



randomLCA: An R Package for Latent Class with Random Effects Analysis

Ken J. Beath
Macquarie University

Abstract

Latent class is a method for classifying subjects, originally based on binary outcome data but now extended to other data types. A major difficulty with the use of latent class models is the presence of heterogeneity of the outcome probabilities within the true classes, which violates the assumption of conditional independence, and will require a large number of classes to model the association in the data resulting in difficulties in interpretation. A solution is to include a normally distributed subject level random effect in the model so that the outcomes are now conditionally independent given both the class and random effect. A further extension is to incorporate an additional period level random effect when subjects are observed over time. The use of the **randomLCA** R package is demonstrated on three latent class examples: classification of subjects based on myocardial infarction symptoms, a diagnostic testing approach to comparing dentists in the diagnosis of dental caries and classification of infants based on respiratory and allergy symptoms over time.

Keywords: latent class, random effects, diagnostic test, sensitivity, specificity, R.

1. Introduction

Latent class models (Lazarsfeld and Henry 1968) are a method originally developed for sociology where they are used to identify clusters or sub-groups of subjects, based on multivariate binary observations, and as such are a form of finite mixture model. Their application has been further expanded into many areas, such as psychology, market research and medicine. Diagnostic classification using latent class methods has been applied in a number of areas, with early applications by Golden (1982) to dementia, Young (1982) to develop diagnostic criteria for schizophrenia and Rindskopf and Rindskopf (1986) for myocardial infarction. An advantage of using latent class analysis over other classification methods is that the classification is model based, allowing use of model selection techniques to determine which classifi-

cation scheme is most appropriate. This compares with classifications developed from simple observation, that may give undue weight to one or more symptoms or outcomes. An example of the problems with this type of analysis is demonstrated by Nyholt, Gillespie, Heath, Merikangas, Duffy, and Martin (2004) who used latent class analysis of headache symptoms to show that classification of migraine with and without aura as separate diagnoses is not supported. While latent class methods have been extended to any outcomes with a variety of distributions, binary is the most commonly used.

The assumption of latent class models is conditional or local independence, where the outcomes are independent conditional on the latent class. This assumes that subjects within a class are homogeneous. Where this does not apply, that is the true classes are heterogeneous, a consequence of this will be to increase the number of latent classes required in attempting to explain the heterogeneity, with possible consequent difficulty in interpretation. A solution is to incorporate random effects so that the outcomes are independent conditional on the latent class and random effect or effects.

There are a number of packages capable of fitting latent class models in R (R Core Team 2017). Two of these solely for fitting of latent class models are **poLCA** (Linzer and Lewis 2011) and **BayesLCA** (White and Murphy 2014). **BayesLCA** is particularly designed to perform Bayesian analyses, but also offers the choice of the EM algorithm and variational Bayes, but has limited facilities for producing plots and summaries. **poLCA** is a more fully featured package which allows for polytomous outcomes and latent class regression, which are not available in **randomLCA**. The advantage of **randomLCA** over the other packages is that it will fit both standard latent class models and those incorporating random effects. This is important for use with diagnostic tests, as it allows for the variation of the test response between subjects, but may be also used to model heterogeneity in other applications, for example see Muthén (2006). Commercial software packages that also allow latent class with random effects are **Mplus** (Muthén and Muthén 2015) and **Latent GOLD Syntax Module** (Vermunt and Magidson 2013), both of which require the model to be defined using a symbolic language.

The purpose of this paper is to describe the **randomLCA** package (Beath 2017). Package **randomLCA** is available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/package=randomLCA>. The remainder of the paper is organized as follows. Section 2 describes the models, starting with standard latent class and then continuing with the random effect extensions, including references allowing for further investigation. Section 3 describes three examples with explanation of how the features of the package may be used. Section 4 summarizes the capabilities of the package and describes some areas in which the package could be extended.

2. Models

2.1. Latent class model

The basis of latent class analysis is that each subject is assumed to belong to one of a finite number of classes, with each class described by a set of parameters that define the distribution of outcomes or manifest variables for a subject, and is a form of finite mixture model (McLachlan and Peel 2000). Generally, the number of classes is unknown and must be

determined from the data. For binary outcomes, the model is

$$P(y_{i1}, y_{i2}, \dots, y_{ik} | c_i = c) = \prod_{j=1}^k \pi_{cj}^{y_{ij}} (1 - \pi_{cj})^{1-y_{ij}},$$

where y_{ij} is the j th binary outcome for subject i , π_{cj} is the probability of the j th outcome being equal to 1 for a subject in class c , k is the number of outcomes and c_i is the class corresponding to the i th subject. The marginal probability, obtained by summing over the classes, for each subject is

$$P(y_{i1}, y_{i2}, \dots, y_{ik}) = \sum_{c=1}^C \eta_c \prod_{j=1}^k \pi_{cj}^{y_{ij}} (1 - \pi_{cj})^{1-y_{ij}},$$

where η_c is the probability of a subject being in class c with $\sum_{c=1}^C \eta_c = 1$ and C is the number of classes. From this can be obtained the marginal likelihood.

A requirement for the estimates of the probabilities π_{cj} is that they be restricted to the interval zero to one, and that the η_c sum to one, something that did not always occur with the original methods used for analysis. A solution developed by [Formann \(1978, 1982\)](#) is to use a logistic (or alternatively probit) transformation, allowing unconstrained estimation of parameters but correctly restricting the probabilities to between zero and one. This can be obtained for the logistic using the following relations $\pi_{cj} = e^{a_{cj}} / (1 + e^{a_{cj}})$ and $\eta_c = e^{\theta_c} / \sum_{\ell=1}^C e^{\theta_\ell}$. Hence we estimate the a_{cj} and θ_c , rather than π_{cj} and η_c . Similar equations apply for the probit transformation.

When used as a classification algorithm the model does not simply return the most likely class for each subject but returns a probability of class membership, based on the observed outcomes. The posterior probability for a set of given observed outcomes can be obtained from Bayes theorem. Given the observed outcomes $y_{i1}, y_{i2}, \dots, y_{ik}$ then the probability that the subject is in class d is:

$$\begin{aligned} P(d | y_{i1}, y_{i2}, \dots, y_{ik}) &= \frac{\eta_d P(y_{i1}, y_{i2}, \dots, y_{ik} | d)}{\sum_{c=1}^C \eta_c P(y_{i1}, y_{i2}, \dots, y_{ik} | c)} \\ &= \frac{\eta_d \prod_{j=1}^k \pi_{dj}^{y_{ij}} (1 - \pi_{dj})^{1-y_{ij}}}{\sum_{c=1}^C \eta_c \prod_{j=1}^k \pi_{cj}^{y_{ij}} (1 - \pi_{cj})^{1-y_{ij}}}. \end{aligned}$$

2.2. Latent class with random effect model

A major difficulty with latent class models is the requirement for local independence or equivalently homogeneity of the outcome probabilities within each class. When the classes are heterogeneous the assumption that the manifest outcomes are independent, conditional on the latent class, does not apply. [Uebersax \(1999\)](#) describes the problems associated with conditional dependence as “to add spurious latent classes that are not truly present at the taxonic level” and [Vacek \(1985\)](#) showed that ignoring the conditional dependence produced biased estimates in the context of diagnostic testing. [Pickles and Angold \(2003, p. 530\)](#) discuss the classification of diseases in psychology as categories or by severity, arguing that “most forms of psychopathology (indeed, most forms of pathology of any sort) manifest both

continuous and discontinuous relationships with other phenomena”, and provide examples of where this may occur.

A solution to the problem of heterogeneity was developed by [Qu, Tan, and Kutner \(1996\)](#) combining a latent class model with a random effect to explain the heterogeneity. The probabilities are transformed to the probit scale and a normally distributed random effect added for each subject, before transforming back to probabilities. An alternative to the probit scale is the logit scale. A model for latent class incorporating a random effect $\lambda \sim N(0, 1)$ is:

$$P(y_{i1}, y_{i2}, \dots, y_{ik} | c_i = c, \lambda_i = \lambda) = \prod_{j=1}^k \pi_{icj}^{y_{ij}} (1 - \pi_{icj})^{1-y_{ij}},$$

where, either, for a probit scaling of the random effect

$$\pi_{icj} = \Phi^{-1}(a_{cj} + b_{cj}\lambda_i)$$

or, for a logistic scaling

$$\pi_{icj} = \frac{\exp(a_{cj} + b_{cj}\lambda_i)}{1 + \exp(a_{cj} + b_{cj}\lambda_i)},$$

and a_{cj} determines the conditional class probability for a value of zero for the random effect, and b_{cj} scales the random effect, and is usually known as the loading or discriminant. The loadings will generally be constrained to be equal between classes, and either the same loading for each outcome ($b_{cj} = b$) or independent loading for each outcome ($b_{cj} = b_j$) are used. The marginal likelihood is obtained by integrating over the random effect and summing over the latent classes, which is then maximized to obtain the parameter estimates. Posterior class probabilities can be obtained as for the standard latent class using Bayes theorem.

2.3. Two level latent class with random effect model

The previously described models for latent class can be considered to be for a single time point. Where the outcomes are observed at multiple time points then consideration must be made for the correlation between time points. A method described for this is the mixed latent Markov model ([Langeheine and Van de Pol 1990](#)). This assumes that the population is a mixture of latent Markov models which consists of a Markov chain which is a latent class at each time point. An alternative model is that of [Beath and Heller \(2009\)](#) which extends the model to two levels with an additional random effect modeling the correlation between outcomes at each time point. This has some similarities to the model by [Muthén and Shedden \(1999\)](#) for longitudinal normally distributed data.

We now define y_{ijt} as the j th binary outcome for subject i at time t , π_{cjt} is the probability of the j th outcome to be equal to 1 for a subject in class c at time t , k is the number of outcomes, T is the number of time points and c is the class corresponding to the i th subject. An additional random effect $\tau_t \sim N(0, 1)$ is incorporated to model the additional correlation between outcomes at a time point, and ℓ_c scales this to the appropriate variance.

$$P(y_{i11}, \dots, y_{ikT} | c_i = c, \lambda_i = \lambda, \tau_{i1} = \tau_1, \dots, \tau_{iT} = \tau_T) = \prod_{t=1}^T \prod_{j=1}^k \pi_{icjt}^{y_{ijt}} (1 - \pi_{icjt})^{1-y_{ijt}},$$

where

$$\pi_{icjt} = \Phi^{-1}(a_{cjt} + b_{cj}(\lambda_i + \ell_c \tau_{it}))$$

or

$$\pi_{icjt} = \frac{\exp(a_{cjt} + b_{cj}(\lambda_i + \ell_c \tau_{it}))}{1 + \exp(a_{cjt} + b_{cj}(\lambda_i + \ell_c \tau_{it}))}.$$

Again, we usually constrain $b_{cj} = b_j$ and $\ell_c = \ell$.

2.4. Model selection

An important aspect of using latent class and latent class with random effect models is the choice of the number of classes, and whether inclusion of a random effect is required. This presents two difficulties. The first involves a test for a parameter on the boundary of the parameter space. For the number of classes, this is the proportion in a class that is zero, and for the random effect, that the random effect variance is zero. As a consequence asymptotic likelihood theory does not hold so other methods must be used. One method is the bootstrapped likelihood ratio test as described by [McLachlan \(1987\)](#). However, a second difficulty is that when models may include a random effect then comparisons must be made between non-nested models. As an example we may wish to compare a latent class model with two or more classes to a latent class with random effect model with a single class. This is equivalent to an item response theory (IRT) model, either the Rasch model when the loadings are constant and a two parameter logistic otherwise ([Bartholomew, Steele, Moustaki, and Galbraith 2002](#)). This is an important choice as it determines if the underlying latent variable is categorical or continuous ([Muthén 2006](#)).

The usual method used is an information criterion ([Lin and Dayton 1997](#)) with the two main ones that are used being the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). [Nylund and Muthén \(2007\)](#) showed using simulation that BIC is superior to AIC for selection in latent class models, and this is the method most often used in applied publications. With BIC the penalty is greater than for AIC and dependent on the number of observations, so the BIC will select models with a smaller number of classes. An alternative version of AIC, AIC3 with a penalty of 3 was shown to have better performance with latent class models by [Dias \(2006\)](#). This will select models with the number of classes between that chosen by BIC and AIC inclusive. No evaluation of the criteria has been performed for latent class with random effect models, and it has seen little use in applied publications. An important point is that with any model selection it is desirable to make use of existing information. So for example, if it is already known that there are at least 2 classes then the model choice should be restricted to these. In the paper I have used BIC for model selection, but as no research has been performed on model selection for random effect latent class models the other information criteria are also provided.

2.5. Identifiability

A difficulty that is sometimes encountered in fitting latent class models is lack of identifiability, which occurs when the value of the maximum likelihood occurs for more than one unique set of parameter estimates. This implies that there is not a single global maximum. A minimum condition is that the number of parameters is less than the number of patterns, but this is not always sufficient ([McHugh 1956](#)). This prevents the fitting of even a 2 class latent class model with only two outcomes, as this requires 5 parameters with only 4 patterns. A 2 class model is just identifiable when fitted to data on 3 outcomes, as there are 7 parameters and 8 possible patterns, however if a random effect is included then 4 outcomes are required for

identifiability. A check is made by **randomLCA** to determine if the model is identifiable based on the number of parameters. As this is not always sufficient, special cases, for example the requirement of at least 5 outcomes to fit a 3 class latent class model, are also flagged as non-identified. A further check is provided by determining that the rank of the Hessian is not less than the number of estimated parameters (Skrondal and Rabe-Hesketh 2004, pp. 150–151), however this may also occur if a parameter is on the boundary of the parameter space.

2.6. Computational methods

The standard latent class models are fitted using an EM algorithm and then switching to a quasi-Newton method when near convergence. To increase the probability that the algorithm converges to the global maximum rather than a local maximum, the EM algorithm is performed a number of times with randomly generated starting values. The estimated parameters from the best fitting EM algorithm are then used as the starting values for the quasi-Newton method. For the latent class with random effect models it is necessary to integrate over the random effects for each subject, for which it is necessary to use an approximation. One of the methods that has been used for this is Gauss-Hermite quadrature, where the integral is approximated by a weighted sum of the function evaluated at defined points, which works well when the function is approximately standard normal. However, for random effect models the likelihood is not, but the standardized likelihood is. Therefore, Gauss-Hermite quadrature is applied to the standardized likelihood, known as adaptive Gauss-Hermite quadrature (Liu and Pierce 1994). This has the advantage over standard Gauss-Hermite quadrature that integration is only performed in the region of the mode, reducing the number of quadrature points required. For the two level random effect model, adaptive Gauss-Hermite quadrature is again used. An orthogonal transform is applied to reduce the integration to two one-dimensional integrations and the method of moments is used to determine the location of the modes, as described in Rabe-Hesketh, Skrondal, and Pickles (2005).

The random effect latent class models are fitted using a generalized EM algorithm (GEM) (Little and Rubin 2002, p. 173) with maximization using a quasi-Newton method. At each expectation step the location of the modes for the adaptive quadrature are recalculated, and the maximization step performed based on these locations. The maximization is not run to convergence at each step, but terminated after a number of quasi-Newton steps. For all models, the EM or GEM algorithm switches to a quasi-Newton for all parameters when close to convergence. For the random effect models starting values are obtained from the model without random effects, and a search performed over possible starting values for the random effect variance.

A difficulty with latent class models is the calculation of standard errors when the parameter estimates are near the boundary of the parameter space. There are several options, one of which is to use Bayesian maximum a posteriori (MAP) estimation (Galindo Garre and Vermunt 2006). This is a form of Bayesian estimation in that a prior probability is placed on the parameters, however the posterior distribution is then maximized similar to maximum likelihood estimation. This is equivalent to the penalized likelihood described by Firth (1993) in which a penalty is placed on extreme values of the logit or probit scale outcome probabilities. The posterior distribution will consequently have a mode, whereas the likelihood may not. In **randomLCA** the prior distribution used is the Dirichlet as in Galindo Garre and Vermunt (2006) but with a much smaller default penalty, where the penalty argument is equal to the

number of extra observations that are effectively added to each cell. A penalty of 0.01 is used, which reduces the risk of numerical problems, without greatly affecting the estimated probabilities. A sensible upper limit for the penalty is 0.5, which has been found to perform well by a number of authors, for example [Rubin and Schenker \(1987\)](#) in application to binary proportions, and may be used when necessary. This will also produce results similar to the **Latent GOLD** software default settings ([Vermunt and Magidson 2013](#)). Setting the penalty to zero will produce results identical to maximum likelihood.

3. Examples

3.1. Myocardial infarction example

This example demonstrates the fitting of data from [Rindskopf and Rindskopf \(1986\)](#), where latent class analysis is used to determine diagnostic classifications based on medical tests. Although this example is for medical data, the model is simply standard latent class so the methods can be applied to data from other areas, for example psychology and sociology. The maximum number of classes that can be fitted is limited to 2 due to identifiability, so we fit models for 1 and 2 classes, assuming that the outcome probabilities are homogeneous in each class.

The fitting function for **randomLCA** is the `randomLCA` function which fits both the standard and random effect models. The command, where only the `patterns` parameter is required, is:

```
randomLCA(patterns, freq = NULL, nclass = 2, calcSE = TRUE, notrials = 20,
  random = FALSE, byclass = FALSE, quadpoints = 21, constload = TRUE,
  blocksize = dim(patterns)[2], level2 = FALSE, probit = FALSE,
  level2size = blocksize, qniterations = 5, penalty = 0.01, EMtol = 1.0e-9,
  verbose = FALSE, seed = as.integer(runif(1, 0, .Machine$integer.max)))
```

For a standard latent class model the parameters of interest are:

patterns: Data frame or matrix of 0s and 1s defining the outcome patterns. This may be either raw data or summarized data with a frequency vector giving the corresponding frequency for each pattern. The patterns may also include missing values, with **randomLCA** using maximum likelihood to fit the models using all available data.

freq: Vector containing frequencies for each outcome pattern. Where this is missing it is created within the `randomLCA` function and used in the fitting algorithm, substantially reducing execution time.

nclass: Number of classes to be fitted.

notrials: The number of random starting values used. It is necessary to use different starting values to reduce the risk that the global maximum of the likelihood is not found. This should be increased in increments of ten until there is no decrease in the maximum log-likelihood. While this does not guarantee a maximum likelihood, the parameter estimates found will usually be close to those at the true global maximum.

penalty: Penalty applied to the likelihood for the outcome probabilities. See Section 2.6 for details.

The remaining arguments will be considered when discussing the relevant models or details may be obtained from the package documentation. Fitting the standard latent class models for one and two classes (note that a three class model is not identifiable) requires only the basic arguments:

```
R> myocardial.lca1 <- randomLCA(myocardial[, 1:4], freq = myocardial$freq,
+   nclass = 1)
R> myocardial.lca2 <- randomLCA(myocardial[, 1:4], freq = myocardial$freq,
+   nclass = 2)
```

The BIC values may be extracted from the fitted objects, and combined into a data frame:

```
R> myocardial.bic <- data.frame(classes = 1:2,
+   bic = c(BIC(myocardial.lca1), BIC(myocardial.lca2)))
R> print(myocardial.bic, row.names = FALSE)
```

```
classes      bic
      1 524.7441
      2 402.2951
```

Using BIC as a selection method, this selects the 2 class model, indicating a breakdown into diseased and non-diseased, which is assumed to represent those with and without myocardial infarction, although the true nature of classes is always debatable. A characteristic of this data is that a single class random effect model has a lower BIC than the 2 class standard latent class, so we need to assume that there are at least two classes, or that the underlying latent variable is categorical rather than continuous.

An alternative is to use the parametric bootstrap (McLachlan 1987) to determine the number of classes, and this is easily performed using the `simulate` function to generate the samples under the null hypothesis and then `refit` to refit both models. The `simulate` function returns a list of data frames simulated from the specified model, and each of these can then be refitted using the specified null and alternative model, as shown in the following code.

```
R> nsims <- 999
R> obslrt <- 2 * (logLik(myocardial.lca2) - logLik(myocardial.lca1))
R> thesims <- simulate(myocardial.lca1, nsim = nsims)
R> simlrt <- as.vector(lapply(thesims, function(x) {
+   submodel <- refit(myocardial.lca1, newpatterns = x)
+   fullmodel <- refit(myocardial.lca2, newpatterns = x)
+   return(2 * (logLik(fullmodel) - logLik(submodel)))
+ })))
```

A p value is obtained by comparing the observed likelihood ratio test statistic to the simulated. This can be performed in a number of ways, of which the most commonly used is described by Davison and Hinkley (1997, p. 148).

```
R> (sum(simlrt >= obslrt) + 1) / (nsims + 1)
```

```
[1] 0.001
```

Showing again the clear evidence in favor of the 2 class model.

Function `summary` may be used to display the fitted results:

```
R> summary(myocardial.lca2)
```

```
Classes      AIC      BIC      AIC3      logLik penlogLik
  2 379.4054 402.2951 388.4054 -180.7027 -180.7829
Class probabilities
Class 1 Class 2
  0.5422  0.4578
Outcome probabilities
      Q.wave History      LDH      CPK
Class 1 0.0001  0.1951 0.0270 0.1956
Class 2 0.7668  0.7914 0.8279 0.9999
```

From this it is clear that Class 2 is the diseased class and Class 1 the non-diseased, where the disease is myocardial infarction, based on the higher outcome probabilities. Outcome probabilities are plotted using the `plot` function, and shown in Figure 1. Note that `plot` is based on `xyplot` so the additional graphical arguments must be the appropriate ones for the `lattice` package (Sarkar 2008).

```
R> plot(myocardial.lca2, type = "b", pch = 1:2, xlab = "Test",
+       ylab = "Outcome Probability",
+       scales = list(x = list(at = 1:4, labels = names(myocardial)[1:4])),
+       key = list(corner = c(0.05, .95), border = TRUE, cex = 1.2,
+         text = list(c("Class 1", "Class 2")),
+         col = trellis.par.get()$superpose.symbol$col[1:2],
+         points = list(pch = 1:2)))
```

Individual results may be obtained from the `summary` output, for example the outcome probabilities. But these may also be obtained using the `outcomeProbs` function, which will additionally give the 95% confidence intervals.

```
R> outcomeProbs(myocardial.lca2)
```

```
Class 1
      Outcome p      2.5 %      97.5 %
Q.wave 6.882403e-05 6.550006e-22 1.0000000
History 1.951160e-01 1.017699e-01 0.3415272
LDH     2.697733e-02 3.831673e-03 0.1665600
CPK     1.956168e-01 9.784176e-02 0.3528814
Class 2
      Outcome p      2.5 %      97.5 %
```

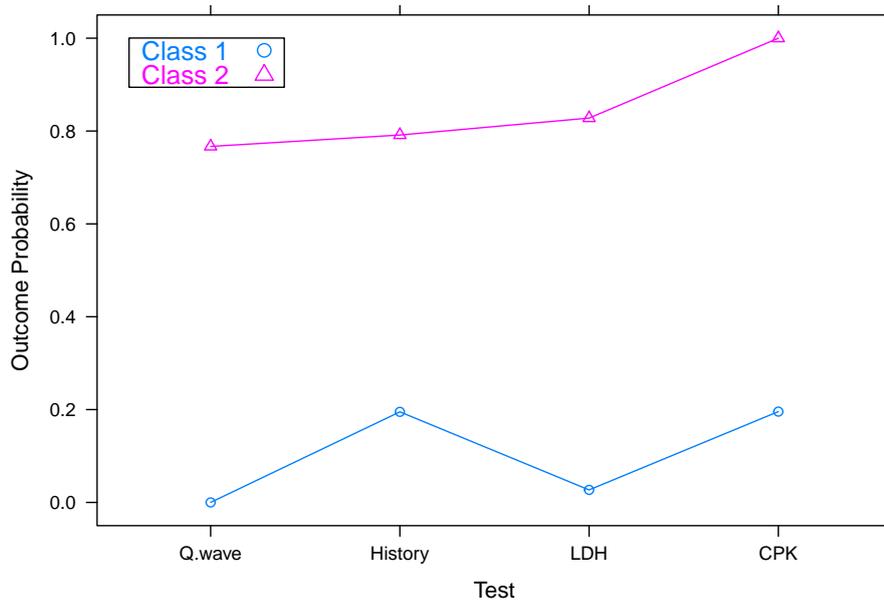


Figure 1: Outcome probabilities for 2 class latent class model for myocardial infarction data.

```

Q.wave  0.7668265  5.919046e-01  0.8817503
History  0.7914044  6.388626e-01  0.8905520
LDH     0.8278971  6.552016e-01  0.9241144
CPK     0.9999367  1.465293e-13  1.0000000

```

For `Q.wave` in Class 1 there appears to be a problem with the standard errors, as assumptions about the normal approximation to the likelihood do not apply close to the boundary. Using the parametric bootstrap with `boot = TRUE` will produce improved results or alternatively, the value of the `penalty` argument could be increased.

```
R> outcomeProbs(myocardial.lca2, boot = TRUE)
```

```

Class 1
      Outcome p      2.5 %      97.5 %
Q.wave  6.882911e-05  4.241592e-05  0.0001315816
History  1.951155e-01  1.033748e-01  0.3755847554
LDH     2.697838e-02  4.759441e-04  0.9879716787
CPK     1.956159e-01  9.335974e-02  0.4145220095
Class 2
      Outcome p      2.5 %      97.5 %
Q.wave  0.7668257  0.5879177  0.8727131
History  0.7914045  0.5907201  0.9025321
LDH     0.8278977  0.5215323  0.9481029
CPK     0.9999367  0.9995986  0.9999949

```

The outcome probabilities give some interesting information. For example, in Class 1, those without myocardial infarction, will have absence of `Q.wave` but in those with myocardial in-

fraction it will only be present with probability 76.7% . The class probabilities can be obtained with `classProbs(myocardial.lca2)` giving 0.54 and 0.46 for Class 1 and 2 respectively.

One aspect of latent class is that no subject is uniquely allocated to a given class, although in some cases a subject may have an extremely high probability of being in a given class. The posterior class probabilities can be obtained as

```
R> print(postClassProbs(myocardial.lca2), row.names = FALSE)
```

Q.wave	History	LDH	CPK	Freq	Class 1	Class 2
1	1	1	1	24	1.670737e-07	9.999998e-01
0	1	1	1	5	7.919596e-03	9.920804e-01
1	0	1	1	4	2.614813e-06	9.999974e-01
0	0	1	1	3	1.110610e-01	8.889390e-01
1	1	0	1	3	2.898730e-05	9.999710e-01
0	1	0	1	5	5.807232e-01	4.192768e-01
1	0	0	1	2	4.534787e-04	9.995465e-01
0	0	0	1	7	9.559027e-01	4.409729e-02
0	0	1	0	1	9.998768e-01	1.232175e-04
0	1	0	0	7	9.999889e-01	1.111584e-05
0	0	0	0	33	9.999993e-01	7.102535e-07

This shows subjects with 3 or 4 positive tests to be strongly classified as having myocardial infarction, and even some with 2 positive tests are well classified. Having only one positive test makes it unlikely that it is myocardial infarction.

3.2. Dentistry example

An important area of application of latent class and random effect latent class is the development or comparison of diagnostic testing methods where there is no gold standard test. A gold standard test is one that is the best available and can usually be assumed to be close to perfect, but usually being more expensive or difficult to perform (Kraemer 1992). Given a gold standard it is easy to construct new tests or compare existing tests, as we know the true disease status of each subject. Latent class methods allow the construction of tests based on the assumption that subjects fall in either two or more classes, with diseased or non-diseased as a minimum, except that the classes can only be inferred from the observed test results. This has the consequence that the status of the subjects is not known exactly, which reduces the accuracy and relies upon the assumptions made about the test result distribution.

There are other methods that have been proposed, the major ones being discrepant resolution and composite reference (Pepe 2003, Chapter 7), both of which have disadvantages and advantages compared to latent class methods. The advantage of the latent class method is its statistical basis, however it has the disadvantage to depend on assumptions about the diagnostic tests, especially the assumption of either conditional independence or normally distributed heterogeneity.

The further arguments to the `randomLCA` function required for a random effect model are:

random: Specifies whether a random effect should be included.

byclass: Allow loadings for the random effect to vary by class.

quadpoints: Number of quadrature points for the adaptive quadrature. These specify how accurate the numerical approximation to the marginal likelihood is, and should be increased until there is negligible improvement in model fit.

constload: The same loading is used for all outcomes when using a random effect model.

blocksize: Blocks allow for the case where loadings are not common to all outcomes but common to subsets of outcomes, allowing for the number of estimated b_{cj} parameters to be reduced compared to individual loadings for each outcome, with **blocksize** specifying the size of each block and the b_{cj} constrained to be equal between blocks. For example with a blocksize of four and four blocks then every fourth outcome will have the same loading, that is $b_{c1} = b_{c5} = b_{c9} = b_{c13}$ and similarly for b_{c2} , b_{c3} and b_{c4} .

probit: Fit a probit model rather than a logistic for the relationship between parameters and outcome probabilities. This is the relationship typically used in some disciplines.

This example shows the fitting of the dentistry data from [Qu *et al.* \(1996\)](#). The data consists of the results of five dentists evaluating X-rays for presence or absence of caries. For consistency with the original paper I have also set `probit = TRUE` to give the probit link. Fitting first the three possible models for one class:

```
R> dentistry.lca1 <- randomLCA(dentistry[, 1:5],
+   freq = dentistry$freq, nclass = 1)
R> dentistry.lca1random <- randomLCA(dentistry[, 1:5],
+   freq = dentistry$freq, nclass = 1, random = TRUE, probit = TRUE)
R> dentistry.lca1random2 <- randomLCA(dentistry[, 1:5],
+   freq = dentistry$freq, nclass = 1, random = TRUE, probit = TRUE,
+   constload = FALSE)
```

This can then be repeated for 2 to 4 classes, and using BIC the BIC can be extracted for each model, and then combined into a data frame to summarize. Note that we cannot use a parametric bootstrap based likelihood ratio test to compare the standard to random effect latent class model as the models are not nested. Here the number of quadrature points will need to be increased for some models to allow convergence.

We can display the BIC values in a table, with the first column for standard latent class, the second for random effects with a constant loading for each dentist, and the third with loading varying by dentist. Note that it is not possible to fit a 4 class random effect model with individual loadings for each dentist due to non-identifiability. As for the previous example we can form the BIC values into a table, where `bic`, `bicrandom` and `bicrandom2` are the BIC values from the standard, random effects with constant loading and random effects with non-constant loading latent class models.

```
R> print(bic.data, row.names = FALSE)
```

classes	bic	bicrandom	bicrandom2
1	17531.13	14974.79	14938.27

2	15021.64	14944.69	14949.37
3	14962.89	14963.54	14992.33
4	15000.03	15007.19	NA

For the standard latent class models the minimum BIC of 14962.9 is obtained for the 3 class model. With addition of the random effect with constant loading, minimum BIC is obtained with a 2 class model with a decrease from the latent class model to 14944.7. Allowing the loadings to vary by dentist (2LCR model obtained by [Qu *et al.* 1996](#)) the minimum BIC of 14938.3 was obtained using a single class model, equivalent to a single factor item response theory (IRT) model ([Bartholomew *et al.* 2002](#), Chapter 7). This assumes that rather than subjects being grouped into classes they simply have different levels of an underlying latent variable, possibly severity. In the absence of any assumptions about the appropriate model this would be the model to be used, and we could conclude that severity of caries was on a continuous scale with each dentist having different thresholds for determining the presence or absence. In the paper by [Qu *et al.* \(1996\)](#) the assumption is made that the underlying latent variable is categorical, that is that there are two distinct types of subjects, and so the 2 class with random effect with constant loading will be used.

Function `summary` may be used to display the fitted results:

```
R> summary(dentistry.lca2random)
```

```
Classes      AIC      BIC      AIC3      logLik penlogLik  Link
  2 14869.56 14944.69 14881.56 -7422.782 -7422.848 Probit
Class probabilities
Class 1 Class 2
  0.821   0.179
Conditional outcome probabilities
      V1      V2      V3      V4      V5
Class 1 0.0057 0.0836 0.0059 0.0258 0.2964
Class 2 0.3804 0.6770 0.6508 0.3999 0.9033
Marginal Outcome Probabilities
      V1      V2      V3      V4      V5
Class 1 0.0192 0.1294 0.0199 0.0558 0.3310
Class 2 0.4017 0.6464 0.6243 0.4179 0.8561
Loadings
0.704419
```

Clearly, based on the lower outcome probabilities, Class 1 is the non-diseased and Class 2 the diseased.

For latent class models with random effects there are two additional arguments to `plot`:

`graphtype`: Type of graph, either "marginal" or "conditional". For marginal the outcome probabilities integrated over the random effect are plotted, and for conditional they are plotted conditional on the random effect, with zero being the default.

`conditionalp`: For a conditional graph the percentile corresponding to the random effect at which the outcome probability is to be calculated. Fifty percent is the default, corresponding to a random effect value of zero.

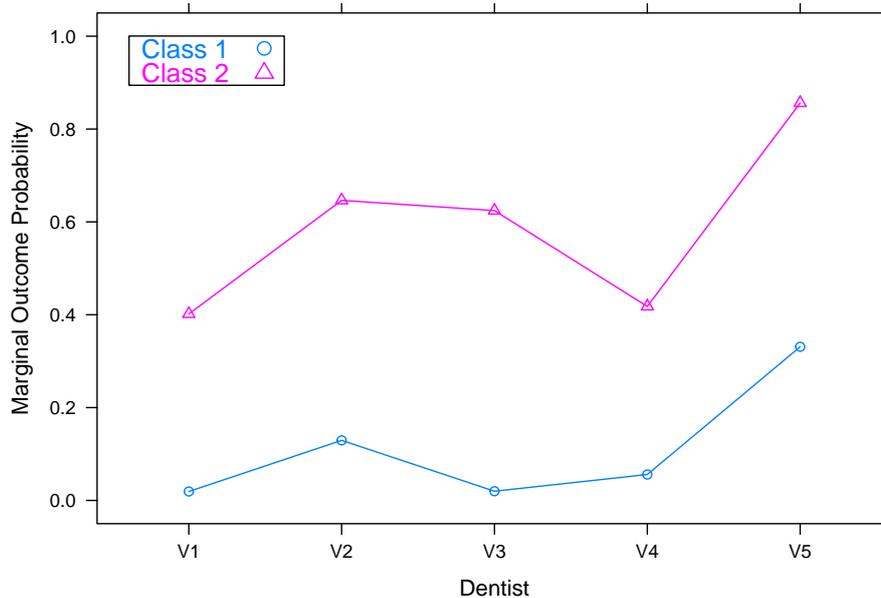


Figure 2: Marginal outcome probabilities for 2 class latent class with random effect (2LCR) model for dentistry data.

The marginal outcome probabilities, obtained by integrating over the random effect can be plotted, as in Figure 2. The marginal outcome probabilities reflect the average probability for any subject in the class having a positive outcome. This differs from the conditional outcome probabilities which are for a subject with zero random effect, and thus represent a typical subject.

```
R> plot(dentistry.lca2random, graphtype = "marginal", type = "b", pch = 1:2,
+       xlab = "Dentist", ylab = "Marginal Outcome Probability",
+       key = list(corner = c(0.05, .95), border = TRUE, cex = 1.2,
+                 text = list(c("Class 1", "Class 2")),
+                 col = trellis.par.get()$superpose.symbol$col[1:2],
+                 points = list(pch = 1:2)))
```

Adding the `boot = TRUE` parameter to the `outcomeProbs` function will obtain bootstrap confidence intervals. Differences from the [Qu *et al.* \(1996\)](#) paper result from their use of an individual loading for each dentist when calculating Table 6.

We can demonstrate the effect of the random effect by plotting the outcome probabilities by varying percentiles of the random effect, using the following code, which will place each percentile in a different panel.

```
R> plot(dentistry.lca2random, graphtype = "conditional", type = "b",
+       pch = 1:2, conditionalp = c(0.025, 0.5, 0.975),
+       scales = list(alternating = FALSE, x = list(cex = 0.8)),
+       xlab = "Dentist", ylab = "Conditional Outcome Probability",
+       key = list(corner = c(0.05, .95), border = TRUE, cex = 1.2,
+                 text = list(c("Class 1", "Class 2")),
+                 col = trellis.par.get()$superpose.symbol$col[1:2], points = TRUE))
```

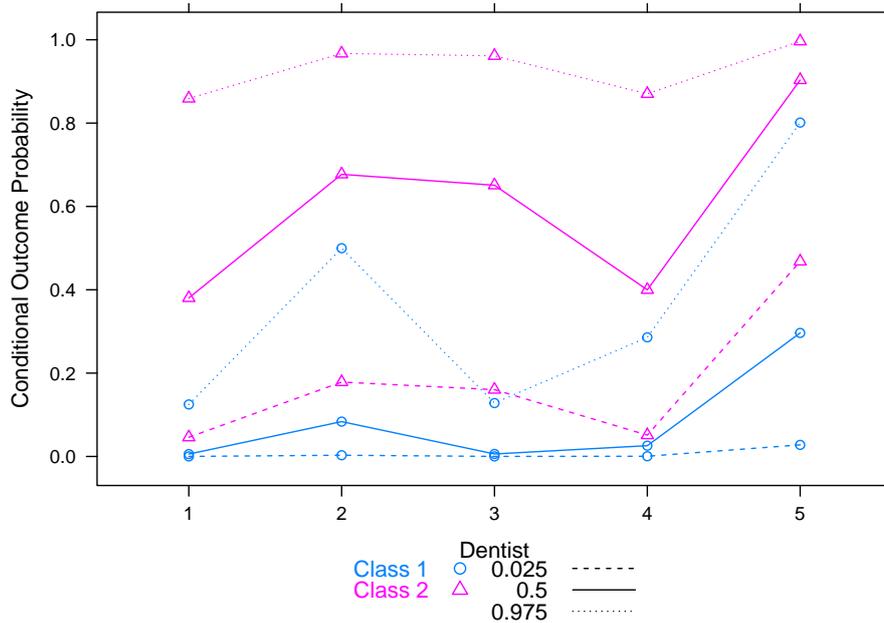


Figure 3: Conditional outcome probabilities for 2 class latent class with random effect (2LCR) model for dentistry data.

If it is desired to have the plots on a single graph, it requires using the `calcCondProb` function which returns a `data.frame` containing the conditional probabilities conditional on the random effect for each class and outcome. We use the 2.5th and 97.5th percentiles, and superpose the plots on the same graph, as shown in Figure 3. As an alternative each class could be placed in a separate panel.

```
R> probs <- calcCondProb(dentistry.lca2random,
+   conditionalp = c(0.025, 0.5, 0.975))
R> my.lty <- c(2, 1, 3)
R> with(probs, xyplot(outcomep ~ outcome, group = class, type = "b",
+   pch = 1:2, panel = function(x, y, groups = groups, ..., type = type,
+   subscripts = subscripts) {
+     panel.superpose(x, y, subscripts, groups, type, ...,
+     panel.groups = function(x, y, col, col.symbol, col.line, pch,
+     ..., subscripts) {
+       thedata <- data.frame(x, y, groups = groups[subscripts],
+       perc = perc[subscripts])
+       by(thedata, thedata$perc, function(x) {
+         lty <- my.lty[x$perc]
+         panel.xyplot(x$x, x$y, col = col, col.symbol = col.symbol,
+         col.line = col.line, lty = lty, pch = pch, type = type)
+       })
+     })
+   },
+   xlab = "Dentist", ylab = "Conditional Outcome Probability",
```

```
+ key = list(space = "bottom", adj = 1,
+   text = list(c("Class 1", "Class 2"),
+     col = trellis.par.get()$superpose.symbol$col[1:2]),
+   points = list(pch = 1:2,
+     col = trellis.par.get()$superpose.symbol$col[1:2]),
+   text = list(levels(perc), col = "black"),
+   lines = list(lty = my.lty, col = "black"), rep = FALSE))
```

Two important concepts in diagnostic testing are sensitivity and specificity. Sensitivity is the probability of obtaining a positive result given that the true state is positive, and specificity is the probability of a negative result given that the true state is negative. Calculation of sensitivity and specificity is shown in the following code, where the number of quadrature points have been increased to ensure convergence for all simulated datasets but may also be obtained by increasing the penalty. The default for evaluation of outcome probabilities for random effect models is marginal, and this is appropriate for obtaining sensitivity and specificity.

```
R> dentistry.lca2random <- randomLCA(dentistry[, 1:5], freq = dentistry$freq,
+   nclass = 2, random = TRUE, quadpoints = 71, probit = TRUE)
R> probs <- outcomeProbs(dentistry.lca2random, boot = TRUE)
```

It is necessary to determine which is the class with higher outcome probabilities, as it is the diseased class. The variable `diseased` gives the number of the diseased class, and `notdiseased` for the non-diseased, and is based on the outcome probabilities being lower for the non-diseased class.

```
R> diseased <- ifelse(probs[[1]]$Outcome[1] < probs[[2]]$Outcome[1], 2, 1)
R> notdiseased <- 3 - diseased
R> sens <- apply(probs[[diseased]], 1, function(x)
+   sprintf("%3.2f (%3.2f, %3.2f)", x[1], x[2], x[3]))
R> spec <- apply(probs[[notdiseased]], 1, function(x)
+   sprintf("%3.2f (%3.2f, %3.2f)", 1 - x[1], 1 - x[3], 1 - x[2]))
R> stable <- data.frame(sens, spec)
R> names(stable) <- c("Sensitivity", "Specificity")
R> print(stable, row.names = TRUE)
```

	Sensitivity	Specificity
V1	0.40 (0.33, 0.48)	0.98 (0.96, 0.99)
V2	0.65 (0.56, 0.71)	0.87 (0.85, 0.89)
V3	0.62 (0.49, 0.72)	0.98 (0.89, 1.00)
V4	0.42 (0.36, 0.48)	0.94 (0.93, 0.96)
V5	0.86 (0.78, 0.91)	0.67 (0.64, 0.70)

The true and false positive rates can be calculated from the outcome probabilities, similar to sensitivity and specificity, and plotted for each dentist, and are shown in Figure 4.

```
R> rates <- data.frame(tpr = probs[[diseased]][, 1],
+   fpr = probs[[notdiseased]][, 1])
```

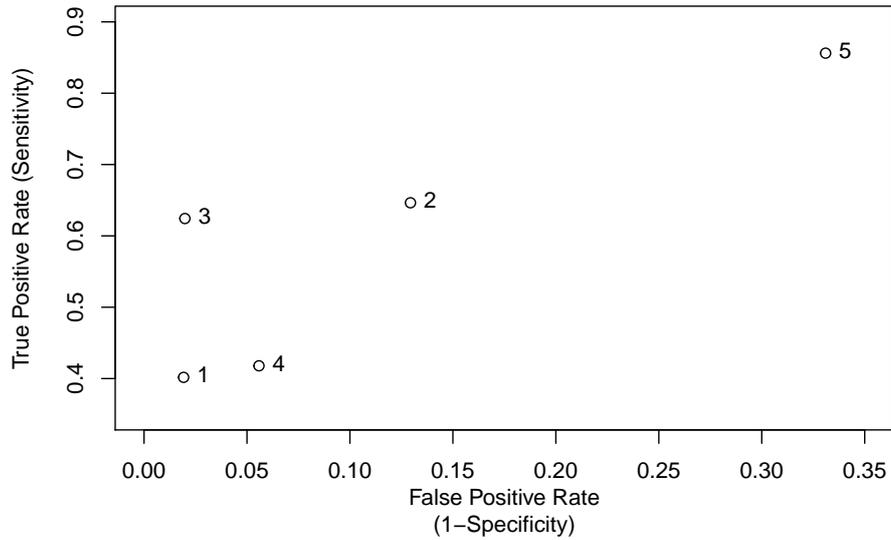


Figure 4: True and false positive rates by dentist.

```
R> plot(tpr ~ fpr, type = "p",
+       xlab = "False Positive Rate\n(1-Specificity)",
+       ylab = "True Positive Rate (Sensitivity)", xlim = c(0.0, 0.35),
+       ylim = c(0.35, 0.9), data = rates)
R> text(rates$fpr, rates$tpr, labels = 1:length(rates$fpr), pos = 4)
```

This gives a better explanation. It appears that the difference between dentists is mainly related to the threshold for what they classify as diseased. Dentist 5 is more likely to correctly identify teeth as diseased but at the expense of being more likely to identify non-diseased teeth as diseased. Note that this is different from an ROC curve where the same data is used but the test threshold is adjusted. Here the dentists may choose different thresholds but may also have different levels of performance.

Posterior class probabilities may again be obtained with `postClassProbs`.

```
R> print(postClassProbs(dentistry.lca2random), row.names = FALSE)
```

V1	V2	V3	V4	V5	Freq	Class 1	Class 2
0	0	0	0	0	1880	0.01510017	0.98489983
0	0	0	0	1	789	0.06917809	0.93082191
0	0	0	1	0	43	0.06932832	0.93067168
0	0	0	1	1	75	0.20226867	0.79773133
0	0	1	0	0	23	0.45787523	0.54212477
0	0	1	0	1	63	0.70501562	0.29498438
0	0	1	1	0	8	0.65707835	0.34292165
0	0	1	1	1	22	0.84127349	0.15872651
0	1	0	0	0	188	0.07570084	0.92429916
0	1	0	0	1	191	0.22845911	0.77154089
0	1	0	1	0	17	0.19730854	0.80269146

```

0 1 0 1 1 67 0.44250827 0.55749173
0 1 1 0 0 15 0.69765202 0.30234798
0 1 1 0 1 85 0.86945151 0.13054849
0 1 1 1 0 8 0.82628366 0.17371634
0 1 1 1 1 56 0.93698420 0.06301580
1 0 0 0 0 22 0.20830077 0.79169923
1 0 0 0 1 26 0.43828265 0.56171735
1 0 0 1 0 6 0.38511871 0.61488129
1 0 0 1 1 14 0.63614562 0.36385438
1 0 1 0 0 1 0.84992008 0.15007992
1 0 1 0 1 20 0.93487333 0.06512667
1 0 1 1 0 2 0.91123313 0.08876687
1 0 1 1 1 17 0.96597651 0.03402349
1 1 0 0 0 2 0.42926553 0.57073447
1 1 0 0 1 20 0.68678150 0.31321850
1 1 0 1 0 6 0.61133943 0.38866057
1 1 0 1 1 27 0.82621026 0.17378974
1 1 1 0 0 3 0.92818897 0.07181103
1 1 1 0 1 72 0.97420228 0.02579772
1 1 1 1 0 1 0.95984998 0.04015002
1 1 1 1 1 100 0.98854608 0.01145392

```

Clearly, as the number of dentists identifying the subject as diseased increases, the posterior probability of being diseased increases, until it is almost one when all dentists identify the subject as diseased. The observed and fitted values may be obtained using the fitted method which returns a data frame containing them. Again, differences from the [Qu *et al.* \(1996\)](#) paper result from a model with different loading for each dentist. We can obtain the fitted values for the two models as follows:

```

R> dentistry.lca2.fitted <- fitted(dentistry.lca2)
R> dentistry.lca2random.fitted <- fitted(dentistry.lca2random)
R> dentistry.fitted <- merge(dentistry.lca2.fitted,
+   dentistry.lca2random.fitted, by = names(dentistry.lca2.fitted)[1:6])
R> names(dentistry.fitted)[6:8] <- c("Obs", "Exp 2LC", "Exp 2LCR")
R> print(dentistry.fitted, row.names = FALSE)

```

V1	V2	V3	V4	V5	Obs	Exp 2LC	Exp 2LCR
0	0	0	0	0	1880	1836.272578	1882.192222
0	0	0	0	1	789	830.353393	779.798018
0	0	0	1	0	43	61.935450	56.084222
0	0	0	1	1	75	49.638811	72.333920
0	0	1	0	0	23	28.630160	25.843397
0	0	1	0	1	63	47.477153	60.365771
0	0	1	1	0	8	4.035482	4.693956
0	0	1	1	1	22	35.146073	25.089529
0	1	0	0	0	188	213.893779	176.134440
0	1	0	0	1	191	152.205458	209.645063

0	1	0	1	0	17	12.146964	17.583716
0	1	0	1	1	67	61.010959	53.823627
0	1	1	0	0	15	11.208702	14.481072
0	1	1	0	1	85	91.569225	78.991512
0	1	1	1	0	8	8.068059	5.582296
0	1	1	1	1	56	86.406394	67.144822
1	0	0	0	0	22	21.210790	16.998041
1	0	0	0	1	26	25.170942	30.581733
1	0	0	1	0	6	2.100453	2.526789
1	0	0	1	1	14	16.081347	10.598155
1	0	1	0	0	1	2.541623	3.323287
1	0	1	0	1	20	24.707672	20.750489
1	0	1	1	0	2	2.180089	1.437972
1	0	1	1	1	17	23.518862	17.796230
1	1	0	0	0	2	6.023171	7.401450
1	1	0	0	1	20	42.001504	31.873362
1	1	0	1	0	6	3.694997	2.415690
1	1	0	1	1	27	39.260267	23.634112
1	1	1	0	0	3	5.667893	4.566129
1	1	1	0	1	72	61.064125	59.378193
1	1	1	1	0	1	5.391983	3.466835
1	1	1	1	1	100	58.385645	102.463952

It can be seen how the random effect model more accurately models the data, with fitted values closer to the observed data.

3.3. Symptoms example

This comprises data on the presence or absence of respiratory and allergy symptoms in the Childhood Asthma Prevention Study (CAPS; [Mihirshahi, Peat, Webb, Tovey, Marks, Mellis, and Leeder 2001](#)) and was used as the example in [Beath and Heller \(2009\)](#). The symptoms of night cough, wheeze, itchy rash and flexural dermatitis since the previous visit were recorded at one month, then quarterly for the first year and then twice yearly until the age of two years. For analysis these are aggregated for each six month period to avoid numerical problems associated with very small probabilities. The aim of the analysis is to identify the number of classes of subjects based on their respiratory and allergy symptoms combined. As well as allowing for the classes to be defined by different levels of the symptoms it will also allow for changes over time.

For **randomLCA** the data is required in wide format with the four outcomes repeated in order for the number of periods. While it is not necessary, each outcome is suffixed with a period and the corresponding time identifier. This makes interpretation of the results easier and also will be used in labeling of the graphs. The structure of the data is:

```
R> names(symptoms)
```

```
[1] "Nightcough.13" "Wheeze.13"      "Itchyrash.13"  "FlexDerma.13"
[5] "Nightcough.45" "Wheeze.45"      "Itchyrash.45"  "FlexDerma.45"
```

```
[9] "Nightcough.6" "Wheeze.6"      "Itchy rash.6"  "FlexDerma.6"
[13] "Nightcough.7" "Wheeze.7"      "Itchy rash.7"  "FlexDerma.7"
[17] "Freq"
```

The following two additional arguments to `randomLCA` are used to define the two level random effect model:

`level2`: Fit 2 level random effect model.

`level2size`: Size of level 2 blocks for fitting 2 level models.

The first model fitted is a standard latent class, to allow for no subject or period effect.

```
R> symptoms.lca2 <- randomLCA(symptoms[, 1:16], freq = symptoms$Freq,
+   nclass = 2)
```

A variation of the random effect latent class model can be fitted allowing the loadings (b_{cj} parameters) for the outcomes to be repeated, that is wheeze at the different time points, for example, will always have the same loading, using the `blocksize` argument. This is equivalent to the 2 level model with the time-dependent random variable having a variance of zero, that is no time-dependent effect. The outcomes have been set to have non-constant loading, although in practice models for constant loading would also be fitted.

```
R> symptoms.lca2random <- randomLCA(symptoms[, 1:16], freq = symptoms$Freq,
+   random = TRUE, nclass = 2, blocksize = 4, constload = FALSE)
```

The two level models are specified through the `level2` argument and the number of outcomes at each time through the `level2size` argument. For these models the penalty is increased to 0.1 to reduce the execution time, but for a two class model this will be about an hour and for a three class model about two hours.

```
R> symptoms.lca2random2 <- randomLCA(symptoms[, 1:16], freq = symptoms$Freq,
+   random = TRUE, level2 = TRUE, nclass = 2, level2size = 4,
+   constload = FALSE, penalty = 0.1)
```

Repeating for up to five classes or when the BIC increases gives the following results. It should be noted that the two level models can take considerable time to fit. This is due to the relatively large number of quadrature points required for this data, as a consequence of a large number of patterns consisting entirely of zeroes. As for the previous examples we can form the BIC values into a table, where `bic`, `bic.random` and `bic.random2` are the BIC values from the standard, random effects and 2 level random effects latent class models.

class	bic	bic.random	bic.random2
1	10844.187	9987.510	9595.674
2	10123.811	9769.743	9363.986
3	9960.839	9700.262	9399.375
4	9910.742	9726.847	NA
5	9916.665	9731.190	NA

This shows the optimal model is the 2 class model with random effects for both subject and period.

```
R> summary(symptoms.lca2random2)
```

```
> summary(symptoms.lca2random2)
```

```
Classes      AIC      BIC      AIC3      logLik penlogLik Link
  2 9198.454 9363.986 9236.454 -4561.227 -4563.496 Logit
Class probabilities
Class 1 Class 2
  0.5704  0.4296
Conditional outcome probabilities
      Nightcough.13 Wheeze.13 Itchy rash.13 FlexDerma.13 Nightcough.45
Class 1      0.3675      0.2319      0.0908      0.0926      0.4408
Class 2      0.8031      0.6244      0.1341      0.0605      0.8731
      Wheeze.45 Itchy rash.45 FlexDerma.45 Nightcough.6 Wheeze.6 Itchy rash.6
Class 1      0.1581      0.0335      0.0625      0.3089      0.0853      0.0029
Class 2      0.6723      0.1677      0.0536      0.6526      0.4577      0.0055
      FlexDerma.6 Nightcough.7 Wheeze.7 Itchy rash.7 FlexDerma.7
Class 1      0.0166      0.2276      0.0476      0.0021      0.0044
Class 2      0.0097      0.6216      0.3772      0.0037      0.0154
Marginal Outcome Probabilities
      Nightcough.13 Wheeze.13 Itchy rash.13 FlexDerma.13 Nightcough.45
Class 1      0.3681      0.2356      0.4003      0.3112      0.4411
Class 2      0.8022      0.6221      0.4190      0.2770      0.8723
      Wheeze.45 Itchy rash.45 FlexDerma.45 Nightcough.6 Wheeze.6 Itchy rash.6
Class 1      0.1617      0.3553      0.2796      0.3097      0.0879      0.2596
Class 2      0.6694      0.4302      0.2679      0.6519      0.4585      0.2842
      FlexDerma.6 Nightcough.7 Wheeze.7 Itchy rash.7 FlexDerma.7
Class 1      0.1894      0.2286      0.0493      0.2478      0.1213
Class 2      0.1595      0.6211      0.3794      0.2686      0.1850
Loadings
      Nightcough Wheeze Itchy rash FlexDerma
      0.1009 0.2043 6.4509 3.1001
Tau
0.9518
```

The marginal outcome probabilities are plotted in Figure 5 as follows.

```
R> plot(symptoms.lca2random2, type = "b",
+       scales = list(x = list(at = 1:4, labels = c(6, 12, 18, 24))),
+       pch = 1:4, xlab = "Period",
+       key = list(corner = c(0.05, .90),
+                 text = list(c("Night Cough", "Wheeze", "Itch Rash", "Flex. Derma.")),
+                 points = list(pch = 1:4),
+                 col = trellis.par.get()$superpose.symbol$col[1:4], border = TRUE))
```

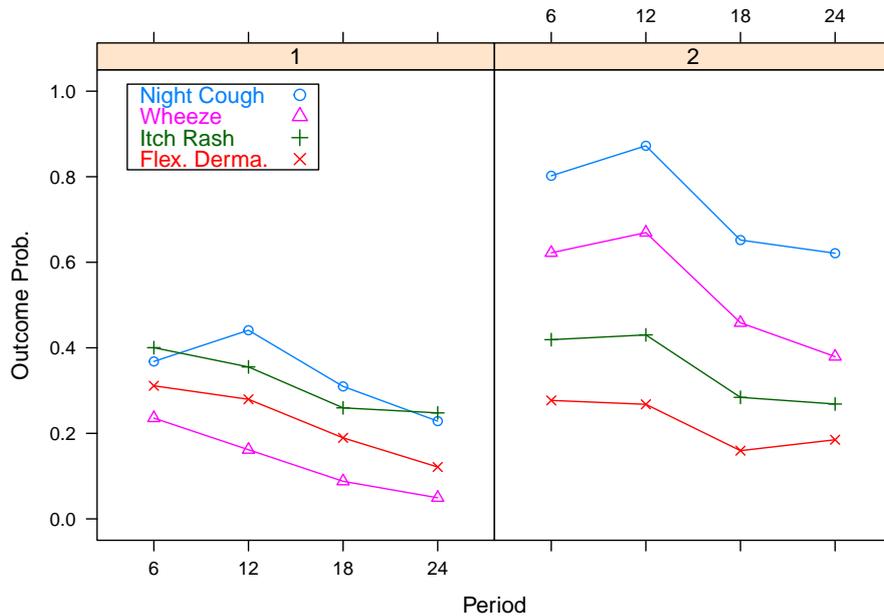


Figure 5: Marginal outcome probabilities for 2 class latent class for symptoms data.

It can be seen that in Class 2 the outcome probabilities are greater than for Class 1, with the difference greatest for the two respiratory symptoms night cough and wheeze. Also over time the outcome probabilities decrease. Similarly to the dentistry example the conditional probabilities can be obtained using `calcCond2Prob` and plotted.

4. Summary

It has been shown how the **randomLCA** package may be used to determine classes of subjects based on observed binary data, and how this may be used to determine sensitivity and specificity for diagnostic tests. In the dentistry and symptoms examples an assumption of heterogeneous classes, where the outcome probabilities are allowed to vary within a class was shown to be an improvement over traditional latent class analysis. The **randomLCA** package also produces a range of plots for describing the classes, allows for bootstrapped standard errors, calculation of marginal outcome probabilities when using random effect models and the use of penalized likelihoods. A further extension to **randomLCA** would be the extension to latent class regression models (Dayton and Macready 1988), where the class probabilities are determined by the covariates. Another extension is to allow for ordinal data using a graded response model (Samejima 1969). A further possible extension to the package is to allow for other data types. However for the random effect models it is difficult to extend the models except for ordinal data.

References

Bartholomew DJ, Steele F, Moustaki I, Galbraith JI (2002). *The Analysis and Interpretation of Multivariate Data for Social Scientists*. Chapman & Hall/CRC.

- Beath K (2017). **randomLCA**: *Random Effects Latent Class Analysis*. R package version 1.0-13, URL <https://CRAN.R-project.org/package=randomLCA>.
- Beath KJ, Heller GZ (2009). “Latent Trajectory Modelling of Multivariate Binary Data.” *Statistical Modelling*, **9**(3), 199–213. doi:10.1177/1471082x0800900302.
- Davison AC, Hinkley DV (1997). *Bootstrap Methods and Their Application*. Cambridge University Press, Cambridge.
- Dayton CM, Macready GB (1988). “Concomitant-Variable Latent-Class Models.” *Journal of the American Statistical Association*, **83**(401), 173–178. doi:10.2307/2288938.
- Dias JG (2006). “Latent Class Analysis and Model Selection.” In M Spiliopoulou, R Kruse, C Borgelt, A Nürnberger, W Gaul (eds.), *From Data and Information Analysis to Knowledge*, pp. 95–102. Springer-Verlag.
- Firth D (1993). “Bias Reduction of Maximum Likelihood Estimates.” *Biometrika*, **80**(1), 27–38.
- Formann AK (1978). “A Note on Parameter Estimation for Lazarsfeld’s Latent Class Analysis.” *Psychometrika*, **43**(1), 123–126. doi:10.1007/bf02294098.
- Formann AK (1982). “Linear Logistic Latent Class Analysis.” *Biometrical Journal*, **24**(2), 171–190. doi:10.1002/bimj.4710240209.
- Galindo Garre F, Vermunt JK (2006). “Avoiding Boundary Estimates in Latent Class Analysis by Bayesian Posterior Mode Estimation.” *Behaviormetrika*, **33**(1), 43–59. doi:10.2333/bhmk.33.43.
- Golden RR (1982). “A Taxometric Model for the Detection of a Conjectured Latent Taxon.” *Multivariate Behavioral Research*, **17**(3), 389–416. doi:10.1207/s15327906mbr1703_6.
- Kraemer HC (1992). *Evaluating Medical Tests: Objective and Quantitative Guidelines*. Sage Publications, Newbury Park.
- Langeheine R, Van de Pol F (1990). “A Unifying Framework for Markov Modelling in Discrete Space and Discrete Time.” *Sociological Methods and Research*, **18**(4), 416–441. doi:10.1177/0049124190018004002.
- Lazarsfeld PF, Henry NW (1968). *Latent Structure Analysis*. Houghton Mifflin, Boston, Massachusetts.
- Lin TH, Dayton CM (1997). “Model Selection Information Criteria for Non-Nested Latent Class Models.” *Journal of Educational and Behavioral Statistics*, **22**(3), 249–264. doi:10.3102/10769986022003249.
- Linzer DA, Lewis JB (2011). “**poLCA**: An R Package for Polytomous Variable Latent Class Analysis.” *Journal of Statistical Software*, **42**(10), 1–29. doi:10.18637/jss.v042.i10.
- Little RJA, Rubin DB (2002). *Statistical Analysis with Missing Data*. 2nd edition. John Wiley & Sons, Hoboken. doi:10.1002/9781119013563.

- Liu Q, Pierce DA (1994). “A Note on Gauss-Hermite Quadrature.” *Biometrika*, **81**(3), 624–629. doi:10.2307/2337136.
- McHugh RB (1956). “Efficient Estimation and Local Identification in Latent Class Analysis.” *Psychometrika*, **21**(4), 331–347. doi:10.1007/bf02296300.
- McLachlan GJ (1987). “On Bootstrapping the Likelihood Ratio Test Statistic for the Number of Components in a Normal Mixture.” *Journal of the Royal Statistical Society C*, **36**(3), 318–324. doi:10.2307/2347790.
- McLachlan GJ, Peel D (2000). *Finite Mixture Models*. John Wiley & Sons, New York. doi:10.1002/0471721182.
- Mihrshahi S, Peat JK, Webb K, Tovey ER, Marks GB, Mellis CM, Leeder SR (2001). “The Childhood Asthma Prevention Study (CAPS): Design and Research Protocol of a Randomized Trial for the Primary Prevention of Asthma.” *Controlled Clinical Trials*, **22**(3), 333–354. doi:10.1016/S0197-2456(01)00112-X.
- Muthén B (2006). “Should Substance Use Disorders be Considered as Categorical or Dimensional?” *Addiction*, **101**(Supplement 1), 6–16. doi:10.1111/j.1360-0443.2006.01583.x.
- Muthén B, Shedden K (1999). “Finite Mixture Modeling with Mixture Outcomes Using the EM Algorithm.” *Biometrics*, **55**(2), 463–469. doi:10.1111/j.0006-341x.1999.00463.x.
- Muthén LK, Muthén BO (2015). *Mplus User’s Guide*. Muthén & Muthén, Los Angeles, 7th edition.
- Nyholt DR, Gillespie NG, Heath AC, Merikangas KR, Duffy DL, Martin NG (2004). “Latent Class and Genetic Analysis Does Not Support Migraine with Aura and Migraine without Aura as Separate Entities.” *Genetic Epidemiology*, **26**(3), 231–244. doi:10.1002/gepi.10311.
- Nylund KL, Muthén BO (2007). “Deciding on the Number of Classes in Latent Class Analysis and Growth Mixture Modeling: A Monte Carlo Simulation Study.” *Structural Equation Modeling*, **14**(4), 535–569. doi:10.1080/10705510701575396.
- Pepe MS (2003). *The Statistical Evaluation of Medical Tests for Classification and Prediction*. Oxford University Press, New York.
- Pickles A, Angold A (2003). “Natural Categories or Fundamental Dimensions: On Carving Nature at the Joints and the Rearticulation of Psychopathology.” *Development and Psychopathology*, **15**(3), 529–551. doi:10.1017/s0954579403000282.
- Qu Y, Tan M, Kutner MH (1996). “Random Effects Models in Latent Class Analysis for Evaluating Accuracy of Diagnostic Tests.” *Biometrics*, **52**(3), 797–810. doi:10.2307/2533043.
- Rabe-Hesketh S, Skrondal A, Pickles A (2005). “Maximum Likelihood Estimation of Limited and Discrete Dependent Variable Models with Nested Random Effects.” *Journal of Econometrics*, **128**(2), 301–323. doi:10.1016/j.jeconom.2004.08.017.

- R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Rindskopf D, Rindskopf W (1986). “The Value of Latent Class Analysis in Medical Diagnosis.” *Statistics in Medicine*, **5**(1), 21–27. doi:10.1002/sim.4780050105.
- Rubin DB, Schenker N (1987). “Logit-Based Interval Estimation for Binomial Data Using the Jeffreys Prior.” *Sociological Methodology*, **17**, 131–144. doi:10.2307/271031.
- Samejima F (1969). “Estimation of Latent Ability Using a Response Pattern of Graded Scores.” *Psychometrika*, **34**(Supplement 1), 1–97. doi:10.1007/BF03372160.
- Sarkar D (2008). *lattice: Multivariate Data Visualization with R*. Springer-Verlag, New York.
- Skrondal A, Rabe-Hesketh S (2004). *Generalized Latent Variable Modeling: Multilevel, Longitudinal and Structural Equation Models*. Chapman & Hall/CRC, Boca Raton.
- Uebersax JS (1999). “Probit Latent Class Analysis with Dichotomous or Ordered Category Measures: Conditional Independence/Dependence Models.” *Applied Psychological Measurement*, **23**(4), 283–297. doi:10.1177/01466219922031400.
- Vacek PM (1985). “The Effect of Conditional Dependence on the Evaluation of Diagnostic Tests.” *Biometrics*, **41**(4), 959–968. doi:10.2307/2530967.
- Vermunt JK, Magidson J (2013). *LG Syntax User’s Guide: Manual for Latent GOLD 5.0 Syntax Module*. Belmont.
- White A, Murphy TB (2014). “**BayesLCA**: An R Package for Bayesian Latent Class.” *Journal of Statistical Software*, **61**(13), 1–28. doi:10.18637/jss.v061.i13.
- Young MA (1982). “Evaluating Diagnostic Criteria: A Latent Class Paradigm.” *Journal of Psychiatric Research*, **17**(3), 285–296. doi:10.1016/0022-3956(82)90007-3.

Affiliation:

Ken J. Beath
Department of Statistics
Faculty of Science and Engineering
Macquarie University NSW 2109
Australia
E-mail: ken.beath@mq.edu.au
URL: <https://directory.science.mq.edu.au/users/kbeath>