence intervals for the measurement parameters are rarely needed.

Although the default values for `flatten.gammas` and `flatten.rhos` are zero, which corresponds to no flattening, it is not a bad idea to use positive flattening constants as a matter of routine.

3.6 Information held in the lca object

An lca object holds a large amount of information that is not displayed by `summary.lca`. If you are interested in computational details, or if you need to extract and save information from the model fitting procedure, you can probably find what you need by examining the components of the object.

```
> names(fit2)
 [1] "u"                 "freq"              "groups"
 [4] "nclass"            "constrain.rhos"    "constrain.gammas"
 [7] "starting.values"   "use.startval.rho"  "use.startval.gamma"
[10] "flatten.rhos"      "flatten.gammas"    "se.method"
[13] "weights"           "clusters"          "strata"
[16] "subpop"            "iter.max"          "tol"
[19] "iseeds"            "ncases"            "nitems"
[22] "ngroups"           "nlevs"             "maxlevs"
[25] "dim.theta"         "group.levels"      "stratum.levels"
[28] "cluster.levels"    "class.levels"      "item.names"
[31] "case.names"        "design.stats.int"  "design.stats.real"
[34] "iter"              "converged"         "loglik"
[37] "logpost"           "loglik.final"      "logpost.final"
[40] "AIC"               "BIC"               "param"
[43] "post.probs"        "se.fail"           "se.rho"
[46] "se.gamma"          "theta.names"       "theta"
[49] "score"             "hessian"           "sandwich.meat"
[52] "cov.theta"         "deff.trace"        "msg"
```

The method `summary.fit` retains all the original components of the lca object and adds some more.

```
> fit2 <- summary(fit2)
> names(fit2)
 [1] "u"                 "freq"              "groups"
 [4] "nclass"            "constrain.rhos"    "constrain.gammas"
 [7] "starting.values"   "use.startval.rho"  "use.startval.gamma"
[10] "flatten.rhos"      "flatten.gammas"    "se.method"
[13] "weights"           "clusters"          "strata"
[16] "subpop"            "iter.max"          "tol"
[19] "iseeds"            "ncases"            "nitems"
[22] "ngroups"           "nlevs"             "maxlevs"
[25] "dim.theta"         "group.levels"      "stratum.levels"
[28] "cluster.levels"    "class.levels"      "item.names"
[31] "case.names"        "design.stats.int"  "design.stats.real"
[34] "iter"              "converged"         "loglik"
[37] "logpost"           "loglik.final"      "logpost.final"
[40] "AIC"               "BIC"               "param"
[43] "post.probs"        "se.fail"           "se.rho"
[46] "se.gamma"          "theta.names"       "theta"
[49] "score"             "hessian"           "sandwich.meat"
[52] "cov.theta"         "deff.trace"        "msg"
```

```
[55] "show.header"        "show.data.model"    "show.fit"
[58] "show.param"         "show.se.gamma"      "show.se.rho"
[61] "n.table"            "freq.table"         "model.table"
[64] "fit.table"          "gamma.rho.table"    "gamma.table"
[67] "rho.table"
```

4 Example: Attitudes toward legalized abortion

4.1 The data

The General Social Survey (GSS) tracks attitudes of adults in the United States on a wide range of issues. The dataset `abortion`, which was extracted from the 2006 GSS, reports responses to six questions about legalized abortion. The help file for this dataset is shown below.

```
abortion                package:lcca                R Documentation

Abortion attitudes from the 2006 General Social Survey

Description:

     This dataset, which was extracted from the 2006 General Social
     Survey, report the responses of adults in the United States to six
     questions about legalized abortion. The questions began, ``Please
     tell me whether or not you think it should be possible for a
     pregnant woman to obtain a legal abortion if...''

Usage:

     abortion

Format:

     a data frame with 4510 rows and 8 variables:


     'SEX' respondent's sex (factor with two levels)

     'ABANY' ``...The woman wants it for any reason?'' (factor with
          five levels)

     'ABDEFECT' ``...If there is a strong chance of serious defect in
          the baby?'' (factor with five levels)

     'ABHLTH' ``...If the woman's own health is seriously endangered by
          the pregnancy?'' (factor with five levels)

     'ABNOMORE' ``...If she is married and does not want any more
          children?'' (factor with five levels)
```

'ABPOOR' ``...If the family has a very low income and cannot afford any more children?'' (factor with five levels)

'ABRAPE' ``...If she became pregnant as a result of rape?'' (factor with five levels)

'WTSSNR' analytic weight adjusted for subsampling of initial nonrespondents (numeric)

Details:

Because of the split half-sample design used in the 2006 GSS, only about half of the sampled adults were asked the abortion questions. The response code '"NAP"' (not applicable) indicates that the question was not asked. The other response codes are '"YES"', '"NO"' and '"DK"' (Don't know). A missing value ('NA') indicates that the question was asked but no answer was given. Analysts should recode '"NAP"' to a missing value. Whether '"DK"' should be converted to a missing value is debatable.

Although the GSS has a complex multistage area sampling plan, it was designed to be self-weighting in the sense that every adult in the sample frame had an approximately equal chance of being selected. However, many sampled persons did not respond to the initial interview request. To help reduce nonresponse bias, a random sample of these nonrespondents were selected for aggressive followup attempts. Those who were successfully interviewed in this followup procedure ought to be assigned greater weight, because they need to represent those who were not selected for followup. The variable 'WTSSNR' is a weight that adjusts for this nonresponse followup procedure, and the GSS documentation recommends that this weight be used in analyses.

Latent-class analyses of abortion questions from earlier GSS surveys were reported by McCutcheon (1987) and by McCutcheon and Nawojczyk (1987).

Source:

Davis, J.A. and Smith, T. W. (2007) _General Social Surveys, 1972-2006_ (machine-readable data file). Chicago: National Opinion Research Center (producer). Storrs, CT: The Roper Center for Public Opinion Research, University of Connecticut (distributor).

References:

McCutcheon, A.L. (1987) Sexual morality, pro-life values, and attitudes toward abortion: a simultaneous latent structure analysis for 1978-1983. _Sociological Methods and Research_, 16, 256-275.

McCutcheon, A.L. and Nawojczyk, M. (1995) Making the break:

```
popular sentiment toward legalized abortion among American and
Polish Catholic laities.  _International Journal of Public Opinion
Research_, 7, 232-252.

For example analyses of this dataset using functions in the LCCA
package, see the manual _LCCA Package for R, Version 1_ in the
subdirectory 'doc'.
```

4.2 Comparing two- and three-class solutions

We can fit a two-class model as follows.

```
> # retrieve the data
> data(abortion)
> summary(abortion)
     SEX        ABANY      ABDEFECT     ABHLTH     ABNOMORE     ABPOOR
 FEMALE:2507   DK  :  50   DK  :  68   DK  :  62   DK  :  65   DK  :  63
 MALE  :2003   NAP :2507   NAP :2507   NAP :2507   NAP :2507   NAP :2507
               NO  :1155   NO  : 495   NO  : 233   NO  :1107   NO  :1106
               YES : 784   YES :1425   YES :1692   YES : 818   YES : 822
               NA's:  14   NA's:  15   NA's:  16   NA's:  13   NA's:  12

  ABRAPE         WTSSNR
 DK  :  76   Min.   :0.36
 NAP :2507   1st Qu.:0.50
 NO  : 429   Median :0.86
 YES :1483   Mean   :1.00
 NA's:  15   3rd Qu.:1.10
             Max.   :6.59

> # recode NAP and DK as missing values, and
> # reverse the order of the levels so that 1=YES and 2=NO
> abortion$ABANY <- factor(abortion$ABANY, levels=c("YES","NO") )
> abortion$ABDEFECT <- factor(abortion$ABDEFECT, levels=c("YES","NO") )
> abortion$ABHLTH <- factor(abortion$ABHLTH, levels=c("YES","NO") )
> abortion$ABNOMORE <- factor(abortion$ABNOMORE, levels=c("YES","NO") )
> abortion$ABPOOR <- factor(abortion$ABPOOR, levels=c("YES","NO") )
> abortion$ABRAPE <- factor(abortion$ABRAPE, levels=c("YES","NO") )
> summary(abortion)
     SEX        ABANY      ABDEFECT     ABHLTH     ABNOMORE     ABPOOR
 FEMALE:2507   YES : 784   YES :1425   YES :1692   YES : 818   YES : 822
 MALE  :2003   NO  :1155   NO  : 495   NO  : 233   NO  :1107   NO  :1106
               NA's:2571   NA's:2590   NA's:2585   NA's:2585   NA's:2582



  ABRAPE         WTSSNR
 YES :1483   Min.   :0.36
 NO  : 429   1st Qu.:0.50
 NA's:2598   Median :0.86
             Mean   :1.00
```

```
             3rd Qu.:1.10
             Max.   :6.59

> # fit a two-class model
> set.seed(234)
> fit2 <- lca( cbind(ABANY,ABDEFECT,ABHLTH,ABNOMORE,ABPOOR,ABRAPE) ~ 1,
+    data=abortion, nclass=2, flatten.gammas=1, flatten.rhos=1)
> summary(fit2)

        Summary of Latent-Class Analysis

=====================================================
Fit statistics
=====================================================

The EM algorithm CONVERGED in: 30 iterations

Standard errors computed successfully.
Standard-error method:   STANDARD

Number of free parameters estimated:      13.000000
Loglikelihood:                         -4633.761747
Loglikelihood + penalty:               -4654.192414
-2 * Loglikelihood:                     9267.523493
AIC (smaller is better):                9293.523493
BIC (smaller is better):                9376.906175

=====================================================
Parameter estimates
=====================================================

Class prevalences (gammas):
Class:             1       2
             0.4297  0.5703

Item-response probabilities (rhos):
   Response category 1
Class:             1       2
   ABANY     0.8843  0.0443
   ABDEFECT  0.9955  0.5438
   ABHLTH    0.9990  0.7840
   ABNOMORE  0.9400  0.0401
   ABPOOR    0.9222  0.0528
   ABRAPE    0.9952  0.6012

   Response category 2
Class:             1       2
   ABANY     0.1157  0.9557
   ABDEFECT  0.0045  0.4562
   ABHLTH    0.0010  0.2160
   ABNOMORE  0.0600  0.9599
   ABPOOR    0.0778  0.9472
   ABRAPE    0.0048  0.3988
```

Class 1, which comprises about 43% of the population, consists of individuals who are support legalized abortion in nearly all circumstances. Class 2, which comprises the remaining 57%, is generally opposed to legalized abortion "for any reason" (ABANY), if the woman doesn't want to have more children (ABNOMORE), or for economic reasons (ABPOOR).

A richer description emerges when we fit a three-class model:

```
> # fit a three-class model
> set.seed(654)
> fit3 <- lca( cbind(ABANY,ABDEFECT,ABHLTH,ABNOMORE,ABPOOR,ABRAPE) ~ 1,
+    data=abortion, nclass=3, flatten.gammas=1, flatten.rhos=1)
> summary(fit3)

        Summary of Latent-Class Analysis

====================================================
Fit statistics
====================================================

The EM algorithm CONVERGED in: 61 iterations

Standard errors computed successfully.
Standard-error method:   STANDARD

Number of free parameters estimated:      20.000000
Loglikelihood:                         -4254.929675
Loglikelihood + penalty:               -4287.879532
-2 * Loglikelihood:                     8509.859350
AIC (smaller is better):                8549.859350
BIC (smaller is better):                8678.140399

====================================================
Parameter estimates
====================================================

Class prevalences (gammas):
Class:             1       2       3
             0.3937  0.4076  0.1986

Item-response probabilities (rhos):
   Response category 1
Class:             1       2       3
   ABANY     0.0710  0.9145  0.0201
   ABDEFECT  0.8217  0.9961  0.0542
   ABHLTH    0.9837  0.9987  0.4002
   ABNOMORE  0.0729  0.9661  0.0134
   ABPOOR    0.1073  0.9397  0.0020
   ABRAPE    0.8659  0.9960  0.1193

   Response category 2
Class:             1       2       3
   ABANY     0.9290  0.0855  0.9799
   ABDEFECT  0.1783  0.0039  0.9458
```

```
ABHLTH    0.0163  0.0013  0.5998
ABNOMORE  0.9271  0.0339  0.9866
ABPOOR    0.8927  0.0603  0.9980
ABRAPE    0.1341  0.0040  0.8807
```

The ρ-parameters for the new Class 2 (41%) are similar to those from the old Class 1; they describe people who favor legalized abortion under any circumstances. But the old Class 2 has now split into the new Class 3 (20%), a group which tends to oppose legalized abortion except when the health of the woman is endangered (ABHLTH), and the new Class 1 (39%), a group which tends to support legalized abortion in cases of moral and ethical dilemma (ABDEFECT, ABHLTH, ABRAPE) but tends oppose it for social and economic reasons (ABANY, ABNOMORE, ABPOOR). Comparing the fit statistics for these two models, we see that introducing 7 additional free parameters has decreased the deviance (2 $\times$ loglikelihood) by $9267.5 - 8509.9 = 757.6$. Latent-class models with different numbers of classes should not be formally compared by a standard likelihood-ratio test, because such comparisons violate the conditions that are necessary for the chisquare approximation to apply. Nevertheless, the loglikelihood does show a dramatic improvement, as do the penalized likelihood fit criteria AIC and BIC. Analyses of GSS data from previous years by McCutcheon (1987) and McCutcheon and Nawojczyk (1995) have led to similar conclusions that a three-class model seems preferable to two.

4.3 A four-class solution

Moving on to a four-class solution, we find that the improvement in fit is less dramatic:

```
> # fit a four-class model
> set.seed(99)
> fit4 <- lca( cbind(ABANY,ABDEFECT,ABHLTH,ABNOMORE,ABPOOR,ABRAPE) ~ 1,
+    data=abortion, nclass=4, flatten.gammas=1, flatten.rhos=1)
> summary(fit4, show.header=F, show.param=F)
=====================================================
Fit statistics
=====================================================

The EM algorithm CONVERGED in: 980 iterations

Standard errors computed successfully.
Standard-error method:   STANDARD

Number of free parameters estimated:      27.000000
Loglikelihood:                         -4248.568686
Loglikelihood + penalty:               -4290.884736
-2 * Loglikelihood:                     8497.137373
AIC (smaller is better):                8551.137373
BIC (smaller is better):                8724.316789
```

Introducing 7 additional parameters decreased the deviance by only 12.7. Moreover, the AIC and BIC statistics both increased, suggesting that the three-class model is preferable.

Another troubling aspect of this four-class model is the presence of a minor mode. If we run the EM algorithm repeatedly from different random starting values, we see that approximately one-third of the solutions have a loglikelihood value that is slightly lower:

```
> # run EM 100 times from different random starts
> loglik4 <- numeric(100)
> set.seed(432)
> for( i in 1:100 ){
+    set.seed(i)
+    loglik4[i] <- lca(
+       cbind(ABANY,ABDEFECT,ABHLTH,ABNOMORE,ABPOOR,ABRAPE) ~ 1,
+       data=abortion, nclass=4, flatten.gammas=1,
+       flatten.rhos=1)$loglik.final
+    }

> # tabulate the loglikelihood values, rounded off to nearest 0.1
> table( round(loglik4, 1) )

-4249.7 -4248.6
     32      68
```

At the major mode, the prevalence of the smallest class is about 6.6%:

```
> # major mode
> set.seed(99)
> fit <- lca(
+       cbind(ABANY,ABDEFECT,ABHLTH,ABNOMORE,ABPOOR,ABRAPE) ~ 1,
+       data=abortion, nclass=4, flatten.gammas=1,
+       flatten.rhos=1)

> round( fit$loglik.final, 1 )
[1] -4248.6

> summary( fit, show.header=F, show.param=F, show.fit=F, show.se.gamma=T)
Standard errors for class prevalences (gammas):
           Est. Std.Err
Class 1 0.06576 0.02552
Class 2 0.34757 0.02701
Class 3 0.39114 0.01487
Class 4 0.19553 0.01125
```

But at the minor mode, the smallest class has a prevalence of about 1.7%:

```
> # minor mode
> set.seed(103)
> fit <- lca(
+       cbind(ABANY,ABDEFECT,ABHLTH,ABNOMORE,ABPOOR,ABRAPE) ~ 1,
+       data=abortion, nclass=4, flatten.gammas=1,
+       flatten.rhos=1)
```

```
> round( fit$loglik.final, 1 )
[1] -4249.7

> summary( fit, show.header=F, show.param=F, show.fit=F, show.se.gamma=T)
Standard errors for class prevalences (gammas):
           Est. Std.Err
Class 1 0.19847 0.01127
Class 2 0.40789 0.01167
Class 3 0.37600 0.02069
Class 4 0.01765 0.01730
```

Minor modes and rare classes are symptomatic of latent-class models that are too complex. Another sign that a model may have too many classes is when the ρ-parameters are difficult to interpret and fail to tell a compelling story about how the classes differ. In this major-mode solution, Classes 1 and 2 both exhibit mixed attitudes toward legalized abortion; the largest difference is with respect to ABPOOR, and whether that difference is important enough to warrant another class is not clear to us.

```
> summary(fit,show.header=F,show.fit=F)
====================================================
Parameter estimates
====================================================

Class prevalences (gammas):
Class:             1       2       3       4
             0.3476  0.0658  0.3911  0.1955

Item-response probabilities (rhos):
   Response category 1
Class:             1       2       3       4
   ABANY     0.0592  0.2760  0.9258  0.0204
   ABDEFECT  0.8064  0.9090  0.9968  0.0522
   ABHLTH    0.9836  0.9797  0.9992  0.3919
   ABNOMORE  0.0677  0.1993  0.9867  0.0132
   ABPOOR    0.0198  0.7753  0.9397  0.0024
   ABRAPE    0.8559  0.9195  0.9971  0.1150

   Response category 2
Class:             1       2       3       4
   ABANY     0.9408  0.7240  0.0742  0.9796
   ABDEFECT  0.1936  0.0910  0.0032  0.9478
   ABHLTH    0.0164  0.0203  0.0008  0.6081
   ABNOMORE  0.9323  0.8007  0.0133  0.9868
   ABPOOR    0.9802  0.2247  0.0603  0.9976
   ABRAPE    0.1441  0.0805  0.0029  0.8850
```

4.4 Analyses with multiple groups

The `lca` function has an optional argument `groups` which allows you to fit models in which the γ's or ρ's vary across levels of a categorical variable. Using this feature, you can test

whether the measurement is invariant across groups. In the abortion attitudes dataset, for example, does the same three-class structure apply to both men and women? To see if it does, we first fit a three-class model in which the ρ's are constrained to be equal for men and women. Next, we fit a model in which the ρ's are allowed to vary. The fit of the two models can be compared by a standard likelihood-ratio test, which has been implemented in a function called `compare.fit`.

```
> # three-class model grouped by SEX with rho's constrained to be equal
> set.seed(588)
> fit3a <- lca( cbind(ABANY,ABDEFECT,ABHLTH,ABNOMORE,ABPOOR,ABRAPE) ~ 1,
+    data=abortion, nclass=3, groups=SEX, constrain.rhos=T,
+    flatten.gammas=1, flatten.rhos=1)
>
> # three-class model grouped by SEX with rho's allowed to vary
> fit3b <- lca( cbind(ABANY,ABDEFECT,ABHLTH,ABNOMORE,ABPOOR,ABRAPE) ~ 1,
+    data=abortion, nclass=3, groups=SEX, constrain.rhos=F,
+    flatten.gammas=1, flatten.rhos=1)

> compare.fit( fit3a, fit3b )
$Chi.Sq
[1] 15.56379

$df
[1] 18

$p
[1] 0.6229616
```

The data show little evidence that the ρ-parameters differ by sex, leading us to conclude that it is reasonable to describe men and women by the same three-class model. Indeed, if we examine the estimates from the unconstrained model, we see that the ρ parameters for men and women look very similar:

```
> summary( fit3b, show.header=F, show.fit=F )
=====================================================
Parameter estimates
=====================================================

Class prevalences (gammas):
Class:            1       2       3
   FEMALE    0.2085  0.4025  0.3889
   MALE      0.1898  0.3771  0.4332

Item-response probabilities (rhos) by group:

   FEMALE
   Response category 1
Class:            1       2       3
   ABANY     0.0057  0.0810  0.9172
   ABDEFECT  0.0404  0.8256  0.9954
```

```
   ABHLTH    0.3825  0.9865  0.9988
   ABNOMORE  0.0046  0.0608  0.9680
   ABPOOR    0.0022  0.1042  0.9445
   ABRAPE    0.1127  0.8562  0.9982

   FEMALE
   Response category 2
Class:            1       2       3
   ABANY     0.9943  0.9190  0.0828
   ABDEFECT  0.9596  0.1744  0.0046
   ABHLTH    0.6175  0.0135  0.0012
   ABNOMORE  0.9954  0.9392  0.0320
   ABPOOR    0.9978  0.8958  0.0555
   ABRAPE    0.8873  0.1438  0.0018

   MALE
   Response category 1
Class:            1       2       3
   ABANY     0.0449  0.0575  0.9095
   ABDEFECT  0.0916  0.8166  0.9954
   ABHLTH    0.4274  0.9850  0.9965
   ABNOMORE  0.0294  0.0900  0.9624
   ABPOOR    0.0087  0.1113  0.9317
   ABRAPE    0.1480  0.8781  0.9918

   MALE
   Response category 2
Class:            1       2       3
   ABANY     0.9551  0.9425  0.0905
   ABDEFECT  0.9084  0.1834  0.0046
   ABHLTH    0.5726  0.0150  0.0035
   ABNOMORE  0.9706  0.9100  0.0376
   ABPOOR    0.9913  0.8887  0.0683
   ABRAPE    0.8520  0.1219  0.0082
```

Do the class prevalences vary by sex? To find out, we apply a similar technique to the γ's:

```
> summary( fit3b, show.header=F, show.fit=F )
> # constrain rho's and gamma's to be equal
> set.seed(32)
> fit3c <- lca( cbind(ABANY,ABDEFECT,ABHLTH,ABNOMORE,ABPOOR,ABRAPE) ~ 1,
+    data=abortion, nclass=3, groups=SEX, constrain.rhos=T, constrain.gammas=T,
+    flatten.gammas=1, flatten.rhos=1)

> # constrain rho's but allow gamma's to vary
> fit3d <- lca( cbind(ABANY,ABDEFECT,ABHLTH,ABNOMORE,ABPOOR,ABRAPE) ~ 1,
+    data=abortion, nclass=3, groups=SEX, constrain.rhos=T, constrain.gammas=F,
+    flatten.gammas=1, flatten.rhos=1)

> compare.fit( fit3c, fit3d )
$Chi.Sq
[1] 4.605563
```

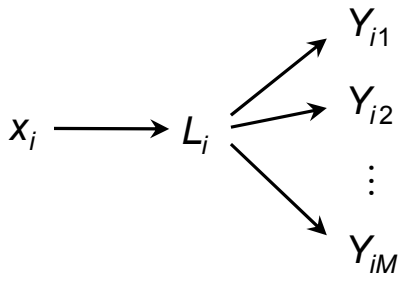

Figure 2: *Latent-class analysis with covariates.*

```
$df
[1] 2

$p
[1] 0.09998036
```

The evidence for differing γ's is stronger but not conclusive.

Another way to see whether class prevalences vary in relation to covariates is to incorporate those covariates into the model as regressors and apply the function `lcacov`, as we now describe.

5 Latent class analysis with covariates

5.1 The model

Let $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$ denote a $p \times 1$ vector of covariates associated with individual i which may influence the probability of belonging to class $L_i = c$ for $c = 1, \ldots, C$. These covariates will not be modeled; they will be treated as fixed constants, analogous to predictors in a regression model. We will suppose that $\boldsymbol{x}_i$ influences the manifest items $\boldsymbol{Y}_i = (Y_{i1}, Y_{i2}, \ldots, Y_{iM})^T$ only through the latent classifier L_i, as shown in Figure 2.

Following common practice, we will relate these covariates to class-membership probabilities through a baseline-category logistic regression model (Agresti, 2002). The probability that an individual i within group g belongs to class c is

$$\gamma_{ic|g} = \Pr(L_i = c \mid g_i = g) = \frac{\exp(\boldsymbol{x}_i{}^T \boldsymbol{\alpha}_{c|g})}{\sum_{c'=1}^{C} \exp(\boldsymbol{x}_i{}^T \boldsymbol{\alpha}_{c'|g})},$$

where $\boldsymbol{\alpha}_{c|g} = (\alpha_{1c|g}, \alpha_{2c|g}, \ldots, \alpha_{pc|g})^T$ is a $p \times 1$ vector of coefficients to be estimated. To identify the parameters, we must choose one of the classes, say class d, to serve as

the baseline or reference class, and we set $\boldsymbol{\alpha}_{d|g} = (0, 0, \ldots, 0)^T$ for that class. Under that restriction, we have

$$\left(\frac{\gamma_{ic|g}}{\gamma_{id|g}}\right) = \boldsymbol{x}_i{}^T \boldsymbol{\alpha}_{c|g},$$

and the elements of $\boldsymbol{\alpha}_{c|g}$ become log-odds ratios for distinguishing class c from class d. Latent-class models that incorporate covariates in this manner were described by Dayton and Macready (1988) and by Bandeen-Roche et al. (1997). In most cases, the first element of $\boldsymbol{x}_i$ will be a constant, $x_{i1} \equiv 1$. When $p = 1$ and $\boldsymbol{x}_i = 1$, this model becomes equivalent to the latent-class model without covariates, where $\gamma_{c|g} = \exp(\boldsymbol{\alpha}_{c|g}) / \sum_{c'=1}^{C} \exp(\boldsymbol{\alpha}_{c'|g})$.

5.2 Estimation procedure

The contribution of case i in group g to the observed-data loglikelihood is now

$$l_i(\boldsymbol{\theta}) = f_i \log \left(\sum_{c=1}^{C} \left\{ \frac{\exp(\boldsymbol{x}_i{}^T \boldsymbol{\alpha}_{c|g})}{\sum_{c'=1}^{C} \exp(\boldsymbol{x}_i{}^T \boldsymbol{\alpha}_{c'|g})} \right\} \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|cg}^{I(y_{im}=r)} \right),$$

and the observed-data likelihood is now

$$l(\boldsymbol{\theta}) = \sum_{g=1}^{G} \sum_{i \in g} f_i \log \left(\sum_{c=1}^{C} \left\{ \frac{\exp(\boldsymbol{x}_i{}^T \boldsymbol{\alpha}_{c|g})}{\sum_{c'=1}^{C} \exp(\boldsymbol{x}_i{}^T \boldsymbol{\alpha}_{c'|g})} \right\} \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|cg}^{I(y_{im}=r)} \right).$$

Once again, we maximize this function by an EM algorithm in which the latent variables L_i play the role of missing data. If the L_i's were known, the loglikelihood would become

$$\begin{aligned} l^*(\boldsymbol{\theta}) &= \sum_{g=1}^{G} \sum_{i \in g} \sum_{c=1}^{C} f_i I(L_i = c) \log \gamma_{ic|g} \\ &\quad \times \sum_{g=1}^{G} \sum_{i \in g} \sum_{c=1}^{C} \sum_{m \in obs_i} \sum_{r=1}^{r_m} f_i I(L_i = c)\, I(y_{im} = r) \log \rho_{mr|cg}. \end{aligned}$$

The E-step is very similar to the one for LCA without covariates. We compute posterior probabilities as before,

$$\begin{aligned} \eta_{ic} &= \Pr(L_i = c \,|\, \boldsymbol{Y}_{i,obs} = \boldsymbol{y}_{i,obs}, g_i = g) \\ &= \frac{\gamma_{ic|g} \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|cg}^{I(y_{im}=r)}}{\sum_{c'=1}^{C} \gamma_{ic'|g} \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|c'g}^{I(y_{im}=r)}}, \end{aligned} \tag{7}$$

but $\gamma_{ic|g}$ is now a function of the covariates $\boldsymbol{x}_i$ and the logistic coefficients $\boldsymbol{\alpha}_{1|g}, \ldots, \boldsymbol{\alpha}_{C|g}$. The M-step for the ρ-parameters is unchanged, but the M-step for the γ-parameters is now replaced by an update of the α's that requires iteration.

The portion of the expected value of $l^*(\boldsymbol{\theta})$ that pertains to the α-parameters is

$$\begin{aligned} Q_\alpha(\boldsymbol{\alpha}) &= \sum_{g=1}^{G} \sum_{i \in g} \sum_{c=1}^{C} f_i \eta_{ic} \log \gamma_{ic|g} \\ &= \sum_{g=1}^{G} \sum_{i \in g} \sum_{c=1}^{C} f_i \eta_{ic} \log \left(\frac{\exp(\boldsymbol{x}_i^T \boldsymbol{\alpha}_{c|g})}{\sum_{c'=1}^{C} \exp(\boldsymbol{x}_i^T \boldsymbol{\alpha}_{c'|g})} \right). \end{aligned} \tag{8}$$

For purposes of the M-step, the η_{ic}'s are considered fixed. If we constrain the α's to be identical across groups,

$$\boldsymbol{\alpha}_{c|1} = \boldsymbol{\alpha}_{c|2} = \cdots \boldsymbol{\alpha}_{c|G} = \boldsymbol{\alpha}_c,$$

then the function becomes

$$\begin{aligned} Q_\alpha(\boldsymbol{\alpha}) &= \sum_{i=1}^{n}\sum_{c=1}^{C} f_i \eta_{ic} \log \gamma_{ic|g} \\ &= \sum_{i=1}^{n}\sum_{c=1}^{C} f_i \eta_{ic} \log\left(\frac{\exp(\boldsymbol{x}_i^T \boldsymbol{\alpha}_c)}{\sum_{c'=1}^{C}\exp(\boldsymbol{x}_i^T \boldsymbol{\alpha}_{c'})}\right). \end{aligned} \tag{9}$$

Maximizing (9) is equivalent to fitting a multinomial logistic regression model to the dataset with fractional outcomes $f_i\eta_{ic}$ in the response categories $c = 1, \ldots, C$ for all cases $i = 1, \ldots, n$. Maximizing (8) is equivalent to fitting the multinomial logistic regression model with fractional outcomes separately within each group $g = 1, \ldots, G$.

The maximizer of (9) can be computed by Newton-Raphson as follows. Let $\boldsymbol{\alpha}$ denote the vector of nonredundant, free α-parameters. In the unconstrained case, we have $\dim(\boldsymbol{\alpha}) = Gp(C-1)$, and in the constrained case, we have $\dim(\boldsymbol{\alpha}) = p(C-1)$. Each iteration of Newton-Raphson can be written as

$$\boldsymbol{\alpha}^{(\text{new})} = \boldsymbol{\alpha}^{(\text{old})} + \left[-Q''_\alpha(\boldsymbol{\alpha}^{(\text{old})})\right]^{-1} Q'_\alpha(\boldsymbol{\alpha}^{(\text{old})}),$$

where

$$Q'_\alpha(\boldsymbol{\alpha}) = \frac{\partial}{\partial \boldsymbol{\alpha}} Q_\alpha(\boldsymbol{\alpha})$$

is the vector of first derivatives, and

$$Q''_\alpha(\boldsymbol{\alpha}) = \frac{\partial^2}{\partial \boldsymbol{\alpha}\, \partial \boldsymbol{\alpha}^T} Q_\alpha(\boldsymbol{\alpha})$$

is the Hessian. In the unconstrained case, the first derivatives are

$$\frac{\partial}{\partial \alpha_{jc|g}} Q_\alpha(\boldsymbol{\alpha}) = \sum_{i \in g} f_i \left(\eta_{ic} - \gamma_{ic|g}\right) x_{ij}.$$

The Hessian has a block-diagonal structure in which the cross-derivatives for different groups vanish,

$$\frac{\partial^2}{\partial \alpha_{jc|g}\, \partial \alpha_{j'c'|g'}} Q_\alpha(\boldsymbol{\alpha}) = 0 \quad \text{for } g' \neq g,$$

and the derivatives within a group are

$$\frac{\partial^2}{\partial \alpha_{jc|g}\, \partial \alpha_{j'c'|g}} Q_\alpha(\boldsymbol{\alpha}) = -\sum_{i \in g} f_i\, \gamma_{ic|g}\, \left[\, I(c = c') - \gamma_{ic'} \,\right] x_{ij}\, x_{ij'}.$$

For the constrained case, the first derivatives are

$$\frac{\partial}{\partial \alpha_{jc}} Q_\alpha(\boldsymbol{\alpha}) = \sum_{i=1}^{n} f_i \left(\eta_{ic} - \gamma_{ic}\right) x_{ij},$$

and the second derivatives are

$$\frac{\partial^2}{\partial \alpha_{jc}\, \partial \alpha_{j'c'}} Q_\alpha(\boldsymbol{\alpha}) = -\sum_{i=1}^{n} f_i\, \gamma_{ic}\, \left[\, I(c = c') - \gamma_{ic'} \,\right] x_{ij}\, x_{ij'}.$$

5.3 Stabilizing the logistic coefficients

In sparse-data situations where the number of covariates is large and/or some of the classes are rare, some of the α-parameters may diverge toward $+\infty$ or $-\infty$ during the M-step. In ordinary logistic regression, this condition is known as quasi-separation (Agresti, 2002). To stabilize the coefficients, we apply a penalty function to $Q_\alpha(\boldsymbol{\alpha})$ that can be viewed as a data-dependent prior distribution. Stabilizing prior distributions for binary logistic regression with an observed response were described by Clogg et al. (1991). In effect, the Clogg prior adds a fictitious fractional number of "successes" and "failures" to each case in the data file. Generalizing their method to our situation, the M-step for $\boldsymbol{\alpha}$ will maximize $Q_\alpha(\boldsymbol{\alpha}) + q_\alpha(\boldsymbol{\alpha})$, where the penalty term is

$$q_\alpha(\boldsymbol{\alpha}) \;=\; \sum_{g=1}^{G} \sum_{i \in g} \sum_{c=1}^{C} k_{ic} \log \gamma_{ic|g} \tag{10}$$

for the unconstrained model and

$$q_\alpha(\boldsymbol{\alpha}) \;=\; \sum_{i=1}^{n} \sum_{c=1}^{C} k_{ic} \log \gamma_{ic} \tag{11}$$

for the constrained model, where the k_{ic}'s are stabilizing constants.

In selecting the stabilizing constants, there are three choices to be made. First, we must decide on the total number of fictitious observations to add to the dataset. For an unconstrained model, the total number of fictitious observations added to group g is $n_g^* = \sum_{i \in g} \sum_{c=1}^{C} k_{ic}$. For a constrained model, the total number of fictitious observations added to the entire sample is $n^* = \sum_{i=1}^{n} \sum_{c=1}^{C} k_{ic}$. Clogg et al. (1991) argue that the total number of fictitious observations should be equal to (or, more generally, proportional to) the number of covariates in the model, because the effective prior precision for the logit-probabilities of class membership will then be the same for any model and any design. Adopting that principle, we get $n_g^* = p$ for $g = 1, \ldots, G$ in an unconstrained model and $n^* = p$ in a constrained model.

The second choice that needs to be made is how to allocate the total number of fictitious observations to the cases in the sample. Clogg et al. opt for an equal allocation. In our situation, an equal allocation would have a slight drawback. We allow the user to supply data in an aggregated fashion, where each case i represents a group of individuals with identical covariates and responses and f_i is the number of sampled individuals in that group, or in a disaggregated fashion, where the cases $i = 1, \ldots, n$ represent individuals and all frequencies $f_1 \ldots, f_n$ are taken to be 1. Users should expect to get identical answers whether the data are aggregated or not. By that principle, the number of fictitious observations allocated to each case should be proportional to f_i. For an unconstrained model, proportional allocation gives

$$\sum_{c=1}^{C} k_{ic} \;=\; p \left(\frac{f_i}{\sum_{i \in g} f_i} \right)$$

for each case in group g. For a constrained model, it gives

$$\sum_{c=1}^{C} k_{ic} \;=\; p \left(\frac{f_i}{\sum_{i=1}^{n} f_i} \right).$$

The third choice that needs to be made is how to allocate the total number of fititious observations for case i across the classes $c = 1, \ldots, C$. Clogg et al. (1991) argue that the allocation should be made in proportion to the outcome-class prevalences estimated without covariates (i.e., from an intercept-only model). Because Clogg et al. were working with a manifest outcome, they could estimate the outcome-class prevalences directly from the marginal distribution of the outcome variable. To implement this procedure in our situation, we need to first run the EM algorithm to convergence without covariates. Let $\tilde{\gamma}_{ic}$ denote the estimated class-membership probability from the model without covariates. (If we are assuming equality of α's across groups, then $\tilde{\gamma}_{ic}$ will be equal for every case $i = 1, \ldots, n$ in the dataset; otherwise, it will be the same for every case within the same group g.) The stabilizing constant becomes

$$k_{ic} \;=\; p \left(\frac{f_i}{\sum_{i \in g} f_i} \right) \tilde{\gamma}_{ic}$$

for the unconstrained model, and

$$k_{ic} \;=\; p \left(\frac{f_i}{\sum_{i=1}^{n} f_i} \right) \tilde{\gamma}_{ic}$$

for the constrained model.

When stabilizing constants are included, the M-step for $\boldsymbol{\alpha}$ proceeds with only a slight modification. For an unconstrained model, the objective function to be maximized becomes

$$Q_\alpha(\boldsymbol{\alpha}) \;+\; q_\alpha(\boldsymbol{\alpha}) \;\;=\;\; \sum_{g=1}^{G} \sum_{i \in g} \sum_{c=1}^{C} (f_i \eta_{ic} + k_{ic}) \log \gamma_{ic|g},$$

and the derivatives are now

$$\frac{\partial}{\partial \alpha_{jc|g}} \left[Q_\alpha(\boldsymbol{\alpha}) \;+\; q_\alpha(\boldsymbol{\alpha}) \right] = \sum_{i \in g} (f_i + k_{i+}) \left(\frac{f_i \eta_{ic} + k_{ic}}{f_i + k_{i+}} \;-\; \gamma_{ic|g} \right) x_{ij}$$

and

$$\frac{\partial^2}{\partial \alpha_{jc|g} \, \partial \alpha_{j'c'|g}} \left[Q_\alpha(\boldsymbol{\alpha}) \;+\; q_\alpha(\boldsymbol{\alpha}) \right] = \;-\; \sum_{i \in g} (f_i + k_{i+}) \;\gamma_{ic|g} \left[I(c = c') \;-\; \gamma_{ic'} \right] x_{ij} \, x_{ij'},$$

where $k_{i+} = \sum_{c=1}^{C} k_{ic}$. For a constrained model, the objective function is

$$Q_\alpha(\boldsymbol{\alpha}) \;+\; q_\alpha(\boldsymbol{\alpha}) \;\;=\;\; \sum_{i=1}^{n} \sum_{c=1}^{C} (f_i \eta_{ic} + k_{ic}) \log \gamma_{ic},$$

and the derivatives are

$$\frac{\partial}{\partial \alpha_{jc}} \left[Q_\alpha(\boldsymbol{\alpha}) \;+\; q_\alpha(\boldsymbol{\alpha}) \right] = \sum_{i=1}^{n} (f_i + k_{i+}) \left(\frac{f_i \eta_{ic} + k_{ic}}{f_i + k_{i+}} \;-\; \gamma_{ic} \right) x_{ij}$$

and

$$\frac{\partial^2}{\partial \alpha_{jc} \, \partial \alpha_{j'c'}} \left[Q_\alpha(\boldsymbol{\alpha}) \;+\; q_\alpha(\boldsymbol{\alpha}) \right] = \;-\; \sum_{i=1}^{n} (f_i + k_{i+}) \;\gamma_{ic} \left[I(c = c') \;-\; \gamma_{ic'} \right] x_{ij} \, x_{ij'}.$$

5.4 Estimating the marginal class prevalences

When covariates are present in the model, it is useful to estimate the marginal class prevalences,

$$\gamma_{\cdot c|g} = \Pr(L_i = c \,|\, g_i = g),$$

where the covariates $\boldsymbol{x}_i$ are no longer being conditioned upon. Note that these marginal γ's could vary across the groups $g = 1, \ldots, G$ even if the α's are constrained to be equal, because the distributions of $\boldsymbol{x}_i$ may vary across groups. Estimates and standard errors for the marginal γ's using the method of expected estimating equations (Wang et al., 2008; Kang & Schafer, submitted).

First, consider the scenario where there are no groups. The marginal γ's to be estimated are then

$$\gamma_{\cdot c} = \Pr(L_i = c)$$

for $c = 1, \ldots, C-1$, and the redundant one is $\gamma_{\cdot C} = 1 - \sum_{c=1}^{C-1} \gamma_{\cdot c}$. If the latent classes were known, we could consistently estimate the marginal γ's by

$$\hat{\gamma}_{\cdot c} = \frac{\sum_{i=1}^{n} f_i \, I(L_i = c)}{\sum_{i=1}^{n} f_i}. \tag{12}$$

We can regard the estimates (12) as the solution to a set of $C-1$ estimating equations arising from a multinomial experiment replicated $\sum_{i=1}^{n} f_i$ times. The loglikelihood function from these multinomial trials is

$$\sum_{i=1}^{n} f_i \left[\sum_{c=1}^{C} I(L_i = c) \log \gamma_{\cdot c} \right] = \sum_{i=1}^{n} f_i \left[\sum_{c=1}^{C-1} I(L_i = c) \log \gamma_{\cdot c} + I(L_i = C) \log \left(1 - \sum_{c=1}^{C-1} \gamma_{\cdot c} \right) \right].$$

Differentiating this expression with respect to the nonredundant $\gamma_{\cdot c}$'s gives the estimating functions $\sum_{i=1}^{n} \omega^*_{ic}$ for $c = 1, \ldots, C-1$, where

$$\omega^*_{ic} = f_i \left[I(L_i = c) \gamma_{\cdot c}^{-1} - I(L_i = C) \left(1 - \sum_{c'=1}^{C-1} \gamma_{\cdot c'} \right)^{-1} \right].$$

Because the L_i's are not observed, we replace the estimating functions by their expectations given the observed data,

$$\begin{aligned} \omega_{ic} &= E(\omega^*_{ic} \,|\, \boldsymbol{Y}_{i,obs} = \boldsymbol{y}_{i,obs}, \boldsymbol{x}_i) \\ &= f_i \left[\eta_{ic} \gamma_{\cdot c}^{-1} - \eta_{iC} \left(1 - \sum_{c'=1}^{C-1} \gamma_{\cdot c'} \right)^{-1} \right]. \end{aligned}$$

The estimated marginal γ's become

$$\hat{\gamma}_{\cdot c} = \frac{\sum_{i=1}^{n} f_i \, \hat{\eta}_{ic}}{\sum_{i=1}^{n} f_i}, \tag{13}$$

where the $\hat{\eta}_{ic}$'s are the estimated posterior probabilities from the final E-step of EM.

5.5 Standard errors for marginal class prevalences

Standard errors for these estimates may be computed as follows. Suppose we collect the nonredundant marginal γ's into a vector, $\boldsymbol{\gamma} = (\gamma_{\cdot 1}, \ldots, \gamma_{\cdot C-1})^T$, and append these parameters onto the vector of free model parameters $\boldsymbol{\theta}$ to get

$$\boldsymbol{\theta}^* = (\boldsymbol{\theta}^T, \boldsymbol{\gamma}^T)^T = (\boldsymbol{\rho}^T, \boldsymbol{\alpha}^T, \boldsymbol{\gamma}^T)^T.$$

The estimate $\hat{\boldsymbol{\theta}}^* = (\hat{\boldsymbol{\theta}}^T, \hat{\boldsymbol{\gamma}}^T)^T$ can be regarded as the joint solution to an expanded set of estimating equations in which the score functions for $\boldsymbol{\theta}$ are stacked upon the expected estimating functions for $\boldsymbol{\gamma}$. The stacked estimating equations are

$$\sum_{i=1}^{n} \boldsymbol{\psi}_i^*(\boldsymbol{\theta}^*) + q'(\boldsymbol{\theta}^*) = \mathbf{0},$$

where $\boldsymbol{\psi}_i$ is the score function for $\boldsymbol{\theta}$ as defined in Section 2.3, $\boldsymbol{\psi}_i^* = (\boldsymbol{\psi}_i^T, \boldsymbol{\omega}_i^T)^T$, $\boldsymbol{\omega}_i = (\omega_{i1}, \ldots, \omega_{i,C-1})^T$, and $q'(\boldsymbol{\theta}^*)$ is the first derivative of the total penalty function q defined by flattening and stabilizing procedures applied to the ρ's and the α's. The sandwich variance estimate for $\hat{\boldsymbol{\theta}}^*$ is

$$V(\hat{\boldsymbol{\theta}}^*) \approx \left(-\sum_{i=1}^{n} \hat{\boldsymbol{\psi}}_i^{*\prime} - q''(\hat{\boldsymbol{\theta}}^*)\right)^{-1} \left(\sum_{i=1}^{n} f_i^{-1} \hat{\boldsymbol{\psi}}_i^* \hat{\boldsymbol{\psi}}_i^{*T}\right) \left[\left(-\sum_{i=1}^{n} \hat{\boldsymbol{\psi}}_i^{*\prime} - q''(\hat{\boldsymbol{\theta}}^*)\right)^{-1}\right]^T, \quad (14)$$

where $\hat{\boldsymbol{\psi}}_i^{*\prime}$ denotes the matrix $\boldsymbol{\psi}_i^{*\prime} = \partial \boldsymbol{\psi}_i / \partial \boldsymbol{\theta}^{*T}$ evaluated at $\boldsymbol{\theta}^* = \hat{\boldsymbol{\theta}}^*$. The matrix $q''(\boldsymbol{\theta}^*)$ has the pattern

$$q''(\boldsymbol{\theta}^*) = \left[\begin{array}{c|c|c} \dfrac{\partial^2 q_\rho}{\partial \boldsymbol{\rho}\, \partial \boldsymbol{\rho}^T} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \dfrac{\partial^2 q_\alpha}{\partial \boldsymbol{\alpha}\, \partial \boldsymbol{\alpha}^T} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right].$$

The matrix $\boldsymbol{\psi}_i^{*\prime}$, which is not symmetric, has the pattern

$$\boldsymbol{\psi}_i^{*\prime} = \left[\begin{array}{c} \dfrac{\partial \boldsymbol{\psi}_i}{\partial \boldsymbol{\theta}^{*T}} \\ \hline \dfrac{\partial \boldsymbol{\omega}_i}{\partial \boldsymbol{\theta}^{*T}} \end{array}\right] = \left[\begin{array}{c|c} \dfrac{\partial^2 l_i}{\partial \boldsymbol{\theta}\, \partial \boldsymbol{\theta}^T} & \mathbf{0} \\ \hline \dfrac{\partial \boldsymbol{\omega}_i}{\partial \boldsymbol{\theta}^T} & \dfrac{\partial \boldsymbol{\omega}_i}{\partial \boldsymbol{\gamma}^T} \end{array}\right] = \left[\begin{array}{c|c|c} \dfrac{\partial^2 l_i}{\partial \boldsymbol{\rho}\, \partial \boldsymbol{\rho}^T} & \dfrac{\partial^2 l_i}{\partial \boldsymbol{\rho}\, \partial \boldsymbol{\alpha}^T} & \mathbf{0} \\ \hline \dfrac{\partial^2 l_i}{\partial \boldsymbol{\alpha}\, \partial \boldsymbol{\rho}^T} & \dfrac{\partial^2 l_i}{\partial \boldsymbol{\alpha}\, \partial \boldsymbol{\alpha}^T} & \mathbf{0} \\ \hline \dfrac{\partial \boldsymbol{\omega}_i}{\partial \boldsymbol{\rho}^T} & \dfrac{\partial \boldsymbol{\omega}_i}{\partial \boldsymbol{\alpha}^T} & \dfrac{\partial \boldsymbol{\omega}_i}{\partial \boldsymbol{\gamma}^T} \end{array}\right]. \quad (15)$$

In the sandwich formula (14), the bread of the sandwich can be written as

$$\left(-\sum_{i=1}^{n}\hat{\boldsymbol{\psi}}_i^{*\prime} - q''(\hat{\boldsymbol{\theta}}^*)\right)^{-1} = \left[\begin{array}{c|c} -\sum_{i=1}^{n}\left(\frac{\partial^2 l_i}{\partial\boldsymbol{\theta}\,\partial\boldsymbol{\theta}^T}\right) - q''(\boldsymbol{\theta}) & \mathbf{0} \\ \hline -\sum_{i=1}^{n}\left(\frac{\partial\boldsymbol{\omega}_i}{\partial\boldsymbol{\theta}^T}\right) & -\sum_{i=1}^{n}\left(\frac{\partial\boldsymbol{\omega}_i}{\partial\boldsymbol{\gamma}^T}\right) \end{array}\right]^{-1}$$

$$= \left[\begin{array}{c|c} \left(-\sum_{i=1}^{n}\left(\frac{\partial^2 l_i}{\partial\boldsymbol{\theta}\,\partial\boldsymbol{\theta}^T}\right) - q''(\boldsymbol{\theta})\right)^{-1} & \mathbf{0} \\ \hline \boldsymbol{A} & \boldsymbol{B} \end{array}\right],$$

where

$$\boldsymbol{A} = -\boldsymbol{B}\left(-\sum_{i=1}^{n}\frac{\partial\boldsymbol{\omega}_i}{\partial\boldsymbol{\theta}^T}\right)\left(-\sum_{i=1}^{n}\left(\frac{\partial^2 l_i}{\partial\boldsymbol{\theta}\,\partial\boldsymbol{\theta}^T}\right) - q''(\boldsymbol{\theta})\right)^{-1}$$

and

$$\boldsymbol{B} = \left(-\sum_{i=1}^{n}\frac{\partial\boldsymbol{\omega}_i}{\partial\boldsymbol{\gamma}^T}\right)^{-1}.$$

The meat of the sandwich (14) is

$$\sum_{i=1}^{n} f_i^{-1}\boldsymbol{\psi}_i^*\boldsymbol{\psi}_i^{*T} = \left[\begin{array}{c|c} \sum_{i=1}^{n} f_i^{-1}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^T & \sum_{i=1}^{n} f_i^{-1}\boldsymbol{\psi}_i\boldsymbol{\omega}_i^T \\ \hline \sum_{i=1}^{n} f_i^{-1}\boldsymbol{\omega}_i\boldsymbol{\psi}_i^T & \sum_{i=1}^{n} f_i^{-1}\boldsymbol{\omega}_i\boldsymbol{\omega}_i^T \end{array}\right].$$

Putting these together, we see that the estimated covariance matrix for $\hat{\boldsymbol{\gamma}}$, the block in the lower right-hand corner of (14), is

$$\begin{aligned} V(\hat{\boldsymbol{\gamma}}) \approx\ & \boldsymbol{A}\left(\sum_{i=1}^{n} f_i^{-1}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^T\right)\boldsymbol{A}^T + \boldsymbol{B}\left(\sum_{i=1}^{n} f_i^{-1}\boldsymbol{\omega}_i\boldsymbol{\psi}_i^T\right)\boldsymbol{A}^T \\ & + \boldsymbol{A}\left(\sum_{i=1}^{n} f_i^{-1}\boldsymbol{\psi}_i\boldsymbol{\omega}_i^T\right)\boldsymbol{B}^T + \boldsymbol{B}\left(\sum_{i=1}^{n} f_i^{-1}\boldsymbol{\omega}_i\boldsymbol{\omega}_i^T\right)\boldsymbol{B}^T. \end{aligned} \tag{16}$$

If a grouping variable is present, we need to estimate $\gamma_{\cdot c|g}$ for each class c within each group g. The estimates, which are now given by

$$\hat{\gamma}_{\cdot c|g} = \frac{\sum_{i\in g} f_i\,\hat{\eta}_{ic}}{\sum_{i\in g} f_i},$$

can be regarded as the solution to a set of $G(C-1)$ estimating equations, $\sum_{i=1}^{n}\omega_{ic|g} = 0$ for $c = 1,\ldots,C-1$ and $g = 1,\ldots,G$, where

$$\omega_{ic|g} = I(g_i = g)\,f_i\left[\eta_{ic}\,\gamma_{\cdot c|g}^{-1} - \eta_{iC}\,\gamma_{\cdot C|g}^{-1}\right].$$

Arrange the $\gamma_{\cdot c|g}$'s into a vector $\boldsymbol{\gamma}$, and define $\boldsymbol{\theta}^* = (\boldsymbol{\theta}^T, \boldsymbol{\gamma}^T)^T$. Similarly, for each i, arrange the $\omega_{ic|g}$'s into vector of estimating functions $\boldsymbol{\omega}_i$, and let $\boldsymbol{\psi}_i^* = (\boldsymbol{\psi}_i^T, \boldsymbol{\omega}_i^T)^T$. The sandwich variance estimate has the same form (14) as before.

5.6 Example: Abortion attitudes

In Section 4.4, we fit a three-class model to the `abortion` dataset and found mild evidence that the class prevalences varied by sex ($p = 0.10$). To perform that test, we fit two models with `lca` using the option `groups=SEX` — one in which the class prevalences varied by `SEX`, the other in which the prevalences varies by `SEX` — and compared the models by a likelihood-ratio test. Now we will perform a similar analysis that uses `SEX` as a covariate in a logistic model using the function `lcacov`.

```
lcacov( formula, data, freq, groups, nclass = 2, reference = 1,
    constrain.rhos = F, constrain.alphas = F, iseeds = NULL,
    iter.max = 5000, tol = 1e-06, starting.values = NULL, flatten.rhos = 0,
    stabilize.alphas = 0, flatten.gammas = 0, se.method = "STANDARD",
    weights, clusters, strata, subpop)
```

The syntax of `lcacov` is very similar to that of `lcov`. A few notable differences are:

- Covariates are now allowed in the model formula on the right-hand side of `~`.
- Covariates may be numeric variables or factors. A factor variable will automatically be entered into the model as a set of dummy codes or contrast terms defined by the factor's `contrasts` attribute, as in the regression functions `lm` and `glm`. See `help(contrasts)` for details.
- The argument `reference` specifies the reference or baseline class for the logistic model.
- The optional argument `flatten.gammas` in `lca` has been replaced with `stabilize.alphas`.

With random starting values, we will not know in advance how the classes $c = 1, \ldots, C$ will be ordered in the solution, which makes it difficult to select a desired class to serve as the reference. One way to address this is to first fit a model with a given random seed, examine the solution, and then refit the model with the same seed and the desired reference class. For example, if we fit a model to the abortion data as follows,

```
> # retrieve the data
> data(abortion)

> # recode NAP and DK as missing values, and
> # reverse the order of the levels so that 1=YES and 2=NO
> abortion$ABANY <- factor(abortion$ABANY, levels=c("YES","NO") )
> abortion$ABDEFECT <- factor(abortion$ABDEFECT, levels=c("YES","NO") )
```

```
> abortion$ABHLTH <- factor(abortion$ABHLTH, levels=c("YES","NO") )
> abortion$ABNOMORE <- factor(abortion$ABNOMORE, levels=c("YES","NO") )
> abortion$ABPOOR <- factor(abortion$ABPOOR, levels=c("YES","NO") )
> abortion$ABRAPE <- factor(abortion$ABRAPE, levels=c("YES","NO") )

> set.seed(123)
> fit <-  lcacov( cbind(ABANY,ABDEFECT,ABHLTH,ABNOMORE,ABPOOR,ABRAPE) ~ SEX,
+    data=abortion, nclass=3, reference=3, flatten.rhos=1, stabilize.alphas=1)
```

and examine the output from `summary(fit)`, we see the following marginal γ's and ρ's:

```
Class prevalences (marginal gammas):
Class:            1       2       3
            0.3938  0.4084  0.1978

Item-response probabilities (rhos):
   Response category 1
Class:            1       2       3
   ABANY     0.0711  0.9144  0.0200
   ABDEFECT  0.8214  0.9962  0.0538
   ABHLTH    0.9834  0.9986  0.4001
   ABNOMORE  0.0727  0.9661  0.0133
   ABPOOR    0.1072  0.9397  0.0020
   ABRAPE    0.8657  0.9960  0.1187

   Response category 2
Class:            1       2       3
   ABANY     0.9289  0.0856  0.9800
   ABDEFECT  0.1786  0.0038  0.9462
   ABHLTH    0.0166  0.0014  0.5999
   ABNOMORE  0.9273  0.0339  0.9867
   ABPOOR    0.8928  0.0603  0.9980
   ABRAPE    0.1343  0.0040  0.8813
```

In this solution, Class 3 contains those who oppose legalized abortion in most or all circumstances, Class 2 contains those who support it in most or all circumstances, and Class 1 contains those whose attitudes are mixed. To designate the opposed group (the current Class 3) as the reference category for the logistic model, we refit the model using the same random seed and `reference=3`. After the model has been fit, we may reorder the classes using `permute.class`.

```
> set.seed(123)
> fit <-  lcacov( cbind(ABANY,ABDEFECT,ABHLTH,ABNOMORE,ABPOOR,ABRAPE) ~ SEX,
+    data=abortion, nclass=3, reference=3, flatten.rhos=1, stabilize.alphas=1)
> fit <- permute.class( fit, c(3,1,2) )
> summary( fit, show.all=T )

  Summary of Latent-Class Analysis with Covariates

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

```
Data and model information
==================================================

Number of cases:   4510

Total frequency for all cases:   4510

Number of measurement items:             6
Number of categories per item:           2 2 2 2 2 2
Number of latent classes:                3
Reference class for logistic model:      Class 1
Number of predictors
   (including a constant, if present):   2

Starting values for rhos:                randomly generated
Random seed 1:                           288
Random seed 2:                           788
Starting values for alphas:              set to zero

Flattening constant for rhos:            1
Stabilizing constant for alphas:         1

Max. number of EM iterations:            5000
Convergence criterion:                   0.000001

==================================================
Fit statistics
==================================================

The EM algorithm CONVERGED in: 65 iterations

Standard errors computed successfully.
Standard-error method:   STANDARD

Number of free parameters estimated:      22.000000
Loglikelihood:                         -4252.624883
Loglikelihood + penalty:               -4286.552199
-2 * Loglikelihood:                     8505.249766
AIC (smaller is better):                8549.249766
BIC (smaller is better):                8690.358919

==================================================
Parameter estimates
==================================================

Class prevalences (marginal gammas):
Class:             1       2       3
             0.1978  0.3938  0.4084

Item-response probabilities (rhos):
   Response category 1
Class:             1       2       3
   ABANY     0.0200  0.0711  0.9144
   ABDEFECT  0.0538  0.8214  0.9962
```

```
   ABHLTH    0.4001  0.9834  0.9986
   ABNOMORE  0.0133  0.0727  0.9661
   ABPOOR    0.0020  0.1072  0.9397
   ABRAPE    0.1187  0.8657  0.9960

   Response category 2
Class:            1       2       3
   ABANY     0.9800  0.9289  0.0856
   ABDEFECT  0.9462  0.1786  0.0038
   ABHLTH    0.5999  0.0166  0.0014
   ABNOMORE  0.9867  0.9273  0.0339
   ABPOOR    0.9980  0.8928  0.0603
   ABRAPE    0.8813  0.1343  0.0040

Logistic regression coefficients (alphas):

, , Class 1/1

            Estimate Std.Err Z.ratio Signif
(Intercept)        0       0     NaN    NaN
SEXMALE            0       0     NaN    NaN

, , Class 2/1

            Estimate  Std.Err Z.ratio Signif
(Intercept)  0.62775 0.097324    6.45 0.0000
SEXMALE      0.14406 0.141240    1.02 0.3077

, , Class 3/1

            Estimate  Std.Err Z.ratio Signif
(Intercept)  0.60626 0.088632   6.840   0.00
SEXMALE      0.27155 0.132230   2.054   0.04

Odds ratios:

, , Class 1/1

            Estimate Lower.95.Pct Upper.95.Pct
(Intercept)        1            1            1
SEXMALE            1            1            1

, , Class 2/1

            Estimate Lower.95.Pct Upper.95.Pct
(Intercept)   1.8734      1.54810       2.2671
SEXMALE       1.1549      0.87568       1.5233

, , Class 3/1

            Estimate Lower.95.Pct Upper.95.Pct
(Intercept)   1.8336       1.5412       2.1814
SEXMALE       1.3120       1.0124       1.7002
```

```
Standard errors for class prevalences (marginal gammas):
           Est. Std.Err
Class 1 0.19784 0.01117
Class 2 0.39378 0.01323
Class 3 0.40837 0.01156

Standard errors for item-response probabilities (rhos):
                                  Est. Std.Err
Class 1, ABANY   , Response 1 0.02000 0.00825
Class 1, ABANY   , Response 2 0.98000 0.00825
Class 1, ABDEFECT, Response 1 0.05383 0.01743
Class 1, ABDEFECT, Response 2 0.94617 0.01743
Class 1, ABHLTH  , Response 1 0.40014 0.03070
Class 1, ABHLTH  , Response 2 0.59986 0.03070
Class 1, ABNOMORE, Response 1 0.01330 0.00696
Class 1, ABNOMORE, Response 2 0.98670 0.00696
Class 1, ABPOOR  , Response 1 0.00195 0.00267
Class 1, ABPOOR  , Response 2 0.99805 0.00267
Class 1, ABRAPE  , Response 1 0.11873 0.02427
Class 1, ABRAPE  , Response 2 0.88127 0.02427
Class 2, ABANY   , Response 1 0.07106 0.01070
Class 2, ABANY   , Response 2 0.92894 0.01070
Class 2, ABDEFECT, Response 1 0.82144 0.01779
Class 2, ABDEFECT, Response 2 0.17856 0.01779
Class 2, ABHLTH  , Response 1 0.98343 0.00636
Class 2, ABHLTH  , Response 2 0.01657 0.00636
Class 2, ABNOMORE, Response 1 0.07275 0.01115
Class 2, ABNOMORE, Response 2 0.92725 0.01115
Class 2, ABPOOR  , Response 1 0.10716 0.01268
Class 2, ABPOOR  , Response 2 0.89284 0.01268
Class 2, ABRAPE  , Response 1 0.86565 0.01527
Class 2, ABRAPE  , Response 2 0.13435 0.01527
Class 3, ABANY   , Response 1 0.91439 0.01115
Class 3, ABANY   , Response 2 0.08561 0.01115
Class 3, ABDEFECT, Response 1 0.99615 0.00245
Class 3, ABDEFECT, Response 2 0.00385 0.00245
Class 3, ABHLTH  , Response 1 0.99864 0.00150
Class 3, ABHLTH  , Response 2 0.00136 0.00150
Class 3, ABNOMORE, Response 1 0.96614 0.00779
Class 3, ABNOMORE, Response 2 0.03386 0.00779
Class 3, ABPOOR  , Response 1 0.93967 0.00941
Class 3, ABPOOR  , Response 2 0.06033 0.00941
Class 3, ABRAPE  , Response 1 0.99595 0.00265
Class 3, ABRAPE  , Response 2 0.00405 0.00265

Significance tests for removal of predictors
Type III, Wald (based on estimated covariance matrix):

             Chi.Sq DF Signif
(Intercept) 52.1680  2  0.000
SEXMALE      4.5463  2  0.103
```

Notice that the logistic coefficients for the reference class, which is now Class 1, have been set to zero. The covariate SEX is a factor and has been expressed as a dummy indicator SEXMALE, defined as 1 if SEX==MALE and 0 if SEX==FEMALE. At the bottom of the summary output is a table that reports test statistics and p-values for the significance of each covariate. The test for SEXMALE has two degrees of freedom, because with three classes the effect of SEXMALE is expressed with two logistic coefficients. This is a Wald test based on the approximation $(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \sim N(0, V(\hat{\boldsymbol{\theta}}))$. Notice that this p-value of 0.103 is very close to the one we reported in Section 4.4 where we used the lca function to test whether the class prevalences varied by SEX. That was a likelihood-ratio test, and with large samples the results from Wald and likelihood-ratio procedures should be similar. We can also perform a likelihood-ratio test in lcacov by fitting the model with and without the covariate and applying compare.fit.

```
> fit0 <-  lcacov( cbind(ABANY,ABDEFECT,ABHLTH,ABNOMORE,ABPOOR,ABRAPE) ~ 1,
+    data=abortion, nclass=3, reference=3, flatten.rhos=1, stabilize.alphas=1)
> compare.fit(fit, fit0)
$Chi.Sq
[1] 4.607716

$df
[1] 2

$p
[1] 0.0998728
```

6 Accounting for complex survey designs

6.1 Survey weights

Data from surveys with complex sampling designs are usually accompanied by weights. The weight for individual i, which we denote by w_i, may be regarded as the number of population individuals represented by the given sampled individual. If individuals were sampled with unequal probabilities (i.e., if some groups were oversampled), modeling procedures that ignore the weights can lead to biased estimates of parameters for the population of interest. Modeling procedures in this LCCA package allow the user to supply weights. If weights are included, the procedure will compute pseudo-maximum likelihood (PML) estimates (Skinner, 1989; Pfefferman, 1993), which maximize the likelihood for a pseudo-population in which individual i has been "cloned" w_i times. PML estimation is formally equivalent to treating w_i as if it were a frequency f_i in an aggregated dataset. For computing standard errors, however, survey weights should not be treated as frequencies. Standard errors require additional information about the design which is not conveyed through the weights.

6.2 The general class of with-replacement designs

Many complex survey designs can be viewed, at least approximately, as special cases of the following general class. The population is divided into $S \geq 1$ sampling strata indexed by $s = 1, \ldots, S$. Within stratum s, primary clusters $c = 1, \ldots, C_s$ are selected with replacement. Within primary cluster c in stratum s, individuals $i = 1, \ldots, n_{cs}$ are sampled by any method, possibly in multiple stages, so that the total sample size is $n = \sum_{s=1}^{S} \sum_{c=1}^{C_s} n_{cs}$. In SUDAAN, this is known as the "with replacement" (WR) design. This is also the design assumed by the "svyreg" and "svylogit" commands in Stata. Design information is conveyed by three user-supplied variables: the individual's survey weight, the cluster identifier (if $n_{cs} > 1$), and the stratum identifier (if $S > 1$). In most surveys, sampling is done without replacement (WOR) to insure that no cluster or individual is selected twice. When sampling is WOR, standard errors computed under a WR assumption tend to be conservative (Wolter, 2007).

Thus far, we have indexed the sampled individuals by a single subscript $i = 1, \ldots, n$, which ignores stratification and clustering. With WR designs, we will sometimes index the individuals by the combination of three subscripts $i = 1, \ldots, n_{cs}$, $c = 1, \ldots, C_s$ and $s = 1, \ldots, S$. Depending on the context, the survey weight for individual i will be denoted either by w_i or by w_{ics}. The estimated size of the population is $N^* = \sum_{i=1}^{n} w_i$.

6.3 Modeling a subpopulation

Analysts often fit models that describes only a subset of the full population (e.g., females). With a simple random sample, we may simply discard the sampled individuals who are not in this subpopulation and fit a model to the remaining individuals, because those who remain are then a simple random sample of the subpopulation of interest. With a complex survey design, however, discarding the individuals who do not belong to the subpopulation is not always appropriate, because the overall design does not necessarily scale down to the subpopulation. To obtain correct standard errors, we will in general need to retain the full sample of all individuals whether or not they belong to the subpopulation. For each individual, we will define an indicator h_i which is equal to 1 if the individual belongs to the subpopulation and 0 otherwise. The number of sampled individuals in the subpopulation is $\sum_{i=1}^{n} h_i$, and the estimated size of the subpopulation is $\sum_{i=1}^{n} h_i w_i$. These indicators play a key role in PML estimation for the subpopulation; contribution of the sampled individual to the pseudo-loglikelihood function becomes the loglikelihood function for that individual multiplied by $h_i w_i$.

When fitting a model to a subpopulation, we may still want to define groups $g = 1, \ldots, G$ within the subpopulation, e.g. for testing invariance of measurement across groups. If so, we will have a grouping variable g_i in addition to the subpopulation indicator h_i. The number of sampled individuals in group g within the subpopulation is then $\sum_{i=1}^{n} I(g_i = g) h_i$, and the estimated size of group g in the subpopulation is $\sum_{i=1}^{n} I(g_i = g) h_i w_i$.

6.4 Rescaling of weights

Survey weights w_i are typically very large. Analysts sometimes rescale weights so that they add up to the sample size, which amounts to

$$\text{replacing } w_i \text{ by } w_i \times \left(\frac{n}{\sum_{i=1}^{n} w_i} \right),$$

or

$$\text{replacing } w_i \text{ by } w_i \times \left(\frac{\sum_{i=1}^{n} h_i}{\sum_{i=1}^{n} h_i w_i} \right).$$

Rescaling the weights has no effect on PML estimates, nor on the standard errors computed by the linearization (sandwich) method that we describe below. If flattening or stabilizing constants are used, however, then a given amount of prior information would be diluted if the algorithm believed that the effective sample size was N^*. For this reason, modeling functions in this LCCA package internally rescale the weights so that the total sample weight in the subpopulation of interest becomes equal to the actual sample size in the subpopulation. That is, we multiply the user-supplied weight w_i by the scaling factor $\sum_i h_i / \sum_i h_i w_i$. Once again, this rescaling has no effect on the PML estimates or standard errors. It simply ensures that, if flattening constants are used, a rescaling of the weights by the user prior to running the program will not affect the results.

6.5 Standard errors for with-replacement designs

Linearized variance estimates under a WR design are obtained as follows. Define the pseudo-score vector for individual i as

$$\boldsymbol{\psi}_i(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} l_i(\boldsymbol{\theta}),$$

where $l_i(\boldsymbol{\theta})$ is individual i's contribution to the pseudo-loglikelihood function, i.e., the loglikelihood for individual i multiplied by the weight w_i and the subpopulation indicator h_i. Using the triple-subscript notation, let $\boldsymbol{\psi}_{ics}(\boldsymbol{\theta})$ denote the pseudo-score vector for sampled individual i within cluster c within stratum s. If the PML estimate $\hat{\boldsymbol{\theta}}$ is not on a boundary of the parameter space, it will solve

$$\sum_{i=1}^{n} \boldsymbol{\psi}_i(\boldsymbol{\theta}) = \sum_{s=1}^{S} \sum_{c=1}^{C_s} \sum_{i=1}^{n_{cs}} \boldsymbol{\psi}_i(\boldsymbol{\theta}) = \mathbf{0}.$$

The estimated covariance matrix for $\hat{\boldsymbol{\theta}}$ comes from a modified sandwich formula,

$$V(\hat{\boldsymbol{\theta}}) \approx \left(-\sum_{i=1}^{n} \boldsymbol{\psi}_i' \right)^{-1} \left(\sum_{s=1}^{S} \sum_{c=1}^{C_s} \left(\boldsymbol{\psi}_{cs} - \bar{\boldsymbol{\psi}}_s \right) \left(\boldsymbol{\psi}_{cs} - \bar{\boldsymbol{\psi}}_s \right)^T \right) \left[\left(-\sum_{i=1}^{n} \boldsymbol{\psi}_i' \right)^{-1} \right]^T,$$

where $\boldsymbol{\psi}_{cs} = \sum_{i=1}^{n_{cs}} \boldsymbol{\psi}_{ics}$ is the total score within cluster c in stratum s, $\bar{\boldsymbol{\psi}}_s = C_s^{-1} \sum_{c=1}^{C_s} \boldsymbol{\psi}_{cs}$ is the average of the cluster totals within stratum s,

$$\boldsymbol{\psi}_i' = \frac{\partial}{\partial \boldsymbol{\theta}^T} \boldsymbol{\psi}_i(\boldsymbol{\theta})$$

is the contribution of individual i to the matrix of second derivatives, and all functions of the unknown parameters are evaluated at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$. If flattening or stabilizing constants are being used, the inner part of the sandwich does not change; only the outer part (the "bread") changes to

$$\left(-\sum_{i=1}^{n} \boldsymbol{\psi}_i' - q''(\boldsymbol{\theta}) \right)^{-1},$$

where $q''(\boldsymbol{\theta})$ is the Hessian of the penalty function q with respect to $\boldsymbol{\theta}$.

To compute standard errors for the marginal class prevalences in a model with covariates, we follow a procedure analogous to that described in Section 5.4. The bread of the augmented sandwich formula (14) is unchanged, and the meat of the augmented sandwich becomes

$$\sum_{s=1}^{S} \sum_{c=1}^{C_s} \left(\boldsymbol{\psi}^*_{cs} - \bar{\boldsymbol{\psi}}^*_s \right) \left(\boldsymbol{\psi}^*_{cs} - \bar{\boldsymbol{\psi}}^*_s \right)^T,$$

where $\boldsymbol{\psi}^*_{cs} = \sum_{i=1}^{n_{cs}} \boldsymbol{\psi}^*_{ics}$ and $\bar{\boldsymbol{\psi}}^*_s = C_s^{-1} \sum_{c=1}^{C_s} \boldsymbol{\psi}^*_{cs}$.

6.6 Testing nested hypotheses

When building latent-class models, we may want to test hypotheses about multiple parameters. When a grouping variable is present, for example, we may wish to test for invariance of the ρ's across groups. Asparouhov and Muthén (2005) derive a corrected likelihood-ratio test (LRT) statistic for PML estimation with survey data which is similar to the robust chi-square statistics of Satorra and Bentler (1988) and Yuan and Bentler (2000). Let l_0 and l_1 denote the maximized values of the pseudo-loglikelihood function under the null and alternative hypotheses, and let d_0 and d_1 denote the number of free parameters estimated under the two models. The design-corrected LRT statistic is $c \cdot 2(l_1 - l_0)$, where

$$c = \frac{d_1 - d_0}{\mathrm{tr}(G_1) - \mathrm{tr}(G_0)},$$

and

$$G_A = \left(-\sum_{i=1}^{n} \boldsymbol{\psi}_i' \right)^{-1} \left(\sum_{s=1}^{S} \sum_{c=1}^{C_s} \left(\boldsymbol{\psi}_{cs} - \bar{\boldsymbol{\psi}}_s \right) \left(\boldsymbol{\psi}_{cs} - \bar{\boldsymbol{\psi}}_s \right)^T \right)$$

is the sandwich bread multiplied by the sandwich meat under Model $A = 0, 1$. The corrected LRT statistic is distributed as $\chi^2_{d_1 - d_0}$ under the null hypothesis provided that the geometric conditions for the usual LRT hold. For example, the null hypothesis must not be located on a boundary of the parameter space. With or without a design correction, the LRT should not be used to compare models with different numbers of classes, because in that case the conditions necessary for an asymptotic chi-square distribution are violated.

6.7 Survey features of the LCCA package

All of the modeling functions in this package (`lca`, `lcacov`, `lcca`) will compute PML estimates and standard errors for WR survey designs. In each case, information about the design is conveyed through three optional arguments: `weights`, `clusters`, and `strata`.

- The `weights` argument should be a numeric variable containing sampling weights. The scaling of these weights is arbitrary.
- The `clusters` argument should be an integer or factor variable containing sampling cluster identifiers.
- The `strata` argument should be an integer or factor variable containing sampling stratum identifiers. It is understood that clusters are nested within strata, so clusters with the same identifier from different strata are known to be different.
- Weights should not be confused with frequencies as supplied by the argument `freq`, because they have different meanings. A frequency of 10 indicates that ten individuals in the sample exhibited the given pattern of responses, but a survey weight of 10 indicates that one sampled individual is representing ten individuals in the population. The same variable supplied as `freq` or `weights` will lead to identical estimates, but the standard errors may be drastically different. You cannot supply both `freq` and `weights` in the same function call. For this reason, data from a survey with unequal probabilities of selection must be supplied in disaggregated form.
- If `weights` is supplied, then the data are assumed to come from a WR design, and the `se.method` option is automatically set to `"SANDWICH"`. The `weights` argument is required for complex survey data; if `weights=NULL`, then `clusters` and `strata` will be ignored.
- If `clusters` is not supplied, then each sampled individual is assumed to be a cluster.
- If `strata` is not supplied, then one stratum is assumed for the whole population.

Another optional argument, `subpop`, allows you to fit a model to a subpopulation when `weights` is present. The `subpop` argument should be a logical variable whose elements are `TRUE` for members of the subpopulation. If `subpop` is supplied and `weights=NULL`, individuals outside of the subpopulation will be ignored when computing estimates and standard errors, and the procedure will be identical to removing the extraneous individuals from the dataset beforehand.

The result from a call to `lca`, `lcacov`, or `lcca` carries information about the sample design. If you apply `summary` to one of these objects with the option `show.data.model=TRUE` or `show.all=TRUE`, then information about the design will be displayed in the printed output.

Nested models fit to the same data may be compared with the function `compare.fit`. If PML estimation was used to fit the models, then `compare.fit` will apply the design-corrected likelihood-ratio procedure of Asparouhov and Muthén (2005).

6.8 Example: Recent smoking from NHANES

The National Health and Nutrition Examination Survey (NHANES) is conducted periodically by the National Center for Health Statistics to assess the health and nutrional status of the United States population. The dataset `NHsmoking` contains variables derived from 2005–2006 NHANES pertaining to recent reported cigarette use. The help file for this dataset is shown below.

```
NHsmoking                package:lcca                R Documentation

Recent cigarette use from NHANES

Description:

     This dataset was derived from the 2005-2006 National Health and
     Nutrition Examination Survey (NHANES) and pertains to
     self-reported recent smoking behavior for persons 12+ years old.

Usage:

     NHsmoking

Format:

     a data frame with 9,950 rows and 9 variables:

     WTMEC2YR respondent's survey weight

     SDMVPSU sampling pseudo-cluster for variance estimation,
          nested within SDMVSTRA

     SDMVSTRA sampling pseudo-stratum for variance estimation

     RIDAGEYR respondent's age in years

     SMQ680r Used tobacco/nicotine in last 5 days?  (1=Yes, 2=No,
          3=Not applicable because AGEYRS<12)

     SMQ690Ar Used cigarettes in last 5 days?  (1=Yes, 2=No,
          3=Not applicable because SMQ680>r)

     SMQ710r No. of days used cigarettes during last 5 days (1=1,
          2=2, 3=3, 4=4, 5=5, 6=Not applicable because SMQ690Ar>1)

     SMQ720r On days used, no. of cigarettes smoked per day
          (1=1-4, 2=5-9, 3=10-19, 4=20+, 5=Not applicable because
          SMQ690Ar>1)

     SMQ725r When did respondent smoke last cigarette?  (1=today,
          2=yesterday, 3=3-5 days ago, 4==Not applicable because
          SMQ690Ar>1)
```

```
Details:

     An r at the end of the variable name indicates a recode of
     an NHANES item. For example, SMQ690Ar is a recoded version
     of the NHANES item SMQ690A. For exact definitions of NHANES
     items, see the documentation for NHANES 2005-2006.

     Recent smoking items are not applicable for participants with
     RIDAGEYR<12.

     The examination sequence created a nested skip pattern for
     SMQ680r, SMQ690Ar, SMQ710r, SMQ720r, and
     SMQ725r. If the response to the first question
     (SMQ680r) ``Used tobacco/nicotine in last 5 days?'' was
     ``No,'' then the remaining items were skipped. If the response was
     ``Yes,'' the participant was asked whether he or she had used
     cigarettes in the last 5 days (SMQ690Ar). If the response
     was ``No,'' then the remaining items were skipped.

     NHANES used a complex multistage sampling design, and design
     information should be taken into account when computing estimates
     and standard errors. The variable WTMEC2YR is the survey
     weight that is intended for analyses involving recent smoking
     items. The pseudo-cluster and pseudo-stratum identifiers
     (SDMVPSU and SDMVSTRA) are not the actual sampling
     strata and primary sampling units but masked versions that protect
     respondents' confidentiality. For more information, see the
     Analytic Guidelines for NHANES 2005-2006.

Source:

     National Center for Health Statistics, Centers for Disease Control
     and Prevention.

References:

     For example analyses of this dataset using functions in the LCCA
     package, see the manual _LCCA Package for R, Version 1_ in the
     subdirectory doc.
```

When fitting latent-class models by PML, there is no obvious way to compare models with different numbers of classes, because the fit criteria AIC and BIC are not defined. As a preliminary step, we suggest ignoring the weights and design information and fitting exploratory models first by ML, understanding that that the parameter estimates may be biased. Once a model has been selected, the design information can be reintroduced to obtain more accurate estimates and standard errors.

Using the `lca` function, we fit models to the five recent-smoking items with varying numbers of classes. Because the recent-smoking items did not apply to children under 12 years old, we used the `subpop` argument to remove those children from consideration. Note that the first smoking item, `SMQ680r`, is equal to 1 or 2 for each member of the subpopulation and 3 for each non-member. If we include this item in a latent-class model with the option

subpop=(RIDAGEYR>=12), an error results because the lca function believes that the item should have three levels, but no values of 3 are found in the subpopulation. We can avoid this problem by recoding the 3's as NA.

Here is an example of a four-class model fit that uses the survey features.

```
> data(NHsmoking)

> # recode 3=Not Applicable as missing for the subpopulation
> NHsmoking$SMQ680r[ NHsmoking$SMQ680r==3 ] <- NA

> set.seed(53)
> fit4 <- lca( cbind(SMQ680r,SMQ690Ar,SMQ710r,SMQ720r,SMQ725r) ~ 1,
+       data=NHsmoking, nclass=4, flatten.gammas=5, flatten.rhos=1,
+       subpop=(RIDAGEYR>=12),
+       weights=WTMEC2YR, clusters=SDMVPSU, strata=SDMVSTRA)

> summary(fit4)

        Summary of Latent-Class Analysis

=====================================================
Fit statistics
=====================================================

The EM algorithm CONVERGED in: 34 iterations

Standard errors computed successfully.
Standard-error method:   SANDWICH

Number of free parameters estimated:      63.00000000
Pseudo-loglikelihood:                  -7877.39791639
Pseudo-loglikelihood + penalty:        -7987.47468076
Design-effect trace:                      19.60787552

=====================================================
Parameter estimates
=====================================================

Class prevalences (gammas):
Class:            1       2       3       4
             0.7316  0.1719  0.0593  0.0371

Item-response probabilities (rhos):
   Response category 1
Class:            1       2       3       4
   SMQ680r   0.0001  0.9995  0.9987  0.9979
   SMQ690Ar  0.0001  0.9994  0.9983  0.0014
   SMQ710r   0.0000  0.0002  0.2344  0.0007
   SMQ720r   0.0000  0.0512  0.6554  0.0008
   SMQ725r   0.0001  0.9938  0.1096  0.0010

   Response category 2
```

```
Class:            1       2       3       4
   SMQ680r   0.9999  0.0005  0.0013  0.0021
   SMQ690Ar  0.0001  0.0003  0.0009  0.9972
   SMQ710r   0.0000  0.0032  0.1849  0.0007
   SMQ720r   0.0000  0.1540  0.1644  0.0008
   SMQ725r   0.0001  0.0057  0.6812  0.0010

   Response category 3
Class:            1       2       3       4
   SMQ680r       NA      NA      NA      NA
   SMQ690Ar  0.9999  0.0003  0.0009  0.0014
   SMQ710r   0.0000  0.0038  0.1696  0.0007
   SMQ720r   0.0000  0.3070  0.1188  0.0008
   SMQ725r   0.0001  0.0002  0.2085  0.0010

   Response category 4
Class:            1       2       3       4
   SMQ680r       NA      NA      NA      NA
   SMQ690Ar      NA      NA      NA      NA
   SMQ710r   0.0000  0.0004  0.3608  0.0007
   SMQ720r   0.0000  0.4876  0.0609  0.0008
   SMQ725r   0.9998  0.0002  0.0007  0.9969

   Response category 5
Class:            1       2       3       4
   SMQ680r       NA      NA      NA      NA
   SMQ690Ar      NA      NA      NA      NA
   SMQ710r   0.0000  0.9924  0.0499  0.0007
   SMQ720r   0.9998  0.0002  0.0005  0.9967
   SMQ725r       NA      NA      NA      NA

   Response category 6
Class:            1       2       3       4
   SMQ680r       NA      NA      NA      NA
   SMQ690Ar      NA      NA      NA      NA
   SMQ710r   0.9998  0.0002  0.0004  0.9965
   SMQ720r       NA      NA      NA      NA
   SMQ725r       NA      NA      NA      NA
```

Examining the ρ-parameters, we see that this four-class solution has an appealing interpretation. Class 1 (estimated to be 73.2% of the subpopulation) contains those who did not use nicotine during the last five days (`SMQ680r==2`). Class 3 (3.7%) contains those who did use nicotine during the last five days but did not smoke cigarettes (`SMQ680r==1` and `SMQ690Ar==2`). Class 2 (17.2%) and Class 3 (5.9%) contain those who did used cigarettes during the last five days, but Class 2 used them every day (`SMQ710r==5`) and in larger quantities (higher values of `SMQ720r`) than Class 3.

When sample design information is taken into account, standard errors for the estimated parameters usually become larger. Here are the standard errors for the class prevalences based on the WR design:

```
> fit4$se.gamma
```

```
            Group 1
Class 1 0.008886725
Class 2 0.007571642
Class 3 0.002913968
Class 4 0.002093518
```

And here are the standard errors computed without the design information:

```
> set.seed(53)
> fit <- lca( cbind(SMQ680r,SMQ690Ar,SMQ710r,SMQ720r,SMQ725r) ~ 1,
+       data=NHsmoking, nclass=4, flatten.gammas=5, flatten.rhos=1,
+       subpop=(RIDAGEYR>=12))

> fit$se.gamma
            Group 1
Class 1 0.005167176
Class 2 0.004303614
Class 3 0.003061122
Class 4 0.002176946

> fit4$se.gamma/fit$se.gamma
          Group 1
Class 1 1.7198419
Class 2 1.7593685
Class 3 0.9519282
Class 4 0.9616764
```

Using the design information, the standard errors for the two largest classes inflated by about 70%.

7 Latent-class causal analysis

7.1 Notation and assumptions

Rubin (1974) introduced a framework and notation for causal inference in nonrandomized studies in which each individual has an outcome for each treatment that could be received. The framework is often called the Rubin causal model (Holland, 1986) or the potential-outcomes model (Rubin, 2005). In previous applications of this model, the putative cause was assumed to be directly observed. Kang and Schafer (submitted) have extended it to situations where the treatment is a latent class. The model for latent-class causal analysis (LCCA) can be described as follows.

Treatment. For each individual $i = 1, \ldots, n$, we suppose that T_i is a latent polytomous treatment variable taking possible values $c = 1, 2, \ldots, C$.

Items measuring the treatment. The treatment is measured by a set of manifest polytomous items $\boldsymbol{U}_i = (U_{i1}, U_{i2}, \ldots, U_{iM})$, where U_{im} takes possible responses $r = 1, \ldots, r_m$. The

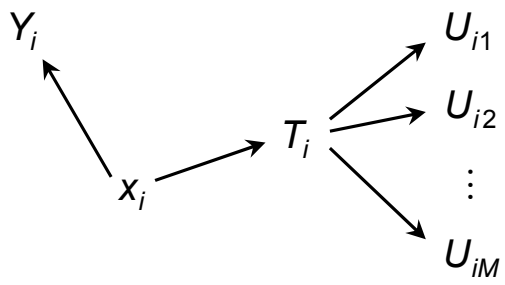

Figure 3: *Latent-class causal analysis.*

realized value of $\boldsymbol{U}_i$ is denoted by $\boldsymbol{u}_i = (u_{i1}, u_{i2}, \ldots, u_{iM})$. We will allow an arbitrary subset of these items to be missing at random, and we partition the items as $\boldsymbol{U}_i = (\boldsymbol{U}_{i,obs}, \boldsymbol{U}_{i,mis})$, where $\boldsymbol{U}_{i,obs}$ is the observed part and $\boldsymbol{U}_{i,mis}$ is the missing part. Similarly, partition $\boldsymbol{u}_i$ as $(\boldsymbol{u}_{i,obs}, \boldsymbol{u}_{i,mis})$.

Potential outcomes. Let $\boldsymbol{Y}_i = (Y_i(1), Y_i(2), \ldots, Y_i(C))^T$ denote a vector of potential outcomes, where $Y_i(c)$ is the outcome that would be realized if $T_i = c$. The observed outcome for individual i is $Y_{i,obs} = Y_i(T_i)$, and its realized value is $y_{i,obs}$. In typical applications, $Y_{i,obs}$ will be measured later than $\boldsymbol{U}_i$, and some individuals may drop out before $Y_{i,obs}$ can be seen. If so, we will suppose that $Y_{i,obs}$ is missing at random for these individuals, and we will still make use of the information in $\boldsymbol{U}_{i,obs}$ to estimate the parameters of the treatment model.

Covariates. Let $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{ip})^T$ denote a column vector of covariates that are completely observed. These covariates will not be explicitly modeled. The realized value of $\boldsymbol{X}_i$ is $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$.

Key assumptions. LCCA makes three key assumptions about the relationships among these variables.

- *Unconfounded treatment assignment:* The treatment T_i is conditionally independent of the potential outcomes $\boldsymbol{Y}_i$ given the covariates $\boldsymbol{x}_i$.
- *Unconfounded measurement:* The items $\boldsymbol{U}_i$ are conditionally independent of the potential outcomes $\boldsymbol{Y}_i$ given the treatment T_i.
- *Local independence.* The items $U_{i1}, \ldots, U_{iM}$ are mutually independent given T_i.

A graphical representation of these relationships is shown in Figure 3. Under these assumptions, the joint distribution of T_i, $\boldsymbol{U}_i$ and $\boldsymbol{Y}_i$ given $\boldsymbol{X}_i$ factors as

$$\Pr(T_i, \boldsymbol{U}_i, \boldsymbol{Y}_i \,|\, \boldsymbol{X}_i) \;=\; \Pr(T_i \,|\, \boldsymbol{X}_i) \left(\prod_{m=1}^{M} \Pr(U_{im} \,|\, T_i) \right) \Pr(\boldsymbol{Y}_i \,|\, \boldsymbol{X}_i). \tag{17}$$

Model parameters. Denote the measurement parameters by

$$\rho_{mr|c} \;=\; \Pr(U_{im} = r \,|\, T_i = c)$$

and the class membership probabilities by

$$\gamma_{ic} = \Pr(T_i = c \mid \boldsymbol{X}_i = \boldsymbol{x}_i).$$

We assume that the γ's follow a baseline-category logistic model,

$$\gamma_{ic} = \frac{\exp\left(\boldsymbol{x}_i{}^T \boldsymbol{\alpha}_c\right)}{\sum_{c'=1}^{C} \exp\left(\boldsymbol{x}_i{}^T \boldsymbol{\alpha}_{c'}\right)}, \tag{18}$$

where $\boldsymbol{\alpha}_c = (\alpha_{1c}, \alpha_{2c}, \ldots, \alpha_{pc})^T$, $c = 1, \ldots, C$ are vectors of coefficients to be estimated (define $\boldsymbol{\alpha}_c = \mathbf{0}$ for the baseline class). Finally, we suppose that

$$\boldsymbol{Y}_i \mid \boldsymbol{X}_i = \boldsymbol{x}_i \sim N(\boldsymbol{\beta}^T \boldsymbol{x}_i, \Sigma), \tag{19}$$

where $\boldsymbol{\beta}$ is a $(p \times C)$ matrix of coefficients to be estimated, and Σ is a $C \times C$ covariance matrix. The cth column of $\boldsymbol{\beta}$ will be noted by $\boldsymbol{\beta}_c = (\beta_{1c}, \beta_{2c}, \ldots, \beta_{pc})^T$. The diagonal elements of Σ are σ_c^2 for $c = 1, \ldots, C$, and the off-diagonal elements are $\sigma_{cc'}$. The correlation coefficients $r_{cc'} = \sigma_{cc'}/\sqrt{\sigma_c^2 \sigma_{c'}^2}$ are strictly inestimable given the observed data and will need to be set to fixed values. However, inferences about average treatment effects are insensitive to assumptions about these correlations.

In equations (18) and (19), we have supposed that the same vector of covariates $\boldsymbol{x}_i$ is used for predicting the treatment status in (18) and the potential outcomes in (19). But these vectors need not be the same. In typical applications, they will be drawn from the same pool of variables $X_{i1}, \ldots, X_{ip}$, but different subsets of covariates may appear in the two parts of the model, perhaps with different transformations or interactions. To allow for this possibility, individual covariates in models (18) and (19) will be denoted by $x_{ij}^{(\alpha)}$ and $x_{ij}^{(\beta)}$, respectively.

7.2 Likelihood function

Collect the free parameters to be estimated into a single parameter vector $\boldsymbol{\theta}$. The elements of $\boldsymbol{\theta}$ consist of

- the nonredundant item-response probabilities, which are $\rho_{mr|c}$ for $r = 1, \ldots, r_m - 1$, $m = 1, \ldots, M$ and $c = 1, \ldots, C$,
- the treatment-model coefficients α_{jc} for each class except the baseline,
- the outcome-model coefficients β_{jc}, and
- the residual variances σ_c^2, $c = 1, \ldots, C$.

The loglikelihood function to be maximized is $l(\boldsymbol{\theta}) = \sum_{i=1}^{n} l_i(\boldsymbol{\theta})$. If individual i remains in the study long enough for $Y_{i,obs} = y_{i,obs}$ to be seen, the individual's contribution to the

loglikelihood is

$$l_i(\boldsymbol{\theta}) = \log \sum_{c=1}^{C} \left\{ \frac{\exp\left(\boldsymbol{x}_i{}^T \boldsymbol{\alpha}_c\right)}{\sum_{c'=1}^{C} \exp\left(\boldsymbol{x}_i{}^T \boldsymbol{\alpha}_{c'}\right)} \right\} \left\{ \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|c}^{I(u_{im}=r)} \right\} \times \left(2\pi\sigma_c^2\right)^{-1/2} \exp\left\{ -\frac{1}{2\sigma_c^2} (y_{i,obs} - \boldsymbol{x}_i{}^T \boldsymbol{\beta}_c)^2 \right\}, \quad (20)$$

where obs_i denotes the subset of $\{1, 2, \ldots, M\}$ corresponding to the items that are observed for individual i. If the individual drops out prior to realization of $Y_{i,obs}$, then the loglikelihood contribution becomes

$$l_i(\boldsymbol{\theta}) = \log \sum_{c=1}^{C} \left\{ \frac{\exp\left(\boldsymbol{x}_i{}^T \boldsymbol{\alpha}_c\right)}{\sum_{c'=1}^{C} \exp\left(\boldsymbol{x}_i{}^T \boldsymbol{\alpha}_{c'}\right)} \right\} \left\{ \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|c}^{I(u_{im}=r)} \right\}. \quad (21)$$

To streamline the notation, we rewrite (20)– (21) as $l_i(\boldsymbol{\theta}) = \log \sum_{c=1}^{C} \gamma_{ic}\, \mathcal{P}_{ic}\, g_{ic}$, where

$$\mathcal{P}_{ic} = \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|c}^{I(u_{im}=r)}$$

and

$$g_{ic} = \exp\left\{ -\frac{1}{2} \log(2\pi) - \frac{1}{2} \log \sigma_c^2 - \frac{1}{2\sigma_c^2} (y_{i,obs} - \boldsymbol{x}_i{}^T \boldsymbol{\beta}_c)^2 \right\}$$

if $Y_{i,obs}$ is seen; if $Y_{i,obs}$ is unseen, define $g_{ic} = 1$. Given the observed data, the posterior probability that subject i belongs to class $T_i = c$ is then

$$\eta_{ic} = \frac{\gamma_{ic}\, \mathcal{P}_{ic}\, g_{ic}}{\sum_{c'=1}^{C} \gamma_{ic'}\, \mathcal{P}_{ic'}\, g_{ic'}}. \quad (22)$$

7.3 Estimation procedure

To estimate $\boldsymbol{\theta}$, we apply an EM algorithm that regards T_i as "missing data." Define the augmented data for individual i as

- $\boldsymbol{X}_i$, $\boldsymbol{U}_{i,obs}$, T_i and $Y_{i,obs}$ if the individual remains in the study long enough for $Y_{i,obs}$ to be seen, and as
- $\boldsymbol{X}_i$, $\boldsymbol{U}_{i,obs}$ and T_i if the individual drops out before $Y_{i,obs}$ can be seen.

Individual's i's contribution to the augmented-data likelihood function is

$$\begin{aligned} L_i^*(\boldsymbol{\theta}) &= \Pr(\boldsymbol{U}_{i,obs} = \boldsymbol{u}_{i,obs}, Y_{i,obs} = y_{i,obs}, T_i \mid \boldsymbol{X}_i) \\ &= \prod_{c=1}^{C} \gamma_{ic}^{I(T_i=c)} \times \prod_{c=1}^{C} \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|c}^{I(T_i=c)\, I(u_{im}=r)} \\ &\quad \times \prod_{c=1}^{C} \exp\left\{ I(T_i = c) \left[-\frac{1}{2} \log(2\pi\sigma_c^2) - \frac{1}{2\sigma_c^2} (y_{i,obs} - \boldsymbol{x}_i{}^T \boldsymbol{\beta}_c)^2 \right] \right\} \end{aligned}$$

if $Y_{i,obs}$ is seen, and

$$\begin{aligned} L_i^*(\boldsymbol{\theta}) &= \Pr(\boldsymbol{U}_{i,obs} = \boldsymbol{u}_{i,obs}, T_i \mid \boldsymbol{X}_i) \\ &= \prod_{c=1}^{C} \gamma_{ic}^{I(T_i=c)} \times \prod_{c=1}^{C} \prod_{m \in obs_i} \prod_{r=1}^{r_m} \rho_{mr|c}^{I(T_i=c)\, I(u_{im}=r)} \end{aligned}$$

if $Y_{i,obs}$ is unseen. The augmented-data loglikelihood is $l^*(\boldsymbol{\theta}) = \sum_{i=1}^{n} l_i^*(\boldsymbol{\theta})$, where $l_i^*(\boldsymbol{\theta}) = \log L_i^*(\boldsymbol{\theta})$. Define $\mathcal{U}_m$ as the subset of $\{1, \ldots, n\}$ corresponding to the individuals for whom U_{im} is seen. Similarly, define $\mathcal{Y}$ as the subset of $\{1, \ldots, n\}$ corresponding to the individuals for whom $Y_{i,obs}$ is seen. The augmented-data loglikelihood can then be written as the sum of three distinct terms,

$$\begin{aligned} l^*(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \sum_{c=1}^{C} I(T_i = c) \log \gamma_{ic} \\ &\quad + \sum_{c=1}^{C} \sum_{m=1}^{M} \sum_{i \in \mathcal{U}_m} \sum_{r=1}^{r_m} I(T_i = c)\, I(u_{im} = r) \log \rho_{mr|c} \\ &\quad + \sum_{c=1}^{C} \sum_{i \in \mathcal{Y}} I(T_i = c) \left\{ -\frac{1}{2} \log(2\pi\sigma_c^2) - \frac{1}{2\sigma_c^2} (y_{i,obs} - \boldsymbol{x}_i{}^T \boldsymbol{\beta}_c)^2 \right\}. \end{aligned} \tag{23}$$

To perform the E-step of the EM algorithm, we replace each missing indicator function $I(T_i = c)$ in $l^*(\boldsymbol{\theta})$ by the posterior probability η_{ic}, where the latter is computed under the current estimated value of $\boldsymbol{\theta}$. The M-step separates into three parts corresponding to the three terms in (23). The second term,

$$\sum_{c=1}^{C} \sum_{m=1}^{M} \sum_{i \in \mathcal{U}_m} \sum_{r=1}^{r_m} \eta_{ic}\, I(u_{im} = r) \log \rho_{mr|c},$$

is maximized at

$$\hat{\rho}_{mr|c} = \frac{\sum_{i \in \mathcal{U}_m} \eta_{ic}\, I(u_{im} = r)}{\sum_{i \in \mathcal{U}_m} \eta_{ic}} \tag{24}$$

for $r = 1, \ldots, r_m$, $m = 1, \ldots, M$ and $c = 1, \ldots, C$. The third term,

$$\sum_{c=1}^{C} \sum_{i \in \mathcal{Y}} \eta_{ic} \left\{ -\frac{1}{2} \log(2\pi\sigma_c^2) - \frac{1}{2\sigma_c^2} (y_{i,obs} - \boldsymbol{x}_i{}^T \boldsymbol{\beta}_c)^2 \right\},$$

is maximized at

$$\hat{\boldsymbol{\beta}}_c = \left(\sum_{i \in \mathcal{Y}} \eta_{ic} \boldsymbol{x}_i \boldsymbol{x}_i{}^T \right)^{-1} \left(\sum_{i \in \mathcal{Y}} \eta_{ic} \boldsymbol{x}_i y_{i,obs} \right), \tag{25}$$

$$\hat{\sigma}_c^2 = \left(\sum_{i \in \mathcal{Y}} \eta_{ic} \right)^{-1} \sum_{i \in \mathcal{Y}} \eta_{ic} (y_{i,obs} - \boldsymbol{x}_i{}^T \hat{\boldsymbol{\beta}}_c)^2 \tag{26}$$

for $c = 1, \ldots, C$. The maximizer of the first term cannot in general be written in closed form, but it may be computed iteratively by a Newton-Raphson procedure. Let $\boldsymbol{\alpha}$ denote the vector containing the coefficients α_{jc} for all classes $c = 1, \ldots, C$ except the baseline class. The function to be maximized is $Q_\alpha(\boldsymbol{\alpha}) = \sum_{i=1}^n \sum_{c=1}^C \eta_{ic} \log \gamma_{ic}$, where the γ_{ic}'s are given by (18) and the η_{ic}'s are regarded as fixed. Each cycle of the Newton-Raphson procedure can be written as

$$\boldsymbol{\alpha}^{(new)} = \boldsymbol{\alpha}^{(old)} + \left[-Q''_\alpha(\boldsymbol{\alpha}^{(old)})\right]^{-1} Q'_\alpha(\boldsymbol{\alpha}^{(old)}),$$

where $Q'_\alpha(\boldsymbol{\alpha})$ is the vector of first derivatives of $Q_\alpha(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$, and $Q''_\alpha(\boldsymbol{\alpha})$ is the matrix of second derivatives. The elements of $Q'_\alpha(\boldsymbol{\alpha})$ are

$$\frac{\partial}{\partial \alpha_{jc}} Q_\alpha = \sum_{i=1}^n (\eta_{ic} - \gamma_{ic})\, x_{ij}^{(\alpha)},$$

and the elements of $Q''_\alpha(\boldsymbol{\alpha})$ are

$$\frac{\partial^2}{\partial \alpha_{jc} \partial \alpha_{j'c'}} Q_\alpha = -\sum_{i=1}^n \gamma_{ic}\, [\, I(c = c') - \gamma_{ic'}\,]\, x_{ij}^{(\alpha)}\, x_{ij'}^{(\alpha)}.$$

The EM algorithm allows us to compute the ML estimate $\hat{\boldsymbol{\theta}}$, but it does not produce standard errors. The covariance matrix for $\hat{\boldsymbol{\theta}}$ can be estimated by

$$\hat{V}(\hat{\boldsymbol{\theta}}) = \left[-\sum_{i=1}^n l''_i(\hat{\boldsymbol{\theta}})\right]^{-1},$$

where $l''_i(\boldsymbol{\theta})$ denotes the matrix of second derivatives of the loglikelihood for case i.

7.4 Estimating average treatment effects

Although the elements of $\boldsymbol{\theta}$ may be interesting, the average treatment effects are contrasts among the means of the potential outcomes *averaged over the covariates*. For causal inference, we need to estimate the vector of marginal means for the entire population, $\boldsymbol{\mu} = E(\boldsymbol{Y}_i)$, or the vector of marginal means within a given treatment class, $\boldsymbol{\mu}_{(d)} = E(\boldsymbol{Y}_i \,|\, T_i = d)$. These parameters are not functions of $\boldsymbol{\theta}$ alone, because our model has not said anything about $\Pr(\boldsymbol{X}_i)$ or $\Pr(\boldsymbol{X}_i \,|\, T_i = c)$. To avoid specifying a model for this possibly high-dimensional set of covariates, we will employ the method of expected estimating functions.

If the potential outcomes $Y_i(c)$ were seen, we could consistently estimate $\mu(c) = E(Y_i(c))$ by $n^{-1} \sum_{i=1}^n Y_i(c)$, which may be regarded as the solution to the estimating equation

$$\sum_{i=1}^n (\, Y_i(c) - \mu(c)\,) = 0. \tag{27}$$

Similarly, if the $Y_i(c)$'s and the T_i's were seen, we could estimate $\mu(c|d) = E(Y_i(c) \,|\, T_i = d)$ by $\sum_{i=1}^n I(T_i = d)\, Y_i(c) / \sum_{i=1}^n I(T_i = d)$, which can be regarded as the solution to the estimating equation

$$\sum_{i=1}^n I(T_i = d)\; (\, Y_i(c) - \mu(c|d)\,) = 0. \tag{28}$$

Because $Y_i(c)$ and T_i are unknown, we replace the estimating functions in these equations by their expected values given the observed data under our latent-class model. That is, we replace $Y_i(c)$ in (27), and $I(T_i = d)$ and $I(T_i = d)Y_i(c)$ in (28), by their expected values given $\boldsymbol{X}_i = \boldsymbol{x}_i$, $\boldsymbol{U}_{i,obs} = \boldsymbol{u}_{i,obs}$, and $Y_{i,obs} = y_{i,obs}$ if the latter is seen. Define

$$\hat{y}_i(c \mid d) \;=\; E(Y_i(c) \mid \boldsymbol{X}_i = \boldsymbol{x}_i, \boldsymbol{U}_{i,obs} = \boldsymbol{u}_{i,obs}, Y_{i,obs} = y_{i,obs}, T_i = d)$$

if $Y_{i,obs}$ is seen for subject i, and

$$\hat{y}_i(c \mid d) \;=\; E(Y_i(c) \mid \boldsymbol{X}_i = \boldsymbol{x}_i, \boldsymbol{U}_{i,obs} = \boldsymbol{u}_{i,obs}, T_i = d)$$

otherwise. It can be shown that

$$\hat{y}_i(c \mid d) \;=\; \begin{cases} y_{i,obs} & \text{if } Y_{i,obs} \text{ is seen and } c = d, \\ \boldsymbol{x}_i{}^T\boldsymbol{\beta}_c \;+\; \left(\dfrac{\sigma_{cd}}{\sigma_d^2}\right)\left(y_{i,obs} \;-\; \boldsymbol{x}_i{}^T\boldsymbol{\beta}_d\right) & \text{if } Y_{i,obs} \text{ is seen and } c \neq d, \text{ and} \\ \boldsymbol{x}_i{}^T\boldsymbol{\beta}_c & \text{if } Y_{i,obs} \text{ is unseen.} \end{cases}$$

The expectations of $Y_i(c)$, $I(T_i = d)$, and $I(T_i = d)Y_i(c)$ given the observed data are then $\sum_{c'=1}^{C} \eta_{ic'}\, \hat{y}_i(c \mid c')$, η_{id} and $\eta_{id}\, \hat{y}_i(c \mid d)$, respectively. Plugging these expressions into (27)–(28) and solving the equations gives

$$\hat{\mu}(c) \;=\; \frac{1}{n}\sum_{i=1}^{n}\sum_{c'=1}^{C} \eta_{ic'}\, \hat{y}_i(c \mid c') \tag{29}$$

and

$$\hat{\mu}(c \mid d) \;=\; \frac{\sum_{i=1}^{n} \eta_{id}\, \hat{y}_i(c \mid d)}{\sum_{i=1}^{n} \eta_{id}}. \tag{30}$$

When computing (29) and (30), we replace the unknown parameters in $\boldsymbol{\theta}$ by their ML estimates.

A covariance matrix for $\hat{\boldsymbol{\mu}} = (\hat{\mu}(1), \ldots, \hat{\mu}(C))^T$ may be estimated as follows. Define $\boldsymbol{\omega}_i = (\omega_i(1), \ldots, \omega_i(C))^T$, where

$$\omega_i(c) \;=\; \sum_{c'=1}^{C} \eta_{ic'}\, \hat{y}_i(c \mid c') \;-\; \mu(c)$$

is the contribution of subject i to the expected estimating function for $\mu(c)$. Define

$$\boldsymbol{S}_i \;=\; l_i'(\boldsymbol{\theta}) \;=\; \frac{\partial}{\partial \boldsymbol{\theta}}\, l_i(\boldsymbol{\theta})$$

as the vector of derivatives of the loglikelihood contribution. The combined estimate $\hat{\boldsymbol{\phi}} = (\hat{\boldsymbol{\theta}}^T, \hat{\boldsymbol{\mu}}^T)^T$ can be regarded as the solution to stacked estimating equations $\sum_{i=1}^{n} \boldsymbol{\psi}_i = \boldsymbol{0}$, where $\boldsymbol{\psi}_i = (\boldsymbol{S}_i^T, \boldsymbol{\omega}_i^T)^T$. Under mild regularity conditions, we have $\sqrt{n}(\hat{\boldsymbol{\phi}} - \boldsymbol{\phi}) \to N(\boldsymbol{0}, \Gamma)$, where $\Gamma = A^{-1}BA^{-1\,T}$, $A = -E(\partial\boldsymbol{\psi}_i/\partial\boldsymbol{\phi}^T)$ and $B = E(\boldsymbol{\psi}_i\boldsymbol{\psi}_i^T)$. An estimated covariance matrix for $\hat{\boldsymbol{\phi}}$ is

$$\hat{V}(\hat{\boldsymbol{\phi}}) \;=\; \left(\sum_{i=1}^{n} \frac{\partial\boldsymbol{\psi}_i}{\partial\boldsymbol{\phi}^T}\right)^{-1} \left(\sum_{i=1}^{n} \boldsymbol{\psi}_i\boldsymbol{\psi}_i^T\right) \left(\sum_{i=1}^{n} \frac{\partial\boldsymbol{\psi}_i}{\partial\boldsymbol{\phi}^T}\right)^{-1\,T}, \tag{31}$$

where all functions on the right-hand side of (31) are evaluated at $\boldsymbol{\phi} = \hat{\boldsymbol{\phi}}$. The matrix $\partial \boldsymbol{\psi}_i / \partial \boldsymbol{\phi}^T$ has the form

$$\frac{\partial \boldsymbol{\psi}_i}{\partial \boldsymbol{\phi}^T} = \left[\begin{array}{c|c} I_i''(\boldsymbol{\theta}) & \mathbf{0} \\ \hline \dfrac{\partial \boldsymbol{\omega}_i}{\partial \boldsymbol{\theta}^T} & \dfrac{\partial \boldsymbol{\omega}_i}{\partial \boldsymbol{\mu}^T} \end{array}\right].$$

An estimated covariance matrix for $\hat{\boldsymbol{\mu}}_{(d)} = (\hat{\mu}(1|d), \ldots, \hat{\mu}(C|d))^T$ may be computed in a similar fashion. Regard $\hat{\boldsymbol{\phi}} = (\hat{\boldsymbol{\theta}}^T, \hat{\boldsymbol{\mu}}_{(d)}^T)^T$ as the solution to stacked estimating equations $\sum_{i=1}^n \boldsymbol{\psi}_i = \mathbf{0}$, where $\boldsymbol{\psi} = ({\boldsymbol{S}_i}^T, \boldsymbol{\omega}_{i(d)}^T)^T$, and $\boldsymbol{\omega}_{i(d)}$ is the vector with elements

$$\omega_i(c \mid d) = \eta_{id} \left[\hat{y}_i(c \mid d) - \mu(c \mid d) \right]$$

for $c = 1, \ldots, C$. The estimated covariance matrix has the same form (31), but $\partial \boldsymbol{\psi}_i / \partial \boldsymbol{\phi}^T$ now becomes

$$\frac{\partial \boldsymbol{\psi}_i}{\partial \boldsymbol{\phi}^T} = \left[\begin{array}{c|c} I_i''(\boldsymbol{\theta}) & \mathbf{0} \\ \hline \dfrac{\partial \boldsymbol{\omega}_{i(d)}}{\partial \boldsymbol{\theta}^T} & \dfrac{\partial \boldsymbol{\omega}_{i(d)}}{\partial \boldsymbol{\mu}_{(d)}^T} \end{array}\right].$$

7.5 Example: A simulated dieting study

Schafer and Kang (2008) presented a simulated observational study to assess the effect of dieting on emotional distress among adolescent girls. Samples were drawn from an artificial population of one million girls. Variables in this population resemble actual variables from the first two waves of the National Longitudinal Study of Adolescent Health (Add Health) (Udry, 2003). However, no actual data from any Add Health participant appears in the population or in the sample; all data were randomly generated from probability distributions estimated from Add Health as described by Schafer and Kang (2008).

Observations for one sample of 6,000 girls from this population are provided in the dataset `diet`. The variables included in this dataset are listed in Table 1. Dieters and nondieters comprise about 20% and 80% of the population, respectively. The treatment variable, a binary indicator of dieting at Wave I, was removed from the dataset and replaced by three conditionally independent binary indicators `U.1`, `U.2`, and `U.3`, with endorsement probabilities of 0.90, 0.85 and 0.80 for dieters and 0.10, 0.15 and 0.20 for nondieters. This structure can be seen by fitting a latent class model with two classes to `U.1`, `U.2`, and `U.3`:

```
> data(diet)
> set.seed(78)
> fit <- lca(cbind(U.1,U.2,U.3) ~ 1, data=diet, nclass=2,
+    flatten.rhos=1, flatten.gammas=1)
> summary(fit, show.header=F, show.fit=F)
==================================================
```

Table 1: *Variables in the simulated dieting dataset.*

Name	Description
DISTRESS.1	Emotional distress score at Wave I
BLACK	1=Black, 0=otherwise
NBHISP	1=non-Black Hispanic, 0=otherwise
GRADE	Grade in school at Wave I (7, ..., 11)
SLFHLTH	Self-rating of overall health (1=excellent, 2=very good, 3=good, 4=fair, 5=poor)
SLFWGHT	Self-rating of weight (1=very underweight, 2=slightly under, 3=about right, 4=slightly over, 5=very over)
WORKHARD	"When you get what you want, it's usually because you worked hard for it" (1=strongly agree, ..., 5=strongly disagree)
GOODQUAL	"You have lots of good qualities" (1=strongly agree, ..., 5=strongly disagree)
PHYSFIT	"You are physically fit" (1=strongly agree, ..., 5=strongly disagree)
PROUD	"You have a lot to be proud of" (1=strongly agree, ..., 5=strongly disagree)
LIKESLF	"You like yourself just the way you are" (1=strongly agree, ..., 5=strongly disagree)
ACCEPTED	"You feel socially accepted" (1=strongly agree, ..., 5=strongly disagree)
FEELLOVD	"You feel loved and wanted" (1=strongly agree, ..., 5=strongly disagree)
U.1	First binary indicator related to dieting
U.2	Second binary indicator related to dieting
U.3	Third binary indicator related to dieting
DISTRESS.2	Distress observed at Wave II

```
Parameter estimates
==================================================

Class prevalences (gammas):
Class:       1       2
         0.786   0.214

Item-response probabilities (rhos):
   Response category 1
Class:       1       2
   U.1  0.9008  0.1371
   U.2  0.8503  0.1541
   U.3  0.8128  0.2086

   Response category 2
Class:       1       2
   U.1  0.0992  0.8629
   U.2  0.1497  0.8459
   U.3  0.1872  0.7914
```

The response variable, DISTRESS.2, is a simulated measure of emotional distress at Wave II. The remaining variables represent confounders recorded at Wave I which influence girls' emotional distress and their propensities to diet. Adjusting for these confounders is essential for estimating the effects of dieting on emotional distress. In particular, it is essential to control for DISTRESS.1, because this measure is strongly related to dieting and to DISTRESS.2

The relationships between dieting and the confounders can be described by regressing the latent-class variable on the confounders:

```
> set.seed(25)
> fit <- lcacov( cbind(U.1,U.2,U.3) ~ DISTRESS.1 + BLACK + NBHISP +
+    GRADE + SLFHLTH + SLFWGHT + WORKHARD + GOODQUAL + PHYSFIT +
+    PROUD + LIKESLF + ACCEPTED + FEELLOVD,
+    data=diet, nclass=2, flatten.rhos=1, stabilize.alphas=1)
> summary(fit, show.header=F, show.fit=F)
==================================================
Parameter estimates
==================================================

Class prevalences (marginal gammas):
Class:       1       2
        0.7935  0.2065

Item-response probabilities (rhos):
   Response category 1
Class:       1       2
   U.1  0.8993  0.1151
   U.2  0.8459  0.1460
   U.3  0.8072  0.2081

   Response category 2
Class:       1       2
```

```
   U.1  0.1007  0.8849
   U.2  0.1541  0.8540
   U.3  0.1928  0.7919

Logistic regression coefficients (alphas):

, , Class 1/1

            Estimate Std.Err Z.ratio Signif
(Intercept)        0       0     NaN    NaN
DISTRESS.1         0       0     NaN    NaN
BLACK              0       0     NaN    NaN
NBHISP             0       0     NaN    NaN
GRADE              0       0     NaN    NaN
SLFHLTH            0       0     NaN    NaN
SLFWGHT            0       0     NaN    NaN
WORKHARD           0       0     NaN    NaN
GOODQUAL           0       0     NaN    NaN
PHYSFIT            0       0     NaN    NaN
PROUD              0       0     NaN    NaN
LIKESLF            0       0     NaN    NaN
ACCEPTED           0       0     NaN    NaN
FEELLOVD           0       0     NaN    NaN

, , Class 2/1

              Estimate  Std.Err Z.ratio Signif
(Intercept) -6.2055000 0.390970 -15.872 0.0000
DISTRESS.1   0.3936000 0.097921   4.020 0.0001
BLACK       -0.5767800 0.109550  -5.265 0.0000
NBHISP      -0.1733000 0.113320  -1.529 0.1262
GRADE        0.1016400 0.029980   3.390 0.0007
SLFHLTH     -0.0789000 0.048764  -1.618 0.1057
SLFWGHT      1.2013000 0.064942  18.498 0.0000
WORKHARD    -0.1698400 0.047935  -3.543 0.0004
GOODQUAL    -0.2116000 0.081894  -2.584 0.0098
PHYSFIT      0.0332490 0.055204   0.602 0.5470
PROUD        0.0022100 0.083225   0.027 0.9788
LIKESLF      0.2662100 0.049560   5.371 0.0000
ACCEPTED    -0.0925140 0.055187  -1.676 0.0937
FEELLOVD     0.0014553 0.068451   0.021 0.9830
```

The most powerful predictor of dieting is `SLFWGHT`, but many other variables, including `DISTRESS.1`, are significantly related to dieting as well.

7.6 Using the lcca function

The syntax of the `lcca` function is shown below.

```
lcca(formula.treatment, formula.outcome, data, nclass = 2,
```

```
    reference = 1, iseeds = NULL, iter.max = 5000, tol = 1e-06,
    starting.values = NULL, flatten.rhos = 0, stabilize.alphas = 0,
    flatten.gammas = 0, se.method = "STANDARD", r.matrix = NULL,
    freq, weights, clusters, strata, subpop)
```

The major difference between this function and `lcacov` is that this function requires two model formulas.

- The first argument, `formula.treatment`, is a formula like the one required for `lcacov`. The expression on the left-hand side of ~ is a matrix of polytomous variables that meassure the latent class, and the right-hand side specifies the covariates used to predict the latent class.
- The second argument, `formula.outcome`, should have the form `Y ~ X1  X2 + ...+`. The variable on the left-hand side of ~ is the numeric outcome variable, and the terms on the right-hand side are covariates predicting the outcome. Missing values in the outcome are allowed and should be conveyed by the R missing value code `NA`.

An application of this function to the dieting data is shown below.

```
> set.seed(25)
> fit <- lcca(
+    formula.treatment = cbind(U.1,U.2,U.3) ~ DISTRESS.1 + BLACK +
+       NBHISP + GRADE + SLFHLTH + SLFWGHT + WORKHARD + GOODQUAL +
+       PHYSFIT + PROUD + LIKESLF + ACCEPTED + FEELLOVD,
+    formula.outcome = DISTRESS.2  ~ DISTRESS.1 + BLACK +
+       NBHISP + GRADE + SLFHLTH + SLFWGHT + WORKHARD + GOODQUAL +
+       PHYSFIT + PROUD + LIKESLF + ACCEPTED + FEELLOVD,
+    data=diet, nclass=2, flatten.rhos=1, stabilize.alphas=1)
> summary(fit)

      Summary of Latent-Class Causal Analysis

==================================================
Fit statistics
==================================================

The EM algorithm CONVERGED in: 73 iterations

Standard errors computed successfully.
Standard-error method:   STANDARD

Number of free parameters estimated:       50.00000
Loglikelihood:                         -12035.26553
Loglikelihood + penalty:               -12050.05682
-2 * Loglikelihood:                     24070.53105
AIC (smaller is better):                24170.53105
BIC (smaller is better):                24505.50679

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

```
Parameter estimates
==================================================

Class prevalences (marginal gammas):
Class:       1       2
        0.7935  0.2065

Item-response probabilities (rhos):
   Response category 1
Class:       1       2
   U.1  0.8990  0.1158
   U.2  0.8459  0.1458
   U.3  0.8072  0.2080

   Response category 2
Class:       1       2
   U.1  0.1010  0.8842
   U.2  0.1541  0.8542
   U.3  0.1928  0.7920

Treatment model coefficients (alphas):

, , Class 1/1

            Estimate Std.Err Z.ratio Signif
(Intercept)        0       0     NaN    NaN
DISTRESS.1         0       0     NaN    NaN
BLACK              0       0     NaN    NaN
NBHISP             0       0     NaN    NaN
GRADE              0       0     NaN    NaN
SLFHLTH            0       0     NaN    NaN
SLFWGHT            0       0     NaN    NaN
WORKHARD           0       0     NaN    NaN
GOODQUAL           0       0     NaN    NaN
PHYSFIT            0       0     NaN    NaN
PROUD              0       0     NaN    NaN
LIKESLF            0       0     NaN    NaN
ACCEPTED           0       0     NaN    NaN
FEELLOVD           0       0     NaN    NaN

, , Class 2/1

               Estimate  Std.Err Z.ratio Signif
(Intercept) -6.24290000 0.392030 -15.925 0.0000
DISTRESS.1   0.41078000 0.098307   4.179 0.0000
BLACK       -0.56539000 0.109510  -5.163 0.0000
NBHISP      -0.17070000 0.113400  -1.505 0.1323
GRADE        0.10105000 0.030006   3.368 0.0008
SLFHLTH     -0.07922600 0.048752  -1.625 0.1041
SLFWGHT      1.20250000 0.065048  18.487 0.0000
WORKHARD    -0.16582000 0.047884  -3.463 0.0005
GOODQUAL    -0.20261000 0.081884  -2.474 0.0133
PHYSFIT      0.03093800 0.055249   0.560 0.5755
PROUD        0.00014289 0.083242   0.002 0.9986
```

```
LIKESLF      0.26926000 0.049580   5.431 0.0000
ACCEPTED    -0.09375700 0.055247  -1.697 0.0897
FEELLOVD     0.00370590 0.068398   0.054 0.9568

Outcome model coefficients (betas):

, , Class 1

              Estimate   Std.Err Z.ratio Signif
(Intercept)  0.00540790 0.0472680   0.114 0.9089
DISTRESS.1   0.52183000 0.0147990  35.261 0.0000
BLACK        0.07083700 0.0133350   5.312 0.0000
NBHISP       0.02526100 0.0162640   1.553 0.1204
GRADE        0.00212620 0.0040005   0.531 0.5951
SLFHLTH      0.02713900 0.0067815   4.002 0.0001
SLFWGHT     -0.00030758 0.0078784  -0.039 0.9689
WORKHARD    -0.01759300 0.0063674  -2.763 0.0057
GOODQUAL     0.01991300 0.0111830   1.781 0.0750
PHYSFIT      0.00336770 0.0078593   0.428 0.6683
PROUD        0.03497500 0.0114770   3.047 0.0023
LIKESLF      0.01863700 0.0074141   2.514 0.0119
ACCEPTED     0.01535400 0.0078433   1.958 0.0503
FEELLOVD     0.04224400 0.0096847   4.362 0.0000

, , Class 2

              Estimate   Std.Err Z.ratio Signif
(Intercept) -0.0480640 0.1235900  -0.389 0.6974
DISTRESS.1   0.5085200 0.0306810  16.575 0.0000
BLACK        0.0922290 0.0358910   2.570 0.0102
NBHISP       0.0438060 0.0357110   1.227 0.2199
GRADE        0.0029893 0.0094719   0.316 0.7523
SLFHLTH     -0.0066383 0.0154600  -0.429 0.6677
SLFWGHT      0.0097890 0.0208250   0.470 0.6383
WORKHARD     0.0121650 0.0158060   0.770 0.4415
GOODQUAL     0.0268440 0.0252990   1.061 0.2887
PHYSFIT     -0.0151720 0.0176360  -0.860 0.3896
PROUD        0.0505120 0.0261750   1.930 0.0536
LIKESLF      0.0410560 0.0153050   2.683 0.0073
ACCEPTED     0.0175930 0.0170350   1.033 0.3017
FEELLOVD     0.0258760 0.0219800   1.177 0.2391

Outcome model residual variances (sigma2):
        Estimate   Std.Err
Class 1  0.13569 0.0030700
Class 2  0.15925 0.0081499

==================================================
Estimated potential-outcome means
==================================================

Average potential outcomes:
        Estimate   Std.Err
Class 1  0.65622 0.0071832
```

```
Class 2  0.65319 0.0219230

Average potential outcomes within classes:
, , Class 1

        Estimate   Std.Err
Class 1  0.64067 0.0074461
Class 2  0.63757 0.0241720

, , Class 2

        Estimate  Std.Err
Class 1  0.71596 0.013062
Class 2  0.71320 0.018385


================================================
Estimated average treatment effects
================================================

Average treatment effects:
                        Estimate  Std.Err Z.ratio Signif
Class 2 minus Class 1 -0.0030304 0.024123  -0.126    0.9

Average treatment effects within classes:
, , Class 1

                       Estimate  Std.Err Z.ratio Signif
Class 2 minus Class 1 -0.003101 0.026654  -0.116 0.9074

, , Class 2

                        Estimate  Std.Err Z.ratio Signif
Class 2 minus Class 1 -0.0027587 0.018968  -0.145 0.8844
```

The estimated average effects of dieting on emotional distress are essentially zero in the overall population and in each treatment class.

References

Agresti, A. (2002) *Categorical Data Analysis* (2nd Ed.). New York: Wiley.

Asparouhov, T. and Muthén, B. (2005) Multivariate statistical modeling with survey data. *Proceedings of the Federal Committee on Statistical Methodology (FCSM) Research Conference*. Washington, DC: Office of Management and Budget.

Bandeen-Roche, K., Miglioretti, D.L., Zeger, S.L., and Rathouz, P.R. (1997) Latent variable regression for multiple discrete outcomes. *Journal of the American Statistical Association*, 92, 1375–1386.

Chung, H. (2003) *Latent-Class Modeling with Covariates*. Unpublished doctoral dissertation. University Park, PA: Department of Statistics, The Pennsylvania State University.

Clogg, C.C. and Goodman, L.A. (1984) Latent structure analysis of a set of multidimensional contingency tables. *Journal of the American Statistical Association*, 79, 762–771.

Clogg, C.C., Rubin, D.B., Schenker, N., Schultz, B., and 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, 68–78.

Collins, L.M. and Lanza, S.T. (2010) *Latent Class and Latent Transition Analysis: With Applications in the Social, Behavioral, and Health Sciences.* New York: Wiley.

Dayton, C.M. and Macready, G.B. (1988) Concomitant-variable latent-class models. *Journal of the American Statistical Association*, 83, 173-178.

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

Holland, P.W. (1986) Statistics and causal inference. *Journal of the American Statistical Association*, 81, 945–970.

Kang, J.D.Y. and Schafer, J.L. (submitted) Estimating average treatment effects when the treatment is a latent class.

Lazarsfeld, P.F. and Henry, N.W. (1968) *Latent Structure Analysis*. Boston: Houghton-Mifflin.

Linzer, D.A. and Lewis, J. (2007) *poLCA: Polytomous Variable Latent Class Analysis.* R package version 1.1. `http://userwww.service.emory.edu/~dlinzer/poLCA`.

Little, R.J.A. and Rubin, D.B. (2002) *Statistical Analysis with Missing Data*, Second Edition. New York: Wiley.

Patterson, B.H., Dayton, C.M. and Graubard, B.I. (2002) Latent class analysis of complex sample survey data: Application to dietary data. *Journal of the American Statistical Association*, 97, 721–729.

Pfeffermann, D. (1993) The role of sampling weights when modeling survey data. *Interna-*

tional Statistical Review, 61, 317–337.

Rubin, D.B. (1974) Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66, 688-701.

Rubin, D.B. (2005) Causal inference using potential outcomes: design, modeling, decisions. *Journal of the American Statistical Association*, 100, 322331.

Satorra, A. and Bentler, P.M. (1988) Scaling corrections for chi-square statistics in covariance structure analysis. *Proceedings of the Business and Economic Statistics Section of the American Statistical Association*, 308–313.

Schafer, J.L. and Kang, J.D.Y. (2008) Average causal effects from observational studies: a practical guide and simulated example. *Psychological Methods*, 13, 279-313.

Skinner, C.J. (1989) Domain means, regression and multivariate analysis. In Analysis of Complex Surveys, C.J. Skinner, D. Holt and T.F.M. Smith (Eds.), pp. 59–87. Chichester: Wiley.

Udry, J.R. (2003) *The National Longitudinal Study of Adolescent Health (Add Health), Waves I and II, 1994-1996; Wave III, 2001-2002* (machine-readable data file and documentation), Chapel Hill, NC: Carolina Population Center, University of North Carolina at Chapel Hill.

Wang, C.Y., Huang, Y., Chao, E.C. and Jeffcoat, M.K. (2008) Expected estimating equations for missing data, measurement error, and misclassification, with application to longitudinal nonignorable missing data. *Biometrics*, 64, 85–95.

Wolter, K.M. (2007) *Introduction to Variance Estimation*, Second Edition. New York: Springer.

Yang, I. and Becker, M.P. (1997) Latent variable modeling of diagnostic accuracy. *Biometrics*, 53, 948–958.

Yuan, K. and Bentler, P.M. (2000) Three likelihood-based methods for mean and covariance structure analysis with nonnormal missing data. *Sociological Methodology* 30, 167–202.