glca: An R Package for Multiple-Group Latent Class Analysis [1]

Youngsun Kim[a], Saebom Jeon[b], Chi Chang[c] and Hwan Chung[a] [2]

[a] Korea University, Seoul, Korea
[b] Mokwon University, Daejeon, Korea
[c] Michigan State University, East Lansing, MI, USA

[1]This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education and Ministry of Science and ICT (2021R1A2C1003486 to Chung and 2020R1F1A1A01055067 to Jeon).

[2]Correspondence should be sent to Hwan Chung, Professor, Department of Statistics, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul 02841, Korea. E-mail: hwanch@korea.ac.kr. Phone: +82-2-3290-2246.

glca: An R Package for Multiple-Group Latent Class Analysis

Latent class analysis (LCA) is one of the most popular discrete mixture models for classifying individuals based on their responses to multiple manifest items. When there are existing subgroups in the data representing different populations, researchers are often interested in comparing certain aspects of latent class structure across these groups in LCA approach. In multiple-group LCA models, individuals are dependent owing to multilevel data structure, where observation units (i.e., individuals) are nested within a higher-level unit (i.e., group). This paper describes the implementation of multiple-group LCA in the R package glca for exploring differences in latent class structure between populations, taking multilevel data structure into account. The package glca deals with the fixed effect LCA and the random effect LCA; the former can be applied in the situation where populations are segmented by the observed group variable itself, whereas the latter can be used when there are too many levels in the group variable to make a meaningful group comparisons. After a brief introduction in these two-types of multiple-group LCA models, we provide the statistical framework, estimation methods, and statistical test procedures implemented in the package glca. A description of the available functions is followed by two practical examples in order to provide R users with utilities for multiple-group LCA.

Keywords: glca, latent class analysis, measurement invariance, multilevel data, R package

Introduction

In this study we introduce the R package glca (Kim & Chung, 2021), which implements multiple-group latent class models for exploring differences between populations in the data with a multilevel structure. LCA is one of the most popular mixture models now being used to divide the population into mutually exclusive subgroups of individuals based on their responses to several manifest items. LCA posits the existence of a latent classifier to explain the association among categorical variables (Goodman, 1974; Clogg & Goodman, 1984). The basic idea of LCA is that population is assumed to consist of heterogeneous subgroups (i.e., latent classes), and the distributions of manifest items vary across latent classes. There are two important assumptions underlying LCA: (1) responses to manifest items are conditionally independent within each latent classifier (i.e., local independence assumption) (Lazarsfeld & Henry, 1968); and (2) individual observations are independent of one another. In multiple-group latent class models, however, the latter is often violated because of the multilevel structure in the data. When individual observations (i.e., level-1 units) are nested within a group (i.e., level-2 unit) in the data, the patterns of responses to manifest items often shows more similarity within a group than between groups. Here, we refer to the variable presenting level-2 unit as *group variable*.

Group similarities and differences may manifest themselves in a variety of ways in LCA. Sometimes measurement models are identical across groups, which means the interpretation

of latent classes is invariant across groups. In multiple-group LCA, measurement model for each group is characterized by probabilities of providing any particular response pattern in a particular latent class. Therefore, *measurement invariance* in multiple-group LCA implies that individuals in a particular latent class but in different groups have identical probabilities of providing any particular response pattern; in other words, when there is measurement invariance, any group differences are in latent class prevalences, not in probabilities of providing any particular response pattern. Here, the probability of providing a specific response to the manifest item in a particular latent class is referred to as *item-response probability*. In other situations the measurement models may differ, suggesting that the latent structure itself is different between groups. Thus, it no longer makes sense simply to compare latent class prevalences directly. Tests of measurement of invariance shed light on this distinction.

There are two approaches to reflect group differences in LCA for the data with a multilevel structure. One approach is to construct a specific latent class model for each group by incorporating group variable directly. This approach, referred to as *fixed-effect LCA*, allows us to compare certain aspects of latent class structure across groups by estimating group-specific model parameters in LCA. For example, latent class regression (LCR) incorporates group variable as a covariate in the form of a logistic regression, and it uses an odds ratio to compare groups in the latent class prevalence (Dayton & Macready, 1988; Bandeen-Roche et al., 1997). In LCR the measurement model is assumed to be identical across groups, and therefore, it is difficult to test measurement invariance. Clogg & Goodman, 1985 proposed simultaneous LCA to examine whether the latent class prevalence and the association between manifest items and the latent class variable differ across groups. This method is more appropriate than LCR for testing measurement invariance, because item-response probabilities are allowed to vary across groups. Both LCR and simultaneous LCA are fixed effect LCA models, which estimate group-specific parameters for class prevalence and/or item-response probability. The fixed-effect LCA is applicable when population membership is always observed directly by group variable and the available sample accordingly is divided into population groups. If there are too many levels in the observed group variable, however, it needs to be reclassified into a small number of categories to explore meaningful differences between subpopulations. The reclassification, which is considered as group-level latent variable, can be inferred by the similarities of the observed groups in their latent class prevalences. To deal with this problem, Vermunt (2003) proposed *random-effect LCA*, where parametric or nonparametric random coefficients are introduced into the LCA model. The core of the parametric random-effect LCA is to control dependency within a group by incorporating random effects, which requires distributional assumption on the random coefficient. Conversely, in nonparametric random-effect LCA random quantities are placed on a group-level latent variable, which is identified by the joint distribution of class prevalences for all groups. The group-level latent variable, referred to as *latent cluster*, serves as group variable, and we can explore differences in latent structure between latent cluster memberships in nonparametric random-effect LCA.

Analysis using latent class models can be conducted by a number of R packages including `poLCA` (Linzer & Lewis, 2011), `e1071` (Meyer et al., 2020), `BayesLCA` (White & Murphy, 2014), and `randomLCA` (Beath, 2017), which are available from the Comprehensive R Archive Network (CRAN)[1]. All these packages can fit the standard LCA model, although they are specialized in diverse aspects. The R package `poLCA` can fit an LCR that allows multiple categories in manifest items. The R package `BayesLCA` implements Bayesian LCA using methods such as Gibbs sampling and variational Bayes. Parametric random-effect LCA can be fitted by the R package `randomLCA` where random coefficients are incorporated into the LCA model to ex-

[1]URL: `http://cran.r-project.org/`

plain heterogeneity in item responses within a class. In other words, random effects are placed on item-response probabilities given the class membership in the form of a logistic regression. Both R packages `BayesLCA` and `randomLCA` can only fit LCA models with binary manifest items. The R package `e1071` provides the function for testing goodness-of-fit of a specified LCA using bootstrap. Commercial softwares such as LatentGOLD and Mplus can be used to define various LC models and fit the data, but to our knowledge, software that focuses on group comparisons within the LC structure, whether the group is observed and/or latent, is not available in R.

In this respect, we have developed the R package `glca` that fits both fixed-effect and random-effect latent class models. The main function of the package implements expectation-maximization algorithm (Dempster et al., 1977), where Newton-Raphson method is employed in maximization step in order to find maximum-likelihood estimates. Moreover, missing values in manifest items can be treated under ignorable missing data condition in the package `glca`. The goodness-of-fit test for measurement invariance across groups is available to the user of the package. In addition, the package quantifies uncertainty for the goodness-of-fit measures using parametric bootstrap (Langeheine et al., 1996), which can be used to select the numbers of latent classes and latent clusters.

Models

Suppose that there are G groups, and the gth group consists of n_g observations for $g = 1, \ldots, G$, and there are M categorical manifest items, where the mth item has r_m categories for $m = 1, \ldots, M$. Let $\mathbf{Y}_{ig} = (Y_{ig1}, \ldots, Y_{igM})^\top$ and $\mathbf{y}_{ig} = (y_{ig1}, \ldots, y_{igM})^\top$ denote a set of item variables and their responses given by the ith individual within the gth group, respectively. The number of possible response patterns of $\mathbf{Y}_{ig}$ is $\prod_{m=1}^{M} r_m$, and it is likely that most of these response patterns are sparse. The multiple-group LCA assumes that associations among manifest items can be explained by the latent classifier L_{ig}, where L_{ig} is the latent class variable having C categories for the ith individual within the gth group. To reflect multiple-group data structures, we discuss two different LCA approaches, namely fixed-effect and random-effect LCA.

Fixed-effect latent class analysis

The fixed-effect LCA can reflect group differences in latent structure by specifying an LCR model for a given subgroup. We extend the simultaneous LCA (Clogg & Goodman, 1985) by incorporating logistic regression in the class prevalence and refer it to *multiple-group latent class regression* (mgLCR). Let $\mathbf{x}_{ig} = (x_{ig1}, \ldots, x_{igp})^\top$ be a subject-specific $p \times 1$ vector of covariates for the ith individual within the gth group, either discrete or continuous. Then, the observed-data likelihood of mgLCR for the ith individual within the gth group can be specified as

$$\begin{aligned}
\mathcal{L}_{ig} &= \sum_{c=1}^{C} P(\mathbf{Y}_{ig} = \mathbf{y}_{ig}, L_{ig} = c \mid \mathbf{x}_{ig}) \\
&= \sum_{c=1}^{C} \left[P(L_{ig} = c \mid \mathbf{x}_{ig}) \prod_{m=1}^{M} P(y_{igm} = k \mid L_{ig} = c) \right] \\
&= \sum_{c=1}^{C} \left[\gamma_{c|g}(\mathbf{x}_{ig}) \prod_{m=1}^{M} \prod_{k=1}^{r_m} \rho_{mk|cg}^{I(y_{igm}=k)} \right], \qquad (1)
\end{aligned}$$

where $I(y_{igm} = k)$ is an indicator function that is equal to 1 when the response to the mth item from the ith individual within the gth group is k and is otherwise equal to 0. The likelihood given in (1) contains two types of parameters:

1. $\rho_{mk|cg}$ represents the probability of an individual within the gth group responding k to the mth item given his or her latent class as c.

2. $\gamma_{c|g}(\mathbf{x}_{ig})$ is the probability of the ith individual belonging to the latent class c within the gth group, which could be influenced by the subject-specific covariates $\mathbf{x}_{ig}$.

The ρ-parameter is the measurement parameter in mgLCR (i.e., item-response probability), describing a tendency of individuals in a latent class c to respond to the mth item for $m = 1, \ldots, M$. Comparison of estimated item-response probabilities across groups is a valuable strategy for quantifying measurement invariance because they solely determine the meaning of the latent class. By comparing the model fit with the parameter held constant across groups (i.e., $\rho_{mk|c} = \rho_{mk|c1} = \cdots = \rho_{mk|cG}$ for $k = 1, \ldots, r_m$, $m = 1, \ldots, M$, and $c = 1, \ldots, C$) against an alternative model with freely varying parameters, we obtain evidence on whether measurement invariance across groups can be assumed. As given in (1), the subject-specific covariates $\mathbf{x}_{ig}$ may influence the probability of the individual belonging to a specific class in the form of logistic regression as

$$\gamma_{c|g}(\mathbf{x}_{ig}) = P(L_{ig} = c \mid \mathbf{x}_{ig}) = \frac{\exp(\alpha_{c|g} + \mathbf{x}_{ig}^{\top}\boldsymbol{\beta}_{c|g})}{\sum_{c'=1}^{C} \exp(\alpha_{c'|g} + \mathbf{x}_{ig}^{\top}\boldsymbol{\beta}_{c'|g})}, \tag{2}$$

where the coefficient vector $\boldsymbol{\beta}_{c|g} = (\beta_{1c|g}, \ldots, \beta_{pc|g})^{\top}$ can be interpreted as the expected change in the log odds of belonging to a class c versus belonging to the referent class C (i.e., $\alpha_{C|g} = 0$ and $\boldsymbol{\beta}_{C|g} = \mathbf{0}$ for $g = 1, \ldots, G$). Then, the observed log-likelihood function for the mgLCR model can be specified as

$$\ell_{mgLCR} = \sum_{g=1}^{G} \sum_{i=1}^{n_g} \log \mathcal{L}_{ig}. \tag{3}$$

It should be noted that, similar to item-response probabilities, coefficients of logistic regression can be constrained to be equal across subgroups (i.e., $\boldsymbol{\beta}_c = \boldsymbol{\beta}_{c|1} = \cdots = \boldsymbol{\beta}_{c|G}$ for $c = 1, \ldots, C$) to test whether the effects of covariates are identical across groups.

Random-effect latent class analysis

The random-effect LCA considers the group variation in the latent class prevalence for each group using random coefficients, for example,

$$P(L_{ig} = c) = \frac{\exp(\alpha_c + \sigma_c \lambda_g)}{\sum_{c'=1}^{C} \exp(\alpha_{c'} + \sigma_{c'} \lambda_g)},$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_G)^{\top}$ represents group variation in the class prevalence. In the parametric random-effect LCA, the random coefficients are assumed to be derived from parametric distributions such as standard normal distribution. However, the nonparametric approach assumes no specific distribution; rather, it only assumes that random coefficients follow a specific probability mass function with some mass points. In other words, the nonparametric approach employs categorical level-2 latent variable (i.e., latent cluster) U_g whose probability mass function is

$P(U_g = w) = \delta_w$ for $w = 1, \ldots, W$. Using the classification mechanics of LCA, the latent cluster membership of level-2 units can be identified by the small number of representative patterns of class prevalences in multiple groups. Therefore, the meaning of the wth level of latent cluster variable is determined by the prevalence of latent classes $P(L_{ig} = c \mid U_g = w)$ for $c = 1, \ldots, C$. Considering latent cluster variable as a group variable, the nonparametric approach provides more meaningful interpretations in group comparison than parametric approach; we can examine whether the latent class structure differs across latent cluster memberships. Therefore, we focus on the nonparametric random-effect LCA, hereafter referred to as *nonparametric latent class regression* (npLCR).

The observed-data likelihood of npLCR for the gth group can be expressed by

$$\begin{aligned}
\mathcal{L}_g &= \sum_{w=1}^{W} P(U_g = w) \prod_{i=1}^{n_g} \left\{ \sum_{c=1}^{C} P(Y_{ig} = y_{ig}, L_{ig} = c \mid U_g = w, \mathbf{x}_{ig}, \mathbf{z}_g) \right\} \\
&= \sum_{w=1}^{W} P(U_g = w) \prod_{i=1}^{n_g} \left\{ \sum_{c=1}^{C} P(L_{ig} = c \mid U_g = w, \mathbf{x}_{ig}, \mathbf{z}_g) \prod_{m=1}^{M} P(Y_{igm} = k \mid L_{ig} = c) \right\} \\
&= \sum_{w=1}^{W} \delta_w \prod_{i=1}^{n_g} \left\{ \sum_{c=1}^{C} \gamma_{c|w}(\mathbf{x}_{ig}, \mathbf{z}_g) \prod_{m=1}^{M} \prod_{k=1}^{r_m} \rho_{mk|c}^{I(y_{igm}=k)} \right\}, \qquad (4)
\end{aligned}$$

where $\mathbf{x}_{ig} = (x_{ig1}, \ldots, x_{igp})^\top$ and $\mathbf{z}_g = (z_{g1}, \ldots, z_{gq})^\top$ denote vectors of subject-specific (i.e., level-1) and group-specific (i.e., level-2) covariates for $i = 1, \ldots, n_g$ and $g = 1, \ldots, G$, respectively. The likelihood given in (4) contains three types of parameters:

1. $\rho_{mk|c}$ represents the probability of an individual responding k to the mth item given his or her latent class as c.

2. $\gamma_{c|w}(\mathbf{x}_{ig}, \mathbf{z}_g)$ is the probability of the ith individual within the gth group belonging to the latent class c given the latent cluster w, which could be influenced by the subject-specific covariates $\mathbf{x}_{ig}$ and the group-specific covariates $\mathbf{z}_g$.

3. δ_w represents the latent cluster prevalence for $w = 1, \ldots, W$.

The class prevalence can be modeled using the logistic regression as

$$\gamma_{c|w}(\mathbf{x}_{ig}, \mathbf{z}_g) = P(L_{ig} = c \mid U_g = w, \mathbf{x}_{ig}, \mathbf{z}_g) = \frac{\exp(\alpha_{c|w} + \mathbf{x}_{ig}^\top \boldsymbol{\beta}_{1c|w} + \mathbf{z}_g^\top \boldsymbol{\beta}_{2c})}{\sum_{c'=1}^{C} \exp(\alpha_{c'|w} + \mathbf{x}_{ig}^\top \boldsymbol{\beta}_{1c'|w} + \mathbf{z}_g^\top \boldsymbol{\beta}_{2c'})}, \qquad (5)$$

where vectors $\boldsymbol{\beta}_{1c|w} = (\beta_{11c|w}, \ldots, \beta_{1pc|w})^\top$ and $\boldsymbol{\beta}_{2c} = (\beta_{21c}, \ldots, \beta_{2qc})^\top$ are logistic regression coefficients for level-1 and level-2 covariates, respectively. Then, the observed log-likelihood of npLCR is specified as

$$\ell_{npLCR} = \sum_{g=1}^{G} \log \mathcal{L}_g. \qquad (6)$$

Note that coefficients for level-1 covariates depend on both latent classes and clusters, while coefficients for level-2 covariates depend only on latent class membership. We may refer the model (5) to the random slope model as coefficients for level-1 covariates are different across latent clusters. The coefficients for level-1 covariates can be constrained to be equal across clusters (i.e., $\boldsymbol{\beta}_{1c} = \boldsymbol{\beta}_{1c|1} = \cdots = \boldsymbol{\beta}_{1c|W}$ for $c = 1, \ldots, C$) to test whether the effects of level-1

covariates are identical across all latent cluster memberships. It should also be noted that the measurement invariance is assumed across latent cluster memberships in npLCR (i.e., $\rho_{mk|c} = \rho_{mk|c1} = \cdots = \rho_{mk|cW}$ for $k = 1, \ldots, r_m$, $m = 1, \ldots, M$, and $c = 1, \ldots, C$). If not, the item response probability may vary across latent cluster memberships, suggesting that the latent class structure itself is different between latent clusters. Thus, it no longer makes sense to use latent class prevalences as identifiers for the latent cluster membership.

Estimation for fixed-effect latent class analysis

The package `glca` finds the maximum-likelihood (ML) estimates for mgLCR and npLCR using expectation-maximization (EM) algorithm (Dempster et al., 1977). The EM algorithm iterates two steps: expectation step (E-step) and maximization step (M-step) in order to find the solution maximizing the log-likelihood functions given in (3) and (6).

For mgLCR, E-step computes the posterior probabilities

$$\theta_{ig(c)} = P(L_{ig} = c \mid \mathbf{Y}_{ig} = \mathbf{y}_{ig}, \mathbf{x}_{ig}) = \frac{\gamma_{c|g}(\mathbf{x}_{ig}) \prod_{m=1}^{M} \prod_{k=1}^{r_m} \rho_{mk|cg}^{I(y_{igm}=k)}}{\sum_{c'=1}^{C} \gamma_{c'|g}(\mathbf{x}_{ig}) \prod_{m=1}^{M} \prod_{k=1}^{r_m} \rho_{mk|c'g}^{I(y_{igm}=k)}}$$

with current estimates for $i = 1, \ldots, n_g$, $g = 1, \ldots, G$, and $c = 1, \ldots, C$. M-step maximizes the complete-data likelihood (i.e., the likelihood for the cross-classification by L_{ig} and $\mathbf{y}_{ig}$) with respect to β- and ρ-parameters. In particular, when all values of $\theta_{ig(c)}$ are known, updated estimates for β-parameters can be calculated by the Newton-Raphson algorithm for multinomial logistic regression given in (2), provided that the computational routine allows fractional responses rather than integer counts (Bandeen-Roche et al., 1997). Therefore, the package `glca` conducts one-cycle of Newton-Raphson algorithm to update β-parameters at every iteration in M-step. If there is no covariate in the model, the class prevalence can be updated directly without estimating β-parameters as $\hat{\gamma}_{c|g} = P(L_{ig} = c) = \sum_{i=1}^{n_g} \theta_{ig(c)}/n_g$ for $c = 1, \ldots, C$ and $g = 1, \ldots, G$. The item-response probabilities, $\rho_{mk|cg}$ can be interpreted as parameters in a multinomial distribution when $\theta_{ig(c)}$ is known, so we have

$$\hat{\rho}_{mk|cg} = \frac{\sum_{i=1}^{n_g} \theta_{ig(c)} I(y_{igm} = k)}{\sum_{i=1}^{n_g} \theta_{ig(c)}}$$

for $k = 1, \ldots, r_m$, $m = 1, \ldots, M$, $c = 1, \ldots, C$, and $g = 1, \ldots, G$. Under the measurement invariance assumption (i.e., $\rho_{mk|c} = \rho_{mk|c1} = \cdots = \rho_{mk|cG}$), the ρ-parameter will be updated as

$$\hat{\rho}_{mk|c} = \frac{\sum_{g=1}^{G} \sum_{i=1}^{n_g} \theta_{ig(c)} I(y_{igm} = k)}{\sum_{g=1}^{G} \sum_{i=1}^{n_g} \theta_{ig(c)}}$$

for $k = 1, \ldots, r_m$, $m = 1, \ldots, M$, and $c = 1, \ldots, C$.

Estimation for random-effect latent class analysis

For npLCR, E-step involves the joint posterior probability

$$\theta_{g(w,c_1,\ldots,c_{n_g})} = P(U_g = w, L_{1g} = c_1, \ldots, L_{n_g g} = c_{n_g} \mid \mathbf{Y}_g = \mathbf{y}_g, \mathbf{x}_g, \mathbf{z}_g), \tag{7}$$

where $\mathbf{y}_g = (\mathbf{y}_{1g}^\top, \ldots, \mathbf{y}_{n_g g}^\top)^\top$ and $\mathbf{x}_g = (\mathbf{x}_{1g}^\top, \ldots, \mathbf{x}_{n_g g}^\top)^\top$ are all observations for M manifest items and p subject-specific covariates from the gth group, respectively. As shown in (7), computational complexity increases exponentially as the number of individuals per group n_g increases in

E-step for npLCR; when the model has W latent clusters and C latent classes, the implementation of the E-step would yield dimensional complexity proportional to $W \times C^{n_g}$ for each group. It is not computationally feasible even with a moderate number of individuals per group. Besides, M-step for npLCR requires only some marginal versions of posterior probabilities rather than the joint posterior probability given in (7). Therefore, the E-step can be modified to alleviate the computational complexity by accommodating the hierarchical structure of npLCR. Vermunt, 2003 proposed the upward-downward (UD) algorithm for calculating the marginal posterior probability directly in the E-step. The UD algorithm is similar to the forward-backward (FB) algorithm for handling hidden Markov models (Baum et al., 1970; Juang & Rabiner, 1991). In the UD algorithm, marginal posterior probabilities $\theta_{ig(w,c)}$ can be calculated as

$$\begin{aligned} \theta_{ig(w,c)} &= P(U_g = w, L_{ig} = c \mid \mathbf{Y}_g = \mathbf{y}_g, \mathbf{x}_g, \mathbf{z}_g) \\ &= P(U_g = w \mid \mathbf{Y}_g = \mathbf{y}_g, \mathbf{x}_g, \mathbf{z}_g) P(L_{ig} = c \mid U_g = w, \mathbf{Y}_{ig} = \mathbf{y}_{ig}, \mathbf{x}_{ig}, \mathbf{z}_g) \\ &= \theta_{g(w)} \theta_{ig(c|w)} \end{aligned} \tag{8}$$

for $i = 1, \ldots, n_g$, $g = 1, \ldots, G$, $c = 1, \ldots, C$, and $w = 1, \ldots, W$. The marginal posterior probability, $\theta_{ig(w,c)}$ is the product of *upward probability*, $\theta_{g(w)}$ and *downward probability*, $\theta_{ig(c|w)}$. In (8), it should be noted that an individual's class membership is assumed to depend only on his/her observations, which can be presented as $P(L_{ig} = c \mid U_g = w, \mathbf{Y}_g = \mathbf{y}_g, \mathbf{x}_g, \mathbf{z}_g) = P(L_{ig} = c \mid U_g = w, \mathbf{Y}_{ig} = \mathbf{y}_{ig}, \mathbf{x}_{ig}, \mathbf{z}_g)$ for $i = 1, \ldots, n_g$. Then, the upward and downward probabilities are easily calculated as

$$\theta_{g(w)} = \frac{\delta_w \prod_{i=1}^{n_g} \left\{ \sum_{c=1}^{C} \gamma_{c|w}(\mathbf{x}_{ig}, \mathbf{z}_g) \prod_{m=1}^{M} \prod_{k=1}^{r_m} \rho_{mk|c}^{I(y_{igm}=k)} \right\}}{\sum_{w=1}^{W} \delta_w \prod_{i=1}^{n_g} \left\{ \sum_{c=1}^{C} \gamma_{c|w}(\mathbf{x}_{ig}, \mathbf{z}_g) \prod_{m=1}^{M} \prod_{k=1}^{r_m} \rho_{mk|c}^{I(y_{igm}=k)} \right\}} \quad \text{and}$$

$$\theta_{ig(c|w)} = \frac{\gamma_{c|w}(\mathbf{x}_{ig}, \mathbf{z}_g) \prod_{m=1}^{M} \prod_{k=1}^{r_m} \rho_{mk|c}^{I(y_{igm}=k)}}{\sum_{c'=1}^{C} \gamma_{c'|w}(\mathbf{x}_{ig}, \mathbf{z}_g) \prod_{m=1}^{M} \prod_{k=1}^{r_m} \rho_{mk|c'}^{I(y_{igm}=k)}}$$

with current estimates, respectively. M-step maximizes the complete-data likelihood (i.e., the likelihood for the cross-classification by U_g, L_{ig} and $\mathbf{y}_{ig}$) with respect to $\boldsymbol{\beta}_{1c|w}$, $\boldsymbol{\beta}_{2c}$, and $\rho_{mk|c}$. In particular, when $\theta_{ig(w,c)}$ is known, updated estimates for β-parameters can be calculated by Newton-Raphson algorithm for multinomial logistic regression given in (5), provided that the computational routine allows fractional responses rather than integer counts (Bandeen-Roche et al., 1997). Therefore, the package `glca` conducts one-cycle of Newton-Raphson algorithm to update β-parameters at every iteration in M-step. If there is no covariate in the model, the class prevalence can be updated directly without estimating β-parameters as $\hat{\gamma}_{c|w} = P(L_{ig} = c \mid U_g = w) = \sum_{g=1}^{G} \sum_{i=1}^{n_g} \theta_{ig(w,c)} / \sum_{g=1}^{G} \theta_{g(w)}$ for $c = 1, \ldots, C$ and $w = 1, \ldots, W$. The cluster prevalence δ_w and the item-response probabilities $\rho_{mk|c}$ can be interpreted as parameters in multinomial distributions, so we have

$$\hat{\delta}_w = \frac{\sum_{g=1}^{G} \theta_{g(w)}}{G} \quad \text{and} \quad \hat{\rho}_{mk|c} = \frac{\sum_{g=1}^{G} \sum_{i=1}^{n_g} \theta_{ig(c)} I(y_{igm} = k)}{\sum_{g=1}^{G} \sum_{i=1}^{n_g} \theta_{ig(c)}} \tag{9}$$

for $w = 1, \ldots, W$, $c = 1, \ldots, C$, $k = 1, \ldots, r_m$, and $m = 1, \ldots, M$. The marginal posterior probability used in (9) are easily obtained by $\theta_{ig(c)} = \sum_{w=1}^{W} \theta_{ig(w,c)}$.

Handling missing data

Missing data occur in nearly all empirical data, despite the vigorous efforts of researchers to prevent it. Missing data cause two general problems. First, if subjects with any missing data

on the variables are removed from the dataset, the sample can be very small especially when the number of missing values is large. This can lead to a great loss of information and poor statistical power. Second, frequently the subjects who provide incomplete data are different from those who provide complete data. If adjustments are not made for these differences, results may be biased.

In the case of random missing-data mechanisms (i.e., ignorable missing data) such as missing completely at random (MCAR) and missing at random (MAR) (Little & Rubin, 2019), two methods for dealing with missing data are typically available: full-information maximum likelihood (FIML) and multiple imputation (MI, Schafer, 1997). In MI plausible values are imputed multiple times in place of missing values to create multiple complete datasets. The use of MI for multiple-group LCA has an advantage in that missing data on covariates and group variable can be handled. However, the disadvantage is that LCA must be fitted separately for each imputed complete dataset, and the results must be combined to obtain the final estimates. FIML is a model-based missing data procedure where model estimates are adjusted on the basis of all of the information provided by subjects with complete data and partially complete data. Most software packages for LCA employ a FIML approach because it requires no additional input from the user other than specifying that what code is used to denote missing data. However, this approach cannot handle missing data when missingness occurs in group variable or covariates in multiple-group LCA.

The package `glca` estimates model parameters using a FIML approach when some responses are found missing on manifest items: in E-step the missing responses are excluded from computing the posterior probability; and in M-step the indicator $I(y_{igm} = k)$ for the missing response is replaced with the updated ρ-parameter from previous iteration. In short, the package `glca` can handle any ignorable missing data on manifest items, but individuals with missing data on group variable or any covariate are deleted from the analysis. However, missing values on group variable and covariates can be treated using multiple multivariate imputation by chained equations (MICE, van Buuren et al., 2006), which is implemented in the R package `mice` (van Buuren & Groothuis-Oudshoorn, 2011). MICE imputation could be used to create multiple sets of complete group variable and covariates for multiple-group LCA. Each complete dataset can be analyzed using the package `glca` and combining results across imputed datasets are easily obtained.

Finding global maximum

Since the log-likelihood of LCA may have several local-maxima problem, the estimated parameters from EM algorithm can be deviated from the globally optimal solution. To cope with this problem, we recommend starting the algorithm using several different initial sets of random values and ascertaining whether they consistently converge to the same solution. If they converge, the solution can be considered as the ML estimates. If not, we recommend examining the distribution of the likelihood values and selecting the largest likelihood value, which usually corresponds to the ML solution. The package `glca` allows investigators to try different starting values either by using random starting values or providing their own starting values. An investigator can select the number of initial sets of random values (default is 10) in the package `glca`, and then the package iterates EM algorithm a small number of times (default is 50) for each set of random values. Among the initial sets of model parameters, those producing the largest value of likelihood will be chosen for the main iteration.

Standard error calculation

The standard error of the estimated parameters can be calculated using the observed empirical information matrix (Mclachlan & Krishnan, 2007, p. 114),

$$\boldsymbol{I}_e(\hat{\boldsymbol{\Psi}}; \mathbf{Y}) = \sum_{g=1}^{G} \mathbf{s}(\mathbf{Y}_g; \hat{\boldsymbol{\Psi}}) \mathbf{s}^\top (\mathbf{Y}_g; \hat{\boldsymbol{\Psi}}),$$

where $\mathbf{s}(\mathbf{Y}_g; \hat{\boldsymbol{\Psi}})$ is the score function of the parameter vector $\boldsymbol{\Psi}$ for the gth group, evaluated at their MLE $\hat{\boldsymbol{\Psi}}$. In the parameter vector $\boldsymbol{\Psi}$, all probability parameters such as δ and ρ-parameters are transformed into free parameters using baseline logit function. The variance of $\hat{\boldsymbol{\Psi}}$ can be obtained by the inverse of $\boldsymbol{I}_e(\hat{\boldsymbol{\Psi}}; \mathbf{Y})$. However, as our target parameters are a re-parameterized version of $\boldsymbol{\Psi}$, we should apply delta method to the variance of $\hat{\boldsymbol{\Psi}}$. Let $q(\boldsymbol{\Psi})$ denote the original parameters of multiple-group latent class models. Then, the variance-covariance matrix of the estimates is

$$\mathsf{Var}\left(q(\hat{\boldsymbol{\Psi}})\right) = J_q(\hat{\boldsymbol{\Psi}}) \mathsf{Var}(\hat{\boldsymbol{\Psi}}) J_q(\hat{\boldsymbol{\Psi}})^\top,$$

where $J_q(\hat{\boldsymbol{\Psi}})$ is the Jacobian matrix of the function $q()$ evaluated at the MLE of $\boldsymbol{\Psi}$. Details of the score functions and the Jacobian matrices are provided in Appendix.

Assessing absolute model fit for measuring goodness-of-fit

Absolute model fit refers to whether a specified multiple-group latent class model provides an adequate representation of the data. Typically, the analyst assesses absolute model fit by fitting a particular model to the observed data and testing the null hypothesis that the observed data has been produced by the fitted model. Thus, one usually hopes to find a model for which the null hypothesis is not rejected. This hypothesis test for LCA is based on a contingency table; the expected cell counts are estimated according to the specified model and its estimated parameters, then compared to the observed cell counts. The likelihood-ratio statistic, G^2 (Agresti, 2013) is used to assess absolute model fit in the package `glca`. The G^2 test statistic is derived from the difference in the log-likelihood values between the fitted model and the saturated model (i.e., expected and observed cell counts, respectively), where the residual degree of freedom is calculated by subtracting the number of parameters in the fitted model from those in the saturated model. The number of parameters for the saturated model is the lesser of number of possible combinations of categorical variables and number of cases in the model.

It should be noted that the contingency table for the LCA type of model is commonly sparse. When there are many cells containing very few observations in the cross-classification table, the large-sample approximation to the chi-square distribution for the G^2 statistic is not appropriate. In such case, the package `glca` allows us to conduct goodness-of-fit test using the bootstrap likelihood-ratio test (BLRT) statistic (Langeheine et al., 1996). This approach generates random datasets multiple times using the estimated parameters and calculates the G^2 statistic for each generated dataset. In BLRT the resulting distribution of the G^2 statistic across the random datasets is used as the reference distribution. The relative position of the G^2 statistic obtained from the original dataset within the reference distribution can be used as a measure of absolute model fit. In fact, the right tail probability of the observed G^2 value is regarded as a bootstrap p-value. For example, if the observed G^2 value falls in the uppermost tail of the reference distribution, we may conclude that this test statistic is unlikely observed under the model corresponding to the null hypothesis. Such finding would provide evidence to reject the null hypothesis.

Assessing relative model fit for exploring group differences

When comparing two or more groups in multiple-group latent class model, it should be checked if the latent features are identical or not across groups. The relative model fit refers to deciding which of two or more models represents a better fit to a particular dataset. The measurement invariance in multiple-group LCA can be tested by comparing the model fits of constrained versus unconstrained model; the unconstrained (full) model allows all parameters to vary across groups, while the constrained (reduced) model allows only the class prevalences to vary but item-response probabilities to be equal across groups. The package `glca` conducts the chi-square likelihood-ratio test (LRT) to assess relative model fit by comparing two competing models for testing measurement invariance.

Similar to the item-response probabilities, the coefficients for the level-1 covariates can also be tested for equality across groups using chi-square LRT in the package `glca`. By comparing the fit of reduced model with the coefficients held constant across groups (i.e., $\boldsymbol{\beta}_c = \boldsymbol{\beta}_{c|1} = \cdots = \boldsymbol{\beta}_{c|G}$ in mgLCR and $\boldsymbol{\beta}_{1c} = \boldsymbol{\beta}_{1c|1} = \cdots = \boldsymbol{\beta}_{1c|W}$ in npLCR for $c = 1, \ldots, C$) against full model with freely varying coefficients, we obtain evidence on whether the effects of level-1 covariates on latent class prevalences can be assumed to be identical across groups. To make the group comparison for coefficients valid, the assumption of measurement invariance must be met to ensure consistent meaning of latent classes across groups.

The deviance statistic, a test statistic of LRT for relative model fit, is obtained by twice the difference in the log-likelihood values of two competing models. The degree of freedom for deviance statistic is the difference in the number of free parameters of the two multiple-group latent class models. For example, the validity of the measurement invariance assumption can be tested by calculating the log-likelihood from the model where item-response probabilities are constrained to be equal across subgroups and comparing it with the log-likelihood from the freely estimated model.

Choosing the numbers of latent classes and latent clusters

The chi-square LRT cannot be used to compare latent class models with a different number of latent classes or clusters because these two models are not nested. Thus, the package `glca` provides several information criteria commonly used in LCA such as Akaike's information criterion (AIC), Bozdogan's criterion (CAIC), and Schwartz's Bayesian information criterion (BIC) (Akaike, 1974; Bozdogan, 1987; Schwarz, 1978) to compare the fit of non-nested competing models; the model with a smaller AIC (or BIC) value is preferred. Another model fit index provided by the package is entropy, which is widely used in research practices although it can be a poor measure for model selection as it often depends on the number of classes (Collins & Lanza, 2009). The model with relatively higher entropy value is preferred.

The package `glca` also generates the empirical distribution of the deviance statistic to help select a better model between two non-nested competing models with a different number of latent classes or clusters using BLRT (Langeheine et al., 1996). The null hypothesis is the simpler model is adequate. Thus, the bootstrap sample will be drawn from the simpler model. Using a generated bootstrap sample, both competing models are estimated and the deviance between these two models is calculated. By repeating this procedure multiple times, we can construct the reference distribution of the deviance. Similar to the bootstrap p-value for the G^2 statistic, the relative position of observed deviance within the reference distribution presents bootstrap p-value; the null model with a bootstrap p-value > 0.05 is preferred with a significance level of $\alpha = 0.05$. An important advantage of using BLRT is that this method can be applied to the

test for comparing two nested latent class models even when the condition for large-sample approximation is not satisfied. It should be noted that the optimal model should be selected by comprehensively considering both conceptual and analytical implications, and the quantitative goodness-of-fit statistics.

The glca package

The R package `glca` contains a collection of functions for exploring group differences in latent structure using multiple-group latent class models. In the following we describe the functionality of the package `glca`.

Description of the main function `glca()`

The main function of this package, called `glca()`, fits a wide range of multiple-group latent class models including fixed-effect and random-effect LCA in order to examine whether the latent structure is identical across groups. The `glca()` function can be called with

```
glca(formula, data, group, nclass, ncluster, std.err, measure.inv,
     coef.inv, init.param, n.init, testiter, maxiter, eps, na.rm,
     seed, verbose)
```

The function `glca()` uses the formula expression in order to specify a multiple-group latent class model. For example, suppose there are four manifest items, Y_1, Y_2, Y_3, and Y_4 in the dataset. These items must be combined as `item(Y1, Y2, Y3, Y4)` and located on the left hand side of the formula. Without any covariate, the formula definition takes the form:

```
formula <- item(Y1, Y2, Y3, Y4) ~ 1
```

The item can be specified by the prefix or suffix of the manifest items as follows:

```
formula <- item(starts.with = "Y") ~ 1
```

Any covariate can be incorporated by replacing `~ 1` with the desired function of covariates. For example, the npLCR with one level-1 covariate (X_1) and one level-2 covariate (Z_1) can be fitted using the following formula:

```
formula <- item(starts.with = "Y") ~ X1 + Z1
```

It should be noted that `glca()` identifies the type of covariates automatically. The function has following arguments:

`data`: The input data, `data.frame` or `matrix`, with individuals in rows and group variable, level-1 and level-2 covariates, and individuals' responses to manifest items in the columns. The data could contain multichotomous responses to manifest items.

`group`: Argument that indicates group variable which has the same length as manifest items on the formula. If `group = NULL` (default), LCA or LCR is fitted.

`nclass`: Integer scalar specifying the number of latent classes. In the default setting, `nclass = 3`.

`ncluster`: Integer scalar specifying the number of latent clusters (default `ncluster = 0`).

std.err: Logical value for whether calculating standard errors for estimates (default std.err = TRUE).

measure.inv: Logical value for whether measurement invariance across groups is assumed (default measure.inv = TRUE).

coeff.inv: Logical value for whether coefficients for level-1 covariates are identical across groups or latent clusters (default coeff.inv = TRUE).

init.param: A set of model parameters to be used as a user-specified initial values for EM algorithm. It should be list with the named parameters and have same structure of param of the glca() output. In default, initial parameters are randomly generated (i.e., default init.param = NULL).

n.init: Integer scalar specifying the number of randomly generated parameter sets to be used as initial values for EM algorithm in order to avoid local maxima problem (default n.init = 10).

testiter: Integer scalar specifying the number of iterations in EM algorithm for each initial parameter set in order to select the best initial parameter set. The initial parameter set that provides the largest log-likelihood is selected for main iteration to estimate model parameters (default testiter = 50).

maxiter: Integer scalar specifying the maximum number of iterations for EM algorithm (default maxiter = 5000).

eps: A convergence tolerance value. When the largest absolute difference between former estimates and current estimates is less than eps, EM algorithm will stop updating and consider the convergence to be reached (default eps = 1e-06).

na.rm: Logical value for whether the routine deletes the rows that have at least one missing manifest item. If na.rm = FALSE (default), FIML approach will be conducted under MAR condition.

seed: In default, the set of initial parameters is drawn randomly. As the same value for seed guarantees the same initial parameters to be drawn, this argument can be used for reproducibility of estimation results.

verbose: Logical value indicating whether glca() should print the estimation procedure onto the screen (default verbose = TRUE).

The output of glca() is list or data.frame that contains information for model specification and results of the data analysis using the specified model. The function glca() returns following outputs:

model: A list containing information on the specified multiple-group latent class models.

var.names: A list containing names of data.

datalist: A list of data used for fitting.

param: A list of parameter estimates.

std.err: A list of standard errors for param.

`coefficient`: A list of logistic regression coefficients for latent class prevalences.

`gof`: A list of goodness-of-fit measures.

`convergence`: A list containing information on convergence.

`posterior`: For mgLCR, `posterior` is a `data.frame` of posterior probabilities belonging to specific latent class for each individual. For npLCR, `posterior` is a `list` containing three type of posterior probabilities; probabilities of belonging to latent cluster for each group, belonging to latent cluster for each latent class, and belonging to latent class for each individual.

For objects from the `glca()`, generic functions for `print()`, `summary()`, `coef()`, `logLik()`, `reorder()`, and `plot()` are available. The generic function `print()` can be accessed with `print(x)` where `x` is an object from the function `glca()`. This function is used to print information from the object `x` to console. The function calls together with the names of variable, model specification, the number of observations, and the number of parameters used in the analysis.

The function `summary()` is a generic function that is summarizing results from the `glca()`. The function can be called via `summary(x)`. The output of the `summary(x)` has two main components; the first component contains information on model specification, and the second component contains parameter estimates.

Generic functions `coef()` and `logLik()` can also be used. The estimates of regression coefficients and their standard errors of the object `x` from the function `glca()` will be extracted via `coef(x)`. Odds ratios, t-values, and their respective p-values for the estimated coefficient will also be extracted. The function `logLik()` extracts log-likelihood and degree of freedom for the model which enables to use other statistical `R` function such as `AIC()` and `BIC()`.

The function `reorder()` is a generic function for reordering estimated parameters. Since the latent classes or clusters can be switched according to the initial value of EM algorithm, the order of estimated parameters can be arbitrary. Researchers may desire to rearrange the order of latent variables for convenient interpretation of estimated parameters. The ordering can be designated by users (`class.order`, `cluster.order`) and can be determined by the magnitude of the probability of responding the first manifest item with the first option.

A generic function for plotting parameter estimates of the specified model is also available to the user. The user can plot parameters of the object `x` with `plot(x)`. This function returns three types of plots: item-response probabilities, marginal class prevalences, and class prevalences by group or cluster. The plot for item-response probabilities has M manifest items on the x-axis, the probability of responding to the kth category for the mth item on the y-axis for $k = 1, \ldots, r_m$. If there is a group variable with the argument `measure.inv = FALSE` in the `glca()` function, separate plot for each group is returned. The plot for marginal class prevalences has latent classes on the x-axis, and the probability of belonging to the respective latent class on the y axis. If there is a group variable, additional plot is returned for displaying class prevalences for each group (when mgLCR is fitted) or latent cluster (when npLCR is fitted). This plot has levels of group or latent cluster variable on the x-axis, and the conditional probability of belonging to latent classes for each group or latent cluster on the y-axis.

Description of the `gofglca()` function for assessing the goodness-of-fit

In multiple-group LCA, the first and fundamental step is to select the appropriate number of latent classes and latent clusters. Once the number of latent components are determined, the

next step is to assess relative model fit for exploring various group differences in latent structure across groups or latent clusters. The assessment of model fit described in Sections - is implemented into a function named `gofglca()`. The function syntax is

```
gofglca(x, ..., criteria, test, nboot, maxiter, eps, seed, verbose)
```

and uses following arguments.

`x`: An output of the `glca()` function. Absolute model fit test will be conducted for the model specified in `x`.

`...`: An optional argument, one or more output objects of the function `glca()`, which enables users to test for relative model fit. Each of these optional objects will be compared to one of other objects, which are specified in this argument.

`criteria`: A character vector specifying which type of information criteria should be returned by the function. Default is to return all types of criteria via `c("logLik", "AIC", "CAIC", "BIC", "entropy")`.

`test`: String that controls which distribution to be used for calculating p-value in hypothesis test. Available options are `"NULL"` (default), `"chisq"` (chi-square distribution), and `"boot"` (bootstrap empirical distribution).

`nboot`: The number of bootstrap samples to be generated when `test = "boot"` (default `nboot = 50`).

`maxiter`: Integer scalar specifying the maximum number of EM iterations for each of bootstrap samples (default `maxiter = 500`).

`eps`: A convergence tolerance value used in EM algorithm for each of bootstrap samples (default `eps = 1e-04`).

`seed`: As the same value for `seed` guarantees the same datasets to be generated, this argument can be used for reproducibility of bootstrap results (default `seed = NULL`).

`verbose`: Logical value indicating whether `gofglca()` should print the assessment procedure onto the screen (default `verbose` = TRUE).

The function `gofglca()` provides the table for analysis of goodness-of-fit containing information criteria, residual degrees of freedom, G^2-statistic, and bootstrap p-value for absolute model fit for the designated objects when `test = "boot"`. With optional objects from the function `glca()`, `gofglca()` also provides the table for analysis of deviance given that these objects share the same manifest items. The table for analysis of deviance describes relative model fit using the deviance statistic between two competing models and its p-value using distribution specified in `test` argument.

Applications

In this section, we provide examples in order to demonstrate the implementation of various multiple-group latent class models on two datasets, which are pre-installed in the `glca` package. The first example is used to show the use of the `glca` package for fitting the fixed-effect LCA (i.e., mgLCR) while the second example is used for fitting the random-effect LCA (i.e., npLCR). The commands `install.packages("glca")` and `library(glca)` will install and load the `glca` package, respectively.

Attitudes toward abortion

The data are taken from the 2008 General Social Survey (Smith et al., 2010) in order to demonstrate the functionality of the glca package. We analyze six binary manifest items measuring 355 respondents' attitudes toward abortion following the strategy suggested by McCutcheon, 1987. There are 3 observations who did not respond to any of these six items, so the glca package will automatically remove these three rows and use 352 rows in the analysis. For each item, respondents were asked whether abortion should be legalized under various circumstances: a strong chance of serious defect in the baby (DEFECT), pregnancy is seriously endangering the woman's health (HLTH), pregnancy as a result of rape (RAPE), due to low income, the family cannot afford any more children (POOR), woman is unmarried and has no plans to marry the man (SINGLE), and woman is married but does not want more children (NOMORE). Each item has two possible levels of response (i.e., "YES" or "NO").

Selecting the number of latent classes: In the first step of the analysis, we conduct a series of latent class models to select a number of latent classes. The number of class in the glca() function is set to 2, 3, or 4 as following commands. It should be noted that we use 10 sets of randomly generated initial parameters to avoid the problem of local maxima (i.e., n.init = 10), and the argument is set as seed = 1 in the glca() function to ensure reproductibility of results unless otherwise noted.

```
data("gss08")
f <- item(DEFECT, HLTH, RAPE, POOR, SINGLE, NOMORE) ~ 1
lca2 <- glca(f, data = gss08, nclass = 2, seed = 1)
lca3 <- glca(f, data = gss08, nclass = 3, seed = 1)
lca4 <- glca(f, data = gss08, nclass = 4, seed = 1)
```

The gofglca() function can be used to assess absolute and relative model fit using BLRT by setting test = "boot" for these three latent class models whose number of classes varies from 2 to 4 as follows:

```
gofglca(lca2, lca3, lca4, test = "boot", seed = 1)

#> Model 1: item(DEFECT, HLTH, RAPE, POOR, SINGLE, NOMORE) ~ 1
#>          nclass: 2
#> Model 2: item(DEFECT, HLTH, RAPE, POOR, SINGLE, NOMORE) ~ 1
#>          nclass: 3
#> Model 3: item(DEFECT, HLTH, RAPE, POOR, SINGLE, NOMORE) ~ 1
#>          nclass: 4
#>
#> Goodness of Fit Table :
#>    logLik     AIC    CAIC     BIC entropy Res.Df    Gsq Boot p-value
#> 1 -740.10 1506.20 1569.43 1556.43    0.95     50 135.13         0.00
#> 2 -687.45 1414.90 1512.17 1492.17    0.88     43  29.83         0.36
#> 3 -684.19 1422.38 1553.70 1526.70    0.79     36  23.31         0.56
#>
#> Analysis of Deviance Table :
#>   npar  logLik Df Deviance Boot p-value
#> 1   13 -740.10
#> 2   20 -687.45  7   105.31         0.00
```

```
#> 3   27 -684.19  7     6.52         0.26
```

The output from the function `gofglca()` comprises of two tables; goodness-of-fit table and analysis-of-deviance table. The former shows model fit criteria such as AIC, CAIC, BIC, and entropy, G^2 statistic, and its bootstrap p-value for absolute model fit. In this example, the bootstrap p-values indicate that the two-class model (`Model 1`) fits data poorly (p-value = 0.00), but the three-class and the four-class models (`Model 2` and `Model 3`) fit data adequately (p-value = 0.36 and 0.56, respectively). The latter table displays deviance statistic comparing two competing models and its bootstrap p-value for the relative model fit. The null hypothesis in the test for comparing `Model 1` and `Model 2` (i.e., the fit of the two-class model is not significantly poorer than the fit of the three-class model) should be rejected (p-value = 0.00). In the test for comparing `Model 2` and `Model 3`, however, the bootstrap p-value (= 0.26) indicates that the fit of the four-class model has not been improved significantly compared to the fit of the three-class model. In addition, the three-class model has the smallest value among these three models in the model fit criteria. Therefore, we can conclude that the three-class model is an appropriate for the `gss08` data.

Considering group variable and testing the measurement invariance: As the group information provided in the data, we can consider the multilevel data structure and compare the latent class structure between higher-level units (i.e., groups). The `glca()` function can incorporate group variable by setting `group` argument as the name of group variable in the data. For example, in order to investigate whether attitudes toward legalizing abortions vary by the final degree of respondents, we may set `DEGREE` as group variable by typing `group = DEGREE` in the `glca()` function . `DEGREE` is coded into four categories (i.e., "<= HS", "HIGH SCHOOL", "COLLEGE", and "GRADUATE") indicating from under high school diploma to graduate degree. Moreover, we can implement the test for measurement invariance across groups using the `glca()` function. The measurement invariance assumption can be adjusted through `measure.inv` argument in `glca()`. The default is `measure.inv = TRUE`, constraining item-response probabilities to be equal across groups. The following commands implement two different tests for group variable: (1) test whether class prevalence is significantly influenced by group variable under the measurement-invariant model; and (2) test whether the measurement models differ across groups.

```
mglca1 <- glca(f, group = DEGREE, data = gss08, nclass = 3, seed = 1)
mglca2 <- glca(f, group = DEGREE, data = gss08, nclass = 3, seed = 1,
               measure.inv = FALSE)
gofglca(lca3, mglca1, mglca2, test = "chisq")

#> Model 1: item(DEFECT, HLTH, RAPE, POOR, SINGLE, NOMORE) ~ 1
#>          nclass: 3
#> Model 2: item(DEFECT, HLTH, RAPE, POOR, SINGLE, NOMORE) ~ 1
#>          Group: DEGREE, nclass: 3, measure.inv: TRUE
#> Model 3: item(DEFECT, HLTH, RAPE, POOR, SINGLE, NOMORE) ~ 1
#>          Group: DEGREE, nclass: 3, measure.inv: FALSE
#>
#> Goodness of Fit Table :
#>    logLik     AIC    CAIC     BIC entropy Res.Df   Gsq
#> 1 -687.45 1414.90 1512.17 1492.17    0.88     43 29.83
#> 2 -672.41 1396.83 1523.28 1497.28    0.88    229 87.85
```

```
#> 3 -650.95 1461.89 1850.98 1770.98    0.89    175 44.91
#>
#> Analysis of Deviance Table :
#>   npar  logLik Df Deviance Pr(>Chi)
#> 1   20 -687.45
#> 2   26 -672.41  6    30.07     0.00
#> 3   80 -650.95 54    42.94     0.86
```

Since the model specified in the object `lca3` (`Model 1` in the output from the `goflca()` function) is constructed with six binary items, the number of parameters for the saturated models is $2^6 - 1 = 63$. Therefore, degree of freedom for `Model 1` is $63 - 20 = 43$. The models specified in `mglca1` and `mglca2` (`Model 2` and `Model 3`) are involved with group variable with four categories, and the number of parameters for the saturated models is $2^6 \times 4 - 1 = 255$. Therefore, degrees of freedom for `Model 2` and `Model 3` are $255 - 26 = 229$ and $255 - 80 = 175$, respectively. It should be noted that model comparison has been conducted through chi-squares by setting `test = "chisq"` because `Model 1` is nested in `Model 2`, and `Model 2` is nested in `Model 3`. The analysis-of-deviance table provided by the `gofglca()` function shows that the chi-square p-value for comparing `Model 1` and `Model 2` is 0.00, while the p-value for comparing `Model 2` and `Model 3` is 0.86. Hence, we can deduce that the measurement invariance assumption can be assumed, but class prevalences vary across levels of `DEGREE`.

Testing the equality of coefficients across groups: We can further consider the subject-specific covariates which may influence the probability of the individual belonging to a specific class. Covariates such as `AGE`, `RACE`, and `SEX` in the `gss08` dataset can be incorporated into the model specified in `mglca1`. `AGE` is respondent's age and considered as a numeric variable in the `glca()` function. The respondent's race, `RACE` has three levels (i.e., "WHITE", "BLACK", and "OTHER"), and the respondent's gender, `SEX` is coded as two categories (i.e., "MALE" and "FEMALE"). We can easily implement the test for exploring group differences using the `gofglca()` function. For example, the following commands implement two different tests for `SEX`: (1) test whether the class prevalence is significantly influenced by `SEX` under the model where the coefficients are constrained to be identical across groups; and (2) test for assessing group difference in the effect of `SEX` on class prevalence by specifying the model where coefficients are allowed to vary across groups and comparing it to the model where coefficients are constrained to be identical across groups. Since the iteration is initiated with random parameters, the order of class labels can be switched under the identical ML solution. Therefore, the random seed is set as `seed = 3` in the `glca()` function for the model specified in `mglcr1` below to ensure identical class order as provided in Figure 1.

```
f.sex <- item(DEFECT, HLTH, RAPE, POOR, SINGLE, NOMORE) ~ SEX
mglcr1 <- glca(f.sex, group = DEGREE, data = gss08, nclass = 3, seed = 3)
mglcr2 <- glca(f.sex, group = DEGREE, data = gss08, nclass = 3, seed = 1,
               coeff.inv = FALSE)
gofglca(mglca1, mglcr1, mglcr2, test = "chisq")
```

```
#> Model 1: item(DEFECT, HLTH, RAPE, POOR, SINGLE, NOMORE) ~ 1
#>          Group: DEGREE, nclass: 3, measure.inv: TRUE
#> Model 2: item(DEFECT, HLTH, RAPE, POOR, SINGLE, NOMORE) ~ SEX
#>          Group: DEGREE, nclass: 3, measure.inv: TRUE, Coef.inv: TRUE
#> Model 3: item(DEFECT, HLTH, RAPE, POOR, SINGLE, NOMORE) ~ SEX
```

```
#>          Group: DEGREE, nclass: 3, measure.inv: TRUE, Coef.inv: FALSE
#>
#> Goodness of Fit Table :
#>    logLik     AIC    CAIC     BIC entropy Res.Df    Gsq
#> 1 -672.41 1396.83 1523.28 1497.28    0.88    229  87.85
#> 2 -666.71 1389.42 1525.60 1497.60    0.88    323 149.97
#> 3 -662.04 1392.09 1557.45 1523.45    0.88    317 140.64
#>
#> Analysis of Deviance Table :
#>   npar  logLik Df Deviance Pr(>Chi)
#> 1   26 -672.41
#> 2   28 -666.71  2    11.41     0.00
#> 3   34 -662.04  6     9.33     0.16
```

The models specified in `mglcr1` and `mglcr2` (`Model 2` and `Model 3` in the output from the `goflca()` function) are involved with an additional covariate, `SEX`, and the number of possible cases is $2^6 \times 4 \times 2 - 1 = 511$. However, as only 352 observations are used for the analysis, the number of parameters for the saturated model becomes $352 - 1 = 351$. Therefore, degrees of freedom for `Model 2` and `Model 3` are $351 - 28 = 323$ and $351 - 34 = 317$, respectively. The analysis-of-deviance table provided by the `gofglca()` function shows that `SEX` has a significant impact on the class prevalence (p-value = 0.00) when we compare `Model 1` with `Model 2`. However, the model without any constraint on coefficients (`Model 3`) is not significantly superior to `Model 2` (p-value = 0.16), indicating that the impact of `SEX` is not group specific. Note that `Model 2` is mathematically equivalent to the LCR with covariates `DEGREE` and `SEX` without the interaction terms, but `Model 2` is more intuitive and useful configuration when comparison of latent structures by group is a major concern.

Summarizing the results from the selected model: Based on the previous analysis, we can conclude that measurement models are equivalent across groups (i.e., measurement invariance assumption is satisfied) in the three-class latent class model. In addition, there is a significant effect of `SEX` on the class prevalence, but there is no group difference in the amount of effect.

Figure 1 displays the estimated parameters from the selected model specified in `mglcr1` using the command `plot(mglcr1)`. The line graph in Figure 1 displays the estimated item-response probabilities. We can see that the identified three classes are clearly distinguished by item-response probabilities. Class 1 represents individuals who are in favor of all the six reasons for abortion, while Class 3 represents those who consistently oppose all the six reasons. Individuals in Class 2 seem to distinguish between favoring the first three reasons (i.e., `DEFECT`, `HLTH`, and `RAPE`) and opposing the last three reasons (i.e., `POOR`, `SINGLE`, and `NOMORE`) for abortion. The first bar graph in Figure 1 describes the estimated marginal class prevalences, and the second bar graph displays the estimated class prevalences for each category of group variable. The values in parentheses printed in the x-axis are the group prevalences. Figure 1 displays that the class prevalence varies across groups, and the respondents with higher degrees are more advocate for legalizing abortion than those with lower degrees. The covariate effect can be confirmed through the Wald test for each of estimated odds ratios and coefficient by the command `coef(mglcr1)`.

The `SEX` coefficient results the same, while the intercept determining the class prevalence differs across groups in the selected model. To avoid redundancy and save space, only the results of the `SEX` coefficient are displayed as follows, but the entire output of the object can be displayed by the command `summary(mglcr1)` in the R console.

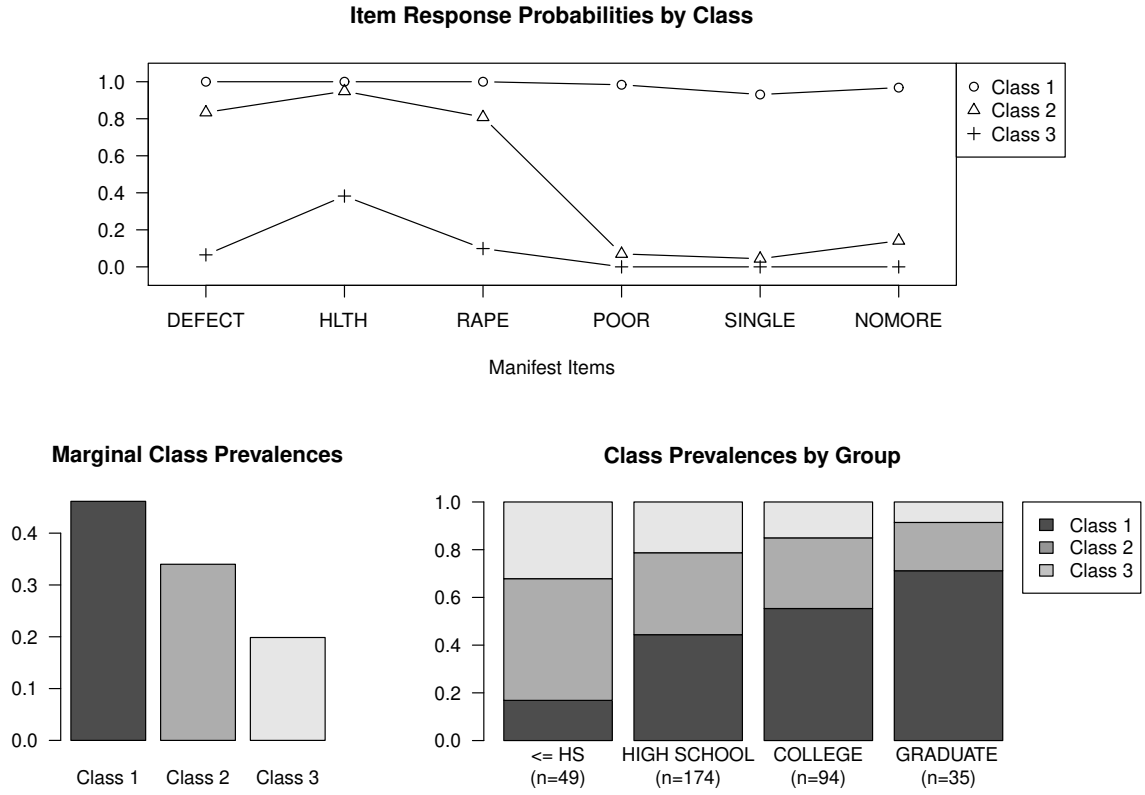

Figure 1: Estimated parameters of the measurement invariant mgLCR for the `gss08` dataset with `DEGREE` as group variable.

```
coef(mglcr1)

#> Coefficients :
#>
#> Class 1 / 3 :
#>           Odds Ratio Coefficient  Std. Error  t value  Pr(>|t|)
#> SEXFEMALE    0.32824    -1.11400     0.09062   -12.29    <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Class 2 / 3 :
#>           Odds Ratio Coefficient  Std. Error  t value  Pr(>|t|)
#> SEXFEMALE    0.35678    -1.03064     0.09971   -10.34    <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated coefficients for SEX and their odds ratios show that females are less advocate for legalizing abortion compared to their male counterparts.

Tobacco smoking behavior

In this subsection, we present an application of the random-effect LCA (i.e., npLCR) to the nyts18 dataset, which is pre-installed in the glca package. The dataset nyts18 comprises five dichotomized manifest items on the life-time experience of several types of tobacco including cigarettes (ECIGT), cigars (ECIGAR), chewing tobacco/snuff/or dip (ESLT), electronic cigarettes (EELCIGT), and hookah or water pipe (EHOOKAH) taken from the National Youth Tobacco Survey 2018 (NYTS 2018, https://www.cdc.gov/tobacco/data_statistics/surveys/nyts). The sample considered in this study includes 1,743 non-Hispanic white students from 45 schools. The number of sampled students from each school is in the range of 30 to 50. The school membership can be identified by SCH_ID and each school is classified as either middle or high school (SCH_LEV).

According to socioecological models, patterns of adolescent tobacco smoking are best understood as embedded within social contexts. These social contexts can be either proximal in terms of individuals and peer groups or more distal in terms of schools and community. Socioecological models suggest that students within the same school often share common socioeconomic status (SES) and cultural characteristics that may cause different tobacco smoking patterns compared to students attending other schools. However, reflecting school (group) effect in an mgLCR is less likely to provide a meaningful summary because there are too many groups, that is, 45 schools in the nyts18 dataset. In this case, npLCR would be more appropriate model to investigate group difference in terms of a small number of latent clusters of schools.

Selecting the number of latent classes: Prior to conducting npLCR, it is necessary to determine the number of level-1 latent classes. Similar to the previous example, the two-, three-, and four-class LCA models can be fitted and compared by the gofglca() function as follow:

```
data("nyts18")
f <- item(starts.with = "E") ~ 1
lca2 <- glca(f, data = nyts18, nclass = 2, seed = 1)
```

```
lca3 <- glca(f, data = nyts18, nclass = 3, seed = 1)
lca4 <- glca(f, data = nyts18, nclass = 4, seed = 1)
lca.gof <- gofglca(lca2, lca3, lca4, test = "boot", seed = 1)
```

The output in the object `lca.gof` (not shown here) indicates that the three-class and the four-class models are adequate in terms of absolute model fit (p-value = 0.28 and 0.78, respectively), but the three-class LCA provides the lowest values in information criteria. For the relative model fit, the two-class LCA is rejected (p-value = 0.00) on comparison with the three-class LCA using the bootstrap. However, the three-class model is not rejected (p-value = 0.10) on comparison with the four-class model. Thus, it seems to be reasonable to select the three-class LCA model for the `nyts18` dataset.

Selecting the number of latent clusters: Now, an npLCR can be implemented by adding `SCH_ID` as group variable and specifying the number of latent clusters (i.e., level-2 latent classes) using the `ncluster` argument in the `glca()` function. The two-, three-, and four-cluster npLCR with three latent classes are fitted and compared using the following commands:

```
nplca2 <- glca(f, group = SCH_ID, data = nyts18, nclass = 3, ncluster = 2,
               seed = 1)
nplca3 <- glca(f, group = SCH_ID, data = nyts18, nclass = 3, ncluster = 3,
               seed = 1)
nplca4 <- glca(f, group = SCH_ID, data = nyts18, nclass = 3, ncluster = 4,
               seed = 1)
gofglca(lca3, nplca2, nplca3, nplca4, test = "boot", seed = 1)

#> Model 1: item(starts.with = "E") ~ 1
#>          nclass: 3
#> Model 2: item(starts.with = "E") ~ 1
#>          Group: SCH_ID, nclass: 3, ncluster: 2
#> Model 3: item(starts.with = "E") ~ 1
#>          Group: SCH_ID, nclass: 3, ncluster: 3
#> Model 4: item(starts.with = "E") ~ 1
#>          Group: SCH_ID, nclass: 3, ncluster: 4
#>
#> Goodness of Fit Table :
#>     logLik     AIC    CAIC     BIC entropy Res.Df    Gsq Boot p-value
#> 1 -2086.86 4207.71 4317.50 4300.50    0.87     14  30.37         0.20
#> 2 -1955.49 3950.97 4080.14 4060.14    0.84   1419 765.73         0.08
#> 3 -1938.73 3923.46 4072.00 4049.00    0.84   1416 732.22         0.22
#> 4 -1938.30 3928.60 4096.51 4070.51    0.84   1413 731.35         0.24
#>
#> Analysis of Deviance Table :
#>   npar   logLik Df Deviance Boot p-value
#> 1   17 -2086.86
#> 2   20 -1955.49  3   262.74         0.00
#> 3   23 -1938.73  3    33.51         0.00
#> 4   26 -1938.30  3     0.87         0.42
```

The goodness-of-fit table shows that the three-cluster model (`Model 3`) has the smallest values in information criteria among others, and the bootstrap p-value (= 0.22) indicates that this

model is appropriate for the data. The analysis-of-deviance table shows that group effect is significant as `Model 1` has better fit than `Model 2` (p-value = 0.0). In addition, the three-cluster model (`Model 3`) has better fit than the two-cluster model (`Model 2`, p-value = 0.00), but there is insignificant difference between the three- and four-cluster models (`Model 3` and `Model 4`) in the model fit (p-value = 0.42). Therefore, we can conclude that the three-cluster model is most appropriate among others.

Testing the equality of coefficients for level-1 covariates across groups: Covariates can be incorporated into npLCR using the `glca()` function; not only the subject-specific (i.e., level-1) covariates (e.g., `SEX`) but also the group-specific (i.e., level-2) covariates (e.g., `SCH_LEV`). As shown in the previous example, subject-specific covariates are constrained to be equal across latent clusters by default (i.e., `coeff.inv = TRUE`). The chi-square LRT test for checking the equality of coefficients for level-1 covariate, `SEX` can be conducted using the `gofglca()` function as follows:

```
f.1 <- item(starts.with = "E") ~ SEX
nplcr1 <- glca(f.1, group = SCH_ID, data = nyts18, nclass = 3, ncluster = 3,
               seed = 1)
nplcr2 <- glca(f.1, group = SCH_ID, data = nyts18, nclass = 3, ncluster = 3,
               seed = 1, coeff.inv = FALSE)
nplcr.gof <- gofglca(nplca3, nplcr1, nplcr2, test = "chisq")
nplca.gof$dtable

#>   npar    logLik Df Deviance   Pr(>Chi)
#> 1   23 -1938.731 NA       NA         NA
#> 2   25 -1935.003  2    7.457 0.02402604
#> 3   29 -1931.103  4    7.799 0.09921874
```

Note that typing `nplca.gof$dtable` into the R console will return the analysis-of-deviance table. Considering the p-values given in the table above, the equality of `SEX` effect can be assumed. Therefore, we conclude that the model specified in `nplcr1` should be selected for the `nyts18` dataset.

Incorporating the level-2 covariates: In npLCR, the meaning of the latent cluster (i.e., level-2 class) is interpreted by the prevalence of latent class (i.e., level-1 class) for each of cluster membership. When level-2 covariates are incorporated into the model, this prevalence is designed to be affected by level-2 covariates as shown (5). In other words, as level-2 covariates significantly influence the prevalence of level-1 class, the meaning of latent cluster may change as the level of level-2 covariate changes. Therefore, the number of clusters could be reduced when level-2 covariates are included compared to the number of clusters of npLCR without any level-2 covariate. As we selected the three-cluster model without any level-2 covariate for the `nyts18` dataset, the fit of the three-cluster model should be compared with the fit of the two-cluster model when `SCH_LEV` is incorporated as a level-2 covariate.Note that different seed values (`seed = 3` and `seed = 6`) are used in the `glca()` function below in order to produce the class order displayed in Figure 2.

```
f.2 <- item(starts.with = "E") ~ SEX + SCH_LEV
nplcr3 <- glca(f.2, group = SCH_ID, data = nyts18, nclass = 3, ncluster = 2,
               seed = 3)
```

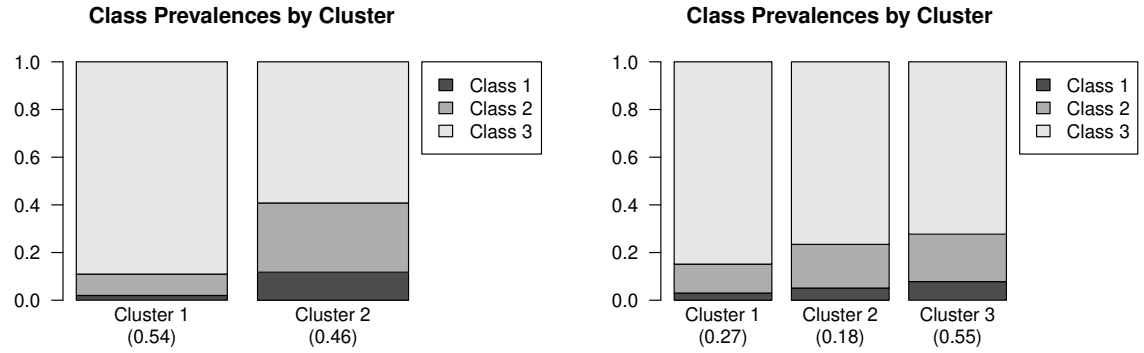

Figure 2: Estimated class prevalences for each cluster membership from the two-cluster and the three-cluster npLCR models specified in `nplcr3` and `nplcr4` for the `nyts18` dataset.

```
nplcr4 <- glca(f.2, group = SCH_ID, data = nyts18, nclass = 3, ncluster = 3,
               seed = 6)
gofglca(nplcr3, nplcr4, test = "boot", seed = 1)

#> Model 1: item(starts.with = "E") ~ SEX + SCH_LEV
#>          Group: SCH_ID, nclass: 3, ncluster: 2,  coef.inv: TRUE
#> Model 2: item(starts.with = "E") ~ SEX + SCH_LEV
#>          Group: SCH_ID, nclass: 3, ncluster: 3, coef.inv: TRUE
#>
#> Goodness of Fit Table :
#>     logLik     AIC    CAIC     BIC entropy Res.Df     Gsq Boot p-value
#> 1 -1919.94 3887.87 4042.87 4018.87    0.83   1709 1052.54         0.12
#> 2 -1916.30 3886.60 4060.97 4033.97    0.84   1706 1045.26         0.06
#>
#> Analysis of Deviance Table :
#>   npar   logLik Df Deviance Boot p-value
#> 1   24 -1919.94
#> 2   27 -1916.30  3     7.28            0
```

The analysis-of-deviance table from the `gofglca()` function shows that three clusters are required even when a level-2 covariate `SCH_LEV` is incorporated (p-value = 0.00). However, information criteria and p-values for absolute model fit may let us reach a different conclusion: the two-cluster model provides smaller values in some information criteria and the p-value (0.12) indicates that the model is appropriate for the dataset. Without strong prior beliefs, the number of latent clusters should be chosen to strike a balance between parsimony, fit, and interpretability. Two bar graphs in Figure 2 display the estimated class prevalences for each cluster membership from the models specified in `nplcr3` and `nplcr4` using the `plot()` function, respectively. The values in parentheses printed in the x-axis are the estimated cluster prevalence. The bar graph in the right, which is generated from the object `nplcr4`, shows that there is not much difference in class prevalence among these three clusters, compared to the bar graph in the left generated from the two-cluster model specified in `nplcr3`. In other words, the three-cluster model is not substantively meaningful, and therefore, we may conclude that the two-cluster model specified in `nplcr3` is more adequate to describe cluster (group) variation in the latent class distribution.

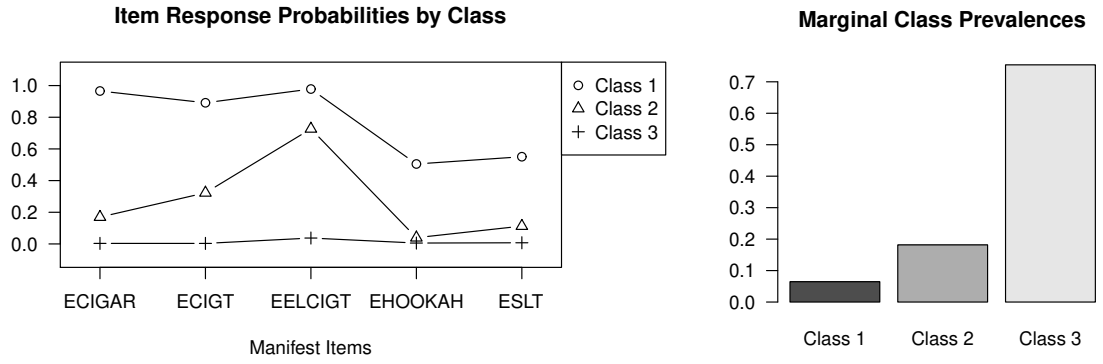

Figure 3: Estimated parameters of three-class and two-cluster npLCR model specified in `nplcr3` for the `nyts18` dataset.

Summarizing the results from the selected model: The estimated parameters from the three-class and two-cluster npLCR model specified in `nplcr3` are shown in Figure 3 using the `plot()` function. Based on the line graph in Figure 3, we deduce that Class 1 represents the poly-user group; Class 2 is the electronic cigarette user group; and Class 3 is the non-smoking group. We already argued that the two latent clusters were clearly distinguished by their class prevalence using the left bar graph in Figure 2. According to this stacked bar graph, the probability of engagement in a certain latent class is significantly different between these two school clusters. About 12% and 29% of students in Cluster 2 belong to Class 1 (poly-user group) and Class 2 (electronic cigarette user group), whereas only 2% and 9% of them belong to Class 1 and Class 2, respectively. The difference in class prevalences by cluster indicates that students who attend a school classified as Cluster 2 are more likely to be smokers relative to a similar student attending a school classified as Cluster 1. The full output of the `glca()` function for the selected model can be displayed by the `summary()` function as follows:

```
summary(nplcr3)

#> Call:
#> glca(formula = f.2, group = SCH_ID, data = nyts18, nclass = 3,
#>     ncluster = 2, seed = 3)
#>
#> Manifest items : ECIGAR ECIGT EELCIGT EHOOKAH ESLT
#> Grouping variable : SCH_ID
#> Covariates (Level 1) : SEX
#> Covariates (Level 2) : SCH_LEV
#>
#> Categories for manifest items :
#>          Y = 1 Y = 2
#> ECIGAR     Yes    No
#> ECIGT      Yes    No
#> EELCIGT    Yes    No
#> EHOOKAH    Yes    No
#> ESLT       Yes    No
#>
```

```
#> Model : Nonparametric multilevel latent class analysis
#>
#> Number of latent classes : 3
#> Number of latent clusters : 2
#> Number of groups : 45
#> Number of observations : 1734
#> Number of parameters : 24
#>
#> log-likelihood : -1919.937
#>      G-squared : 1052.541
#>            AIC : 3887.874
#>            BIC : 4018.87
#>
#> Marginal prevalences for latent classes :
#> Class 1 Class 2 Class 3
#> 0.06455 0.18159 0.75385
#>
#> Marginal prevalences for latent clusters :
#> Cluster 1 Cluster 2
#>   0.54083   0.45917
#>
#> Class prevalences by cluster :
#>           Class 1 Class 2 Class 3
#> Cluster 1  0.0199 0.08978 0.89032
#> Cluster 2  0.1175 0.29044 0.59206
#>
#> Logistic regression coefficients (level 1) :
#> Cluster 1
#>             Class 1/3 Class 2/3
#> (Intercept)   -2.6345   -1.0654
#> SEXFemale      0.6307    0.1159
#>
#> Cluster 2
#>             Class 1/3 Class 2/3
#> (Intercept)   -0.8997    0.1671
#> SEXFemale      0.6307    0.1159
#>
#> Logistic regression coefficients (level 2) :
#>                       Class 1/3 Class 2/3
#> SCH_LEVMiddle School   -2.5656   -1.9522
#>
#> Item-response probabilities (Y = 1) :
#>         ECIGAR  ECIGT EELCIGT EHOOKAH   ESLT
#> Class 1 0.9657 0.8914  0.9782  0.5049 0.5507
#> Class 2 0.1696 0.3227  0.7266  0.0394 0.1127
#> Class 3 0.0034 0.0033  0.0372  0.0056 0.0074
#>
#> Item-response probabilities (Y = 2) :
```

```
#>          ECIGAR  ECIGT EELCIGT EHOOKAH   ESLT
#> Class 1 0.0343 0.1086  0.0218  0.4951 0.4493
#> Class 2 0.8304 0.6773  0.2734  0.9606 0.8873
#> Class 3 0.9966 0.9967  0.9628  0.9944 0.9926
```

As shown in the previous example, the result of Wald test for each of estimated odds ratios and coefficients can be obtained by typing `coef(nplcr3)` into the R console (results not shown here). The Wald test shows that females are more likely to belong to Class 1 than Class 3, indicating that females are at a higher risk than their male counterparts. For level-2 covariate SCH_LEV, middle schools are less likely to belong to Class 1 or 2 than Class 3, that is, middle-school students tend to smoke less than high-school students.

Imputing the cluster membership: Researchers often want to explore the effects of level-2 covariates on the imputed latent cluster membership. Note that we selected the two-cluster model specified in `nplcr3`. We can easily impute the latent cluster membership for 45 schools using the posterior probabilities from the model specified in `nplcr3` and fit the logistic regression with SCH_LEV as a covariate. The posterior probabilities for latent cluster can be accessed by `nplcr3$posterior$cluster`. The following codes generate the imputed latent cluster membership for each school and save the cluster membership in `ndata` with level-2 covariate SCH_LEV.

```
tmp1 <- unique(nyts18[c("SCH_ID", "SCH_LEV")])
tmp2 <- nplcr3$posterior$cluster
tmp3 <- data.frame(SCH_ID = rownames(tmp2),
                   Cluster = factor(apply(tmp2, 1, which.max)))
ndata <- merge(tmp1, tmp3)
head(ndata)
```

```
#>   SCH_ID       SCH_LEV Cluster
#> 1 00b895 Middle School       1
#> 2 066e6c   High School       2
#> 3 0690d1   High School       2
#> 4 0fc94b   High School       2
#> 5 12d5ad Middle School       1
#> 6 16082c Middle School       1
```

Using the logistic regression, the effect of level-2 covariate SCH_LEV on the latent cluster membership can be estimated as following:

```
fit <- glm(Cluster ~ SCH_LEV, family = binomial, data = ndata)
```

Summary

The `glca` package mainly focuses on exploring group differences in latent structure using multiple-group latent class models. We have demonstrated the functionality of the `glca` package by fitting two different types of latent class models; mgLCR and npLCR with level-1 and/or level-2 covariates. It should be noted that the `glca` package can deal with the manifest item that has more than two outcome categories, and it requires less computing time compared to

other R packages for LCA. The glca package allows users to conduct model comparisons using bootstrap samples and visualize model results by producing a range of plots for describing the classes. It also produces standard errors for the estimated model parameters and posterior probabilities of latent class and latent cluster membership.

Although the glca package provides various tools for latent class models with multiple groups, a further extension is required. For instance, it cannot provide the expected frequencies of data predicted by the fitted model. Also, there is more sophisticated technique to prevent local maxima problem such as using other deterministic or simulated annealing methods (Ueda & Nakano, 1998; Chang & Chung, 2013). Another extension is to incorporate the parametric approach into npLCR. A further extension to the glca package is to enable a latent group variable, which is constructed by other latent class models, to incorporate the group effect.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) (2021R1A2C1003486 to Chung and 2020R1F1A1A01055067 to Jeon).

References

Agresti, A. (2013). *Categorical data analysis* (Third Edition ed.). Wiley & Sons.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716-723. Retrieved from https://doi.org/10.1109/TAC.1974.1100705

Bandeen-Roche, K., Miglioretti, D. L., Zeger, S. L., & Rathouz, P. J. (1997). Latent variable regression for multiple discrete outcomes. *Journal of the American Statistical Association*, *92*(440), 1375–1386. Retrieved from https://doi.org/10.1080/01621459.1997.10473658

Baum, L. E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, *41*(1), 164–171. Retrieved from https://doi.org/10.1214/aoms/1177697196

Beath, K. J. (2017). randomLCA: An R package for latent class with random effects analysis. *Journal of Statistical Software*, *81*(13), 1–25. Retrieved from https://doi.org/10.18637/jss.v081.i13

Bozdogan, H. (1987, 02). Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. *Psychometrika*, *52*, 345-370. Retrieved from https://doi.org/10.1007/BF02294361

Chang, H.-C., & Chung, H. (2013). Dealing with multiple local modalities in latent class profile analysis. *Computational Statistics and Data Analysis*, *68*, 296–310. Retrieved from https://doi.org/10.1016/j.csda.2013.07.016

Clogg, C. C., & Goodman, L. A. (1984). Latent structure analysis of a set of multidimensional contingency tables. *Journal of the American Statistical Association*, *79*, 762–771. Retrieved from https://doi.org/10.2307/2288706

Clogg, C. C., & Goodman, L. A. (1985). Simultaneous latent structure analysis in several groups. *Sociological Methodology*, *15*, 81–110. Retrieved from https://doi.org/10.2307/270847

Collins, L. M., & Lanza, S. T. (2009). *Latent class and latent transition analysis: With applications in the social, behavioral, and health sciences* (Vol. 718). John Wiley & Sons.

Dayton, C. M., & Macready, G. B. (1988). Concomitant-variable latent-class models. *Journal of the American Statistical Association*, *83*, 173–178. Retrieved from https://doi.org/10.2307/2288938

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, *39*, 1–38. Retrieved from https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

Goodman, L. A. (1974). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*, *61*, 215–231. Retrieved from https://doi.org/10.1093/biomet/61.2.215

Juang, B. H., & Rabiner, L. R. (1991). Hidden markov models for speech recognition. *Technometrics*, *33*(3), 251-272. Retrieved from https://doi.org/10.1080/00401706.1991.10484833

Kim, Y., & Chung, H. (2021). glca: An R package for multiple-group latent class analysis [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=glca (R package version 1.3.1)

Langeheine, R., Pannekoek, J., & van de Pol, F. (1996). Bootstrapping goodness-of-fit measures in categorical data analysis. *Sociological Methods & Research*, *24*, 492-516. Retrieved from https://doi.org/10.1177/0049124196024004004

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

Linzer, D. A., & Lewis, J. B. (2011). poLCA: An R package for polytomous variable latent class analysis. *Journal of Statistical Software*, *42*(10), 1–29. Retrieved from https://doi.org/10.18637/jss.v042.i10

Little, R. J. A., & Rubin, D. B. (2019). *Statistical analysis with missing data* (Third ed.). New York: Wiley. Retrieved from http://doi.org/10.1002/9781119013563

McCutcheon, A. L. (1987). Sexual morality, pro-life values, and attitudes toward abortion: A simultaneous latent structure analysis for 1978-1983. *Sociological Methods & Research*, *16*(2), 256–275. Retrieved from https://doi.org/10.1177/0049124187016002003

Mclachlan, G., & Krishnan, T. (2007). *The em algorithm and extensions*. Wiley. Retrieved from https://doi.org/10.1002/9780470191613

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C.-C., & Lin, C.-C. (2020). e1071: Misc functions of the department of statistics, probability theory group (formerly: E1071), tu wien [Computer software manual]. Retrieved from https://CRAN.R-project.org/package=e1071 (R package version 1.7-4)

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. New York, NY: Chapman & Hall/CRC.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*(2), 461–464. Retrieved from https://doi.org/10.1214/aos/1176344136

Smith, T. W., Marsden, P., Hout, M., & Kim, J. (2010). *General social surveys*. Chicago, IL: NORC at the University of Chicago NORC at the University of Chicago.

Ueda, N., & Nakano, R. (1998). Deterministic annealing em algorithm. *Neural Networks*, *11*, 271-282. Retrieved from https://doi.org/10.1016/S0893-6080(97)00133-0

van Buuren, S., Brand, J. P., Groothuis-Oudshoorn, C. G., & Rubin, D. B. (2006). Fully conditional specification in multivariate imputation. *Journal of Statistical Computa-*

tion and Simulation, *76*(12), 1049–1064. Retrieved from https://doi.org/10.1080/10629360600810434

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, *45*(3), 1-67. Retrieved from https://doi.org/10.18637/jss.v045.i03

Vermunt, J. K. (2003). Multilevel latent class models. *Sociological Methodology*, *33*(1), 213-239. Retrieved from https://doi.org/10.1111/j.0081-1750.2003.t01-1-00131.x

White, A., & Murphy, T. B. (2014). BayesLCA: An R package for bayesian latent class analysis. *Journal of Statistical Software*, *61*(13), 1–28. Retrieved from https://doi.org/10.18637/jss.v061.i13

Appendix

Score functions and Jacobian matrices

Fixed-effect latent class analysis: Let $\boldsymbol{\alpha}_g$ and $\boldsymbol{\beta}_g$ be vectorized parameters containing all coefficients of $\alpha_{c|g}$ and $\boldsymbol{\beta}_{c|g}$ given in (2) for $c = 1, \ldots, C-1$ and $g = 1, \ldots, G$, respectively. Further, let $\mathbf{s}(\mathbf{Y}_g; \boldsymbol{\alpha}_g)$ and $\mathbf{s}(\mathbf{Y}_g; \boldsymbol{\beta}_g)$ denote score functions of $\boldsymbol{\alpha}_g$ and $\boldsymbol{\beta}_g$, respectively. Then, the element of $\mathbf{s}(\mathbf{Y}_g; \boldsymbol{\alpha}_g)$ and the $p \times 1$ sub-vector of $\mathbf{s}(\mathbf{Y}_g; \boldsymbol{\beta}_g)$ are obtained as

$$\frac{\partial \log \mathcal{L}_g}{\partial \alpha_{c|g}} = \sum_{i=1}^{n_g} \left[\theta_{ig(c)} - \gamma_{c|g}(\mathbf{x}_{ig})\right] \quad \text{and} \quad \frac{\partial \log \mathcal{L}_g}{\partial \boldsymbol{\beta}_{c|g}} = \sum_{i=1}^{n_g} \left[\mathbf{x}_{ig}\left(\theta_{ig(c)} - \gamma_{c|g}(\mathbf{x}_{ig})\right)\right] \tag{10}$$

for $c = 1, \ldots, C-1$ and $g = 1, \ldots, G$, respectively. Note that $\mathcal{L}_g$ in (10) is the product of observed-data likelihoods given in (1) for all observations in the gth group (i.e., $\mathcal{L}_g = \prod_{i=1}^{n_g} \mathcal{L}_{ig}$).

Let $\boldsymbol{\rho}_g$ denote vectorized item-response probabilities containing all $\boldsymbol{\rho}_{m|cg} = (\rho_{m1|cg}, \ldots, \rho_{mr_m|cg})^\top$ for $m = 1, \ldots, M$, $c = 1, \ldots, C$ and $g = 1, \ldots, G$. Each of ρ-parameters in $\boldsymbol{\rho}_{m|cg}$ is re-parameterized by the baseline logit function $\pi_{mk|cg} = \ln(\rho_{mk|cg}/\rho_{mr_m|cg})$ for $k = 1, \ldots, r_m - 1$, and let $\boldsymbol{\pi}_g$ denote vectorized free parameters containing all $\boldsymbol{\pi}_{m|cg} = (\pi_{m1|cg}, \ldots, \pi_{mr_m-1|cg})^\top$ for $m = 1, \ldots, M$, $c = 1, \ldots, C$ and $g = 1, \ldots, G$. Further, let $\mathbf{s}(\mathbf{Y}_g; \boldsymbol{\pi}_g)$ denote score function of all free parameters $\boldsymbol{\pi}_g$. Then, the element of score function $\mathbf{s}(\mathbf{Y}_g; \boldsymbol{\pi}_g)$ is obtained as

$$\frac{\partial \log \mathcal{L}_g}{\partial \pi_{mk|cg}} = \sum_{i=1}^{n_g} \left[\theta_{ig(c)}\left(I(y_{igm} = k) - \rho_{mk|cg}\right)\right] \tag{11}$$

for $k = 1, \ldots, r_m - 1$, $m = 1, \ldots, M$, $c = 1, \ldots, C$, and $g = 1, \ldots, G$.

Let $\boldsymbol{\Psi}$ denote vector for all free parameters $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1^\top, \ldots, \boldsymbol{\alpha}_G^\top)^\top$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \ldots, \boldsymbol{\beta}_G^\top)^\top$, and $\boldsymbol{\pi} = (\boldsymbol{\pi}_1^\top, \ldots, \boldsymbol{\pi}_G^\top)^\top$, and let $q(\boldsymbol{\Psi})$ be the function to transform back to the original parameters of mgLCR. Then, the Jacobian matrix for the function $q(\boldsymbol{\Psi})$ is

$$J_q(\boldsymbol{\Psi}) = \begin{bmatrix} J_q(\boldsymbol{\alpha}, \boldsymbol{\beta}) & \mathbf{0} \\ \mathbf{0} & J_q(\boldsymbol{\pi}) \end{bmatrix}, \tag{12}$$

where $J_q(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is an identity matrix of size equal to the number of regression coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The sub-matrix of $J_q(\boldsymbol{\pi})$ given in (12) can be specified by $\partial \boldsymbol{\rho}_{m|cg}/\partial \boldsymbol{\pi}_{m'|c'g'}$, which is the matrix of size $r_m \times (r_{m'} - 1)$ for $m, m' = 1, \ldots, M$; $c, c' = 1, \ldots, C$; and $g, g' = 1, \ldots, G$. The element of this sub-matrix in the kth row and the k'th column can be obtained as

$$\frac{\partial \rho_{mk|cg}}{\partial \pi_{m'k'|c'g'}} = I(g = g')I(c = c')I(m = m')\rho_{mk|cg}\left(I(k = k') - \rho_{m'k'|c'g'}\right) \tag{13}$$

for $k = 1, \ldots, r_m$ and $k' = 1, \ldots, r_m - 1$.

Random-effect latent class analysis: The prevalences for latent clusters, $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_W)^\top$ are re-parametrized by the baseline logit function $\zeta_w = \ln(\delta_w/\delta_W)$ for $w = 1, \ldots, W-1$. Let $\mathbf{s}(\mathbf{Y}_g; \boldsymbol{\zeta})$ denote score function of $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_{W-1})^\top$ for the gth group. Then, the wth element of $\mathbf{s}(\mathbf{Y}_g; \boldsymbol{\zeta})$ is obtained as

$$\frac{\partial \log \mathcal{L}_g}{\partial \delta_w} = \theta_{g(w)} - \delta_w$$

for $w = 1, \ldots, W-1$, where $\mathcal{L}_g$ is the observed-data likelihood of npLCR for the gth group given in (4).

Let $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}_1$ be vectorized parameters containing all coefficients of level-1 covariates, $\alpha_{c|w}$ and $\boldsymbol{\beta}_{1c|w}$ given in (5) for $c = 1, \ldots, C-1$ and $w = 1, \ldots, W$, respectively. Further, let $\mathbf{s}(\mathbf{Y}_g; \boldsymbol{\alpha})$ and $\mathbf{s}(\mathbf{Y}_g; \boldsymbol{\beta}_1)$ denote score functions of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}_1$ for the gth group, respectively. Then, the element of $\mathbf{s}(\mathbf{Y}_g; \boldsymbol{\alpha})$ and the $p \times 1$ sub-vector of $\mathbf{s}(\mathbf{Y}_g; \boldsymbol{\beta}_1)$ are obtained as

$$\begin{aligned}
\frac{\partial \log \mathcal{L}_g}{\partial \alpha_{c|w}} &= \sum_{i=1}^{n_g} \left[\theta_{ig(c)} - \theta_{g(w)} \gamma_{c|w}(\mathbf{x}_{ig}, \mathbf{z}_g)\right] \quad \text{and} \\
\frac{\partial \log \mathcal{L}_g}{\partial \boldsymbol{\beta}_{1c|w}} &= \sum_{i=1}^{n_g} \left[\mathbf{x}_{ig} \left(\theta_{ig(c)} - \theta_{g(w)} \gamma_{c|w}(\mathbf{x}_{ig}, \mathbf{z}_g)\right)\right]
\end{aligned}$$

for $c = 1, \ldots, C-1$, $w = 1, \ldots, W$, and $g = 1, \ldots, G$, respectively. Let $\boldsymbol{\beta}_2$ denote vectorized parameters containing all coefficients of level-2 covariates, $\boldsymbol{\beta}_{2c}$ given in (5) for $c = 1, \ldots, C-1$. The $q \times 1$ sub-vector of score function for $\boldsymbol{\beta}_2$, $\mathbf{s}(\mathbf{Y}_g; \boldsymbol{\beta}_2)$ can be obtained as

$$\frac{\partial \log \mathcal{L}_g}{\partial \boldsymbol{\beta}_{2c}} = \mathbf{z}_g \sum_{i=1}^{n_g} \left[\theta_{ig(c)} - \sum_{w=1}^{W} \theta_{g(w)} \gamma_{c|w}(\mathbf{x}_{ig}, \mathbf{z}_g)\right]$$

for $c = 1, \ldots, C-1$ and $g = 1, \ldots, G$. The score functions for the free parameters of item-response probabilities are identical to mgLCR given in (11).

Let $\boldsymbol{\Psi}$ denote vector for all free parameters $\boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2$, and $\boldsymbol{\pi}$, and let $q(\boldsymbol{\Psi})$ be the function to transform back to the original parameters of npLCR. Then, the Jacobian matrix for the function $q(\boldsymbol{\Psi})$ is

$$J_q(\boldsymbol{\Psi}) = \begin{bmatrix} J_g(\boldsymbol{\zeta}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & J_q(\boldsymbol{\alpha}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & J_q(\boldsymbol{\pi}) \end{bmatrix},$$

where $J(\boldsymbol{\alpha}, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$ is an identity matrix of size equal to the number of regression coefficients given in (5). The sub-matrix of $J_q(\boldsymbol{\pi})$ are identical to those given in (13), an the elements of $J_q(\boldsymbol{\zeta})$ can be obtained by

$$\frac{\partial \delta_w}{\partial \zeta_{w'}} = \delta_w \left(I(w = w') - \delta_{w'}\right)$$

for $w = 1, \ldots, W$ and $w' = 1, \ldots, W-1$.