ROYAL STATISTICAL SOCIETY
DATA | EVIDENCE | DECISIONS

Journal of the Royal Statistical Society
Statistics in Society
Series A

J. R. Statist. Soc. A (2019)
182, *Part* 4, *pp.* 1463–1486

Estimating the number of serious road injuries per vehicle type in the Netherlands by using multiple imputation of latent classes

Laura Boeschoten and Ton de Waal

Tilburg University, and Centraal Bureau voor de Statistiek, The Hague, The Netherlands

and Jeroen K. Vermunt

Tilburg University, The Netherlands

[Received March 2018. Final revision April 2019]

Summary. Statistics that are published by official agencies are often generated by using population registries, which are likely to contain classification errors and missing values. A method that simultaneously handles classification errors and missing values is multiple imputation of latent classes (MILC). We apply the MILC method to estimate the number of serious road injuries per vehicle type in the Netherlands and to stratify the number of serious road injuries per vehicle type into relevant subgroups by using data from two registries. For this specific application, the MILC method is extended to handle the large number of missing values in the stratification variable 'region of accident' and to include more stratification covariates. After applying the extended MILC method, a multiply imputed data set is generated that can be used to create statistical figures in a straightforward manner, and that incorporates uncertainty due to classification errors and missing values in the estimate of the total variance.

Keywords: Classification error; Combined data set; Latent class analysis; Missing values; Multiple imputation

1. Introduction

When statistics are published by government or other official agencies, population registries are often utilized to generate these statistics. Here, caution is advised as population registries are collected for administrative purposes so they may not align conceptually with the target of interest. Furthermore, they are likely to contain process-delivered classification errors. Another issue is that population registries are likely not to have registered every single unit in the population of interest, so the population registry is not complete.

An official agency dealing with the issues of classification errors and missing units in registers when generating statistics is the Institute for Road Safety Research (in Dutch: Stichting Wetenschappelijk Onderzoek Verkeersveiligheid (SWOV)). An important statistic that the SWOV publishes every year is the number of serious road injuries in the Netherlands. The number of serious road injuries is important because it is used to define the road safety target (Reurings and Stipdonk, 2011). To gain more insight into the total number of serious road injuries, it can be further stratified by vehicle type, severity of injury and region (Reurings and Bos, 2012).

Address for correspondence: Laura Boeschoten, Department of Methodology and Statistics, Tilburg School of Social and Behavioral Sciences, Tilburg University, Warandelaan 2, Tilburg 5000 E, The Netherlands.
E-mail: L.Boeschoten@uvt.nl

When estimating the number of serious road injuries in the Netherlands, the SWOV uses information from police and hospital registries. These registries contain classification errors and are incomplete. The SWOV estimates the number of units that are missing in both registries by a method based on capture–recapture (Reurings and Stipdonk, 2011). However, a procedure to correct for classification errors and missing values within the observed cases has not been applied.

A method to deal simultaneously with classification errors and missing values within the observed cases is the recently proposed multiple imputation of latent classes (MILC) method (Boeschoten *et al.*, 2017). The MILC method combines two existing statistical methods: multiple imputation and latent class analysis. To apply the MILC method, it is necessary to have multiple population registries that can be linked at a unit level. All registries are required to contain identifier variables for their cases which makes it possible to link the information for a specific case in one registry to its information in the other registries. In such a combined data set, variables are selected that measure the same construct but originate from the different registries. They are used as indicators of a latent variable of which it can be said that it contains the 'true scores' which are estimated by using a latent class model. Information from the latent class model is then used to create multiple imputations of the 'true variable'. The multiply imputed data sets can be used to generate statistics of interest, graphs or frequency tables. Uncertainty due to classification errors and missing cases is reflected in the differences between the imputations and is incorporated in the estimate of the total variance (Rubin (1987), page 76).

In this paper, the MILC method is applied to a linked data set containing a police and a hospital registry, to estimate the number of serious road injuries per vehicle type. Next, two variables measuring vehicle type are used as indicators of a latent variable measuring the 'true' vehicle type. Because of the way in which this data set is constructed, a special feature of this data set is that, whenever one of these two indicators is missing, the other is observed. To stratify the serious road injuries into relevant groups, covariates are included in the latent class model.

A statistic that is currently not straightforward to estimate is the number of serious road injuries per vehicle type per region, because the variable 'region of accident' is observed in the police registry only and contains many missing cases. To estimate this statistic, the MILC method is extended in two ways. First, the MILC method is extended to estimate two latent variables simultaneously (vehicle type and region of accident). For the latent variable vehicle type, two imperfectly measured indicators are specified. For the latent variable region of accident, one indicator (containing missing values) is assumed to be a perfect representation of the latent variable, next to a second, imperfectly measured, indicator. Second, the MILC method is extended to incorporate more covariates for investigating relevant stratifications in general. In the remainder of this paper, we refer to this as the 'extended MILC method'.

In the next section, a more detailed description of the data to which the extended MILC method is applied is given. In the third section, a detailed description is given of how the extended MILC method is applied to the unit-linked police–hospital data sets. In addition, an illustrative simulation study is performed. Here, the results that are obtained after applying the extended MILC method are compared with results that are obtained after applying a more traditional hierarchical assignment procedure. In the fourth section, the output from the latent class model and the number of serious road injuries are discussed.

The programs that were used to analyse the data can be obtained from

```
https://rss.onlinelibrary.wiley.com/hub/journal/1467985x/series-
a-datasets
```

2. Background

The extended MILC method is applied on a unit-linked data set containing a police and a hospital registry. It is applied separately to data sets from 1994, 2009 and 2014 as the quality of the registries has changed substantially over time. In this section, the process of constructing these data sets is described and variables of interest are discussed in more detail.

For every year, units that are observed in the two sources are linked by using information on personal and accident characteristics (Reurings and Stipdonk, 2009). Changes in registration systems over time influenced the success rate of the linking procedure. In addition, a weighting factor was determined for many of the individual cases (Bos *et al.*, 2017).

2.1. Variables measuring 'vehicle type'

As can be seen in Table 1, the variable vehicle type is observed in both the police and the hospital

Table 1. Cross-table between the variables measuring vehicle type originating from the police registry (columns) and from the hospital registry (rows) for the years 1994, 2009 and 2013†

Year	*Category*	*Missing value*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*9*	*Total*
1994	Missing value	—	561	245	318	122	42	137	90	14	1529
	1 M–car	918	2596	11	72	12	22	25	2	1	3659
	2 M–moped	702	29	1131	21	60	2	8	2	1	1956
	3 M–bicycle	397	40	70	1111	2	1	53	25	4	1703
	4 M–motorcycle	347	16	41	2	633	3	0	0	0	1042
	5 M–other	450	408	106	104	35	50	116	8	2	1279
	6 M–pedestrian	421	128	37	231	4	5	537	5	5	1373
	7 N–bicycle	3625	28	41	221	3	3	11	296	3	4231
	8 N–other	34	1	0	2	0	4	0	2	0	43
	9 N–pedestrian	94	2	2	2	0	0	20	6	22	148
	Total	6988	3809	1684	2084	871	132	907	436	52	16963
2009	Missing value	—	209	111	126	38	20	62	26	6	598
	1 M–car	779	969	8	29	8	17	3	0	0	1813
	2 M–moped	1117	4	611	10	23	20	2	0	0	1787
	3 M–bicycle	565	23	17	701	0	9	20	9	0	1344
	4 M–motorcycle	668	9	74	2	367	6	0	0	0	1126
	5 M–other	350	51	40	21	11	23	23	1	1	521
	6 M–pedestrian	363	39	15	62	2	2	202	2	2	689
	7 N–bicycle	6369	17	22	161	2	4	5	144	4	6728
	8 N–other	99	0	2	4	0	0	0	4	1	110
	9 N–pedestrian	136	0	1	4	0	0	6	8	16	171
	Total	10446	1321	901	1120	451	101	323	194	30	14887
2013	Missing value	—	59	29	33	15	36	11	5	1	189
	1 M–car	877	566	3	1	4	65	3	0	0	1519
	2 M–moped	2220	8	419	3	167	63	2	1	0	2883
	3 M–bicycle	944	4	11	451	0	155	10	7	0	1582
	4 M–motorcycle	69	0	10	0	21	3	0	0	0	103
	5 M–other	556	18	8	1	19	27	4	0	0	633
	6 M–pedestrian	392	2	3	30	0	64	123	0	1	615
	7 N–bicycle	7230	12	7	41	1	29	2	44	1	7367
	8 N–other	13	0	0	0	0	0	0	0	0	13
	9 N–pedestrian	117	0	0	1	0	4	2	0	5	129
	Total	12418	669	490	561	227	446	157	57	8	15033

†Note that there are no observations for the category 'Non-motorized—other' in the police registry.

registry and has nine categories. The categories make a distinction between injuries caused by motorized vehicles (with an 'M' in the category label) and non-motorized vehicles (with an 'N' in the category label). For example, there is a category 'M–bicycle' and 'N–bicycle'. The difference between these categories is that, for the category M–bicycle, the injured person was on a bike and experienced an accident with a motorized vehicle, whereas, for the category N–bicycle, the injured person was on a bike and no motorized vehicle was involved in the accident. The distinction between motorized and non-motorized is important because it provides information on the cause of the injury. For example, when the number of injuries increases in the category N–bicycle, it can be caused by unsafe bicycle lanes. If the number of injuries increases in the category M–bicycle, it can be caused by a high speed limit on roads that are shared by cars and bicycles.

As shown in Table 1, many injuries were classified differently by the police and the hospital. In addition, it can also be seen that injuries in the 'non-motorized' ('N') categories are particularly often missing in the police registry, as the police are generally not involved in, for example, one-sided bicycle accidents. Also note that the category 'N–other' is not observed in the police registry at all.

2.2. Variables describing relevant subgroups

Besides estimating the number of serious road injuries per vehicle type, stratifications in relevant subgroups need to be made, such as age, gender, severity of injury or region of accident. To be

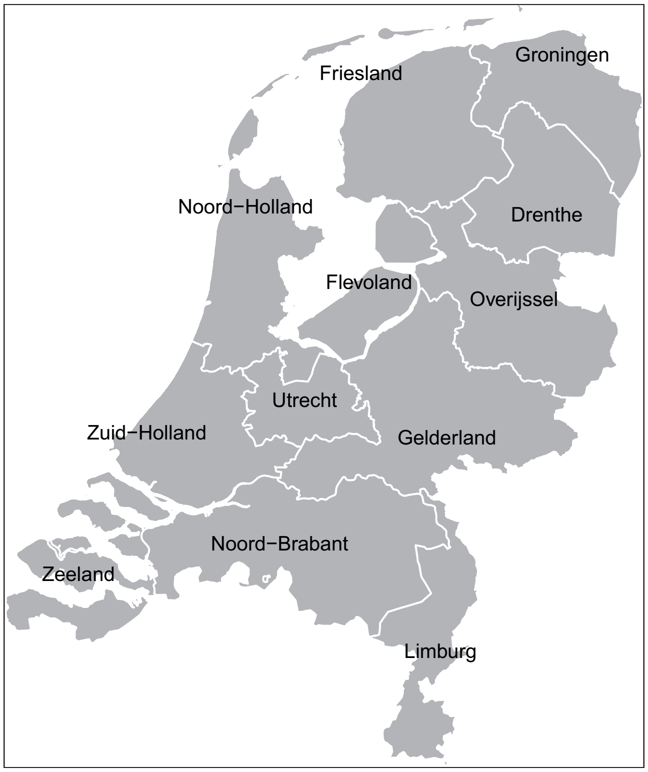


Fig. 1. Map of the Netherlands

able to make such stratifications, the variables need to be included as covariates in the latent class model that is used to estimate 'true vehicle type'.

The reason for estimating the latent class model is to create imputations for true vehicle type for every observed case. To be able to stratify all cases, the covariates need to be observed completely as well. For the variables 'age', 'gender' and 'injury severity' this is so. For the variable region of accident, this is a problem, as this variable is observed in the police registry only.

To solve the issue of missing values in region of accident the traditional MILC method is extended in such a way that missing values in region of accident are imputed simultaneously whereas the latent variable true vehicle type is estimated. To create these imputations, information is used from region of hospital, which is observed for the cases that contain missing values for region of accident. The two variables have a strong, but not perfect, relationship. For example, from the serious road injuries in 2013 of which the injured person was in a hospital in Groningen, 53 were also registered to have taken place in Groningen, whereas 12 of those accidents were registered to have taken place in Friesland (Table 2), which is a neighbouring region of Groningen. There was also one person in a hospital in Groningen for whom the accident was registered to be in Zuid-Holland, which is quite far away from Groningen (see Fig. 1 for the regions of the Netherlands). A reason for this observation can be classification error in one of the registries or incorrect linkage of a case in the police registry to a case in the hospital registry (wrongfully assuming that the cases contained the same person). However, it is also possible that this person indeed had a road accident in Zuid-Holland and was transfered to a hospital in Groningen because it was closer to the person's home or it could provide a form of specialized healthcare.

3. Applying the extended multiple imputation of latent classes method

In this section, it is described step by step how the extended MILC method is applied to estimate the number of serious road injuries per vehicle type in the Netherlands. The procedure of applying the MILC method starts with the data set that is linked and processed as described in the previous section.

3.1. Bootstrapping for parameter uncertainty

To account for parameter uncertainty when applying the extended MILC method, we use a non-parametric bootstrap procedure. This involves creating M bootstrap samples by drawing observations from the observed data set with replacement. Subsequently, for each bootstrap sample, the latent class model of interest is estimated and the M imputations are created by using the M sets of parameter values that are obtained. This is preferable over creating imputations based on the maximum likelihood estimates that are obtained with the observed data, which would imply ignoring the uncertainty regarding the estimated parameters of the latent class model. Thus, by applying a non-parametric bootstrap procedure, parameter uncertainty is incorporated in the final pooled standard error estimates of the statistics of interest.

3.2. Specifying the latent class model

The second step of the extended MILC method is specifying the latent class model. The latent class model is estimated separately for each bootstrap sample so that the differences between the parameters in the different latent class models reflect parameter uncertainty. A graphical overview of the specified latent class model can be found in Fig. 2. First, the latent variable measuring vehicle type, X_1, is specified. The variables measuring vehicle type originating from

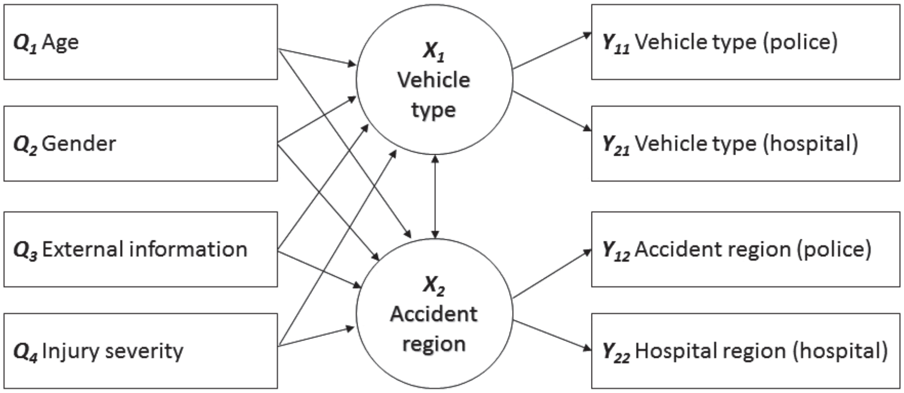

Q_1 Age
Q_2 Gender
Q_3 External information
Q_4 Injury severity
X_1 Vehicle type
X_2 Accident region
Y_{11} Vehicle type (police)
Y_{21} Vehicle type (hospital)
Y_{12} Accident region (police)
Y_{22} Hospital region (hospital)

Fig. 2. Graphical overview of the latent class model specified in Latent GOLD

the police registry, Y_{11}, and from the hospital registry, Y_{21}, are specified as indicators of this latent variable. Note that this notation differs from traditional notation where X-variables are predictors and Y-variables are responses, e.g. in regression analysis. As was discussed in Section 2, the vehicle type indicator variables contain nine categories in total: six representing motorized vehicles and three representing non-motorized vehicles. However, specifying nine latent classes would be problematic, since the number of observed non-motorized accidents in the police registry is very low. Therefore, the non-motorized categories are grouped into one category, resulting in the specification of a seven-class model. By saving the original scores of this indicator variable in separate variables, these can be reassigned to the accidents which were assigned to the latent class 'accidents without motorized vehicle' after multiple imputation. For this, the proportions of the categories in the observed data are used.

Second, all covariates of interest need to be included in the latent class, because otherwise point estimates describing the relationship between a latent variable and an excluded covariate will be biased (Bolck *et al.*, 2004). As discussed in Section 2, region of accident cannot be included directly as a covariate as it contains a large proportion of missing values. Therefore, multiple imputations are created for this variable to be able to stratify for vehicle type over the different regions in the Netherlands. For this purpose, a second latent variable is specified to measure region of accident, X_2. The first indicator is region of accident measured in the police registry, Y_{12}. The second indicator variable is region of hospital, Y_{22}. Since the first indicator is actually the variable for which imputations are created, the relationship between the latent variable and the indicator variable is restricted such that, if the indicator variable is observed, this score is assigned directly to the latent variable as well. Only if this indicator variable contains a missing value are the outcomes of this latent class model used.

Other covariates that are needed to make relevant stratifications can be included in the latent class model directly, since they do not contain any missing values. The other covariates that are included in the latent class model are

(a) age, 0–17, 18–44, 46–69 and 70 years or older (Q_1),
(b) gender, male or female (Q_2),
(c) external information, standard, falling, non-public road, no driving vehicle and other (Q_3), and
(d) injury severity by using the abbreviated injury scale, which is an anatomical scoring system where injuries are ranked on a scale from 1 to 6. As '1' represents 'minor injuries' and '6' represents 'unsurvivable injuries', these do not fit in the scope of this research, as

this research pertains to 'serious road injuries'. Therefore, the following scores on the abbreviated injury scale are included: '2' means 'moderate'; '3' means 'serious'; '4' means 'severe'; '5' means 'critical' (Q_4) (Wong, 2011).

To ensure that all parameters can be estimated for each bootstrap sample, only main effects of the covariates are included in the latent class model.

The latent class model for response pattern $P(\mathbf{Y}=\mathbf{y}|\mathbf{Q}=\mathbf{q})$ is

$$P(\mathbf{Y}=\mathbf{y}|\mathbf{Q}=\mathbf{q})=\sum_{x_1=1}^{7}\sum_{x_2=1}^{12}\prod_{l_1=1}^{2}P(Y_{l_1,1}=y_{l_1,1}|X_1=x_1)\prod_{l_2=1}^{2}P(Y_{l_2,2}=y_{l_2,2}|X_2=x_2)$$
$$\times P(X_1=x_1,X_2=x_2|\mathbf{Q}=\mathbf{q}). \qquad (1)$$

In this latent class model, X_1 represents the latent variable vehicle type with seven classes and X_2 represents the latent variable region of accident with 12 classes. Furthermore, $\mathbf{Q}$ represents the covariate variables and $\mathbf{Y}$ represents the indicator variables, where l_1 stands for the two indicator variables corresponding to X_1 and l_2 for the two indicator variables corresponding to X_2 (which corresponds to what can be seen in Fig. 2). The latent class model is estimated by using Latent GOLD 5.1 (Vermunt and Magidson, 2015), where the recommendations by Vermunt *et al.* (2008) for large data sets have been followed to ensure convergence. See Appendix A for the Latent GOLD syntax that was used.

By specifying the previously described latent class model, the first assumption made is that the probability of obtaining a specific response pattern is a weighted average of all conditional response probabilities, which is also known as the mixture assumption. Second, the assumption is made that the observed indicators are independent of each other given a unit's score on the underlying true measure. In other words, this means that, if a classification error is made in the police registry, we assume that this is independent of the probability of also having a classification error in the hospital registry. For most cases this assumption can be considered realistic, since the police registry and the hospital registry are generally filled out by two different and independent people. In rare situations, dependences might arise. For example, in a 'hit-and-run' situation, both registries will probably be filled out on the basis of information that is provided by the victim and are therefore not independent. Third, the assumption is made that the misclassification in the indicators is independent of the covariates. It is unlikely that scores on covariates such as age or gender will influence this. However, for example for the variable 'external information', it can be that, if an accident takes place outside the public road, it is more difficult for the police to reach this location and therefore the probability of an error can increase. Fourth, the assumption is made that the covariate variables are free of error. This is, of course, an unrealistic assumption, especially given the substantial amounts of classification error that is found in the vehicle type indicator variables. At this point we unfortunately do not have any information about the extent of possible classification errors in the other variables. However, these errors are considered less problematic as long as they are random. Lastly, assumptions are made with respect to the missingness mechanisms in the data. More specifically, the mechanism that governs the probability that each data point has of being missing is considered missing at random for the variables 'vehicle type observed in the police registry', Y_{11}, and region of accident, Y_{12}, as the probability of being missing is larger for 'non-motorized' vehicles, which is measured by the hospital registry, Y_{21}. Formally, it can be stated that Y_{11} consists of a part $Y_{11,\text{obs}}$ and $Y_{11,\text{mis}}$ and that a vector R_{11} can be defined:

$$R_{11}=\begin{cases}0 & \text{if } Y_{11,\text{obs}}, \qquad (2)\\ 1 & \text{if } Y_{11,\text{mis}}. \qquad (3)\end{cases}$$

Table 2. Cross-table between the variables region of hospital (columns) and region of accident (rows) for the years 1994, 2009 and 2013

Year		*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*Total*
1994	Missing value	345	419	213	627	772	499	1152	1140	123	997	590	111	6988
	1 Groningen	314	4	2	5	2	1	4	2	0	0	2	1	337
	2 Friesland	17	393	5	7	0	1	1	5	0	3	1	2	435
	3 Drenthe	57	3	230	14	1	1	1	3	0	1	0	0	311
	4 Overijssel	2	3	26	711	7	4	6	4	0	9	2	4	778
	5 Gelderland	2	0	3	112	977	108	10	23	1	46	4	3	1289
	6 Utrecht	3	3	1	2	52	564	38	7	2	6	3	1	682
	7 Noord-Holland	4	2	2	6	15	11	1538	29	1	14	9	4	1635
	8 Zuid-Holland	6	4	7	8	16	22	30	1564	4	20	8	2	1691
	9 Zeeland	0	0	0	0	2	0	1	9	212	24	0	0	248
	10 Noord-Brabant	1	2	1	5	60	6	17	35	2	1550	33	0	1712
	11 Limburg	0	2	1	1	19	2	5	3	1	12	690	1	737
	12 Flevoland	0	1	0	6	6	5	10	3	0	0	0	89	120
	Total	751	836	491	1504	1929	1224	2813	2827	346	2682	1342	218	16963
2009	Missing value	435	586	267	667	1523	865	2014	1728	151	1185	840	185	10446
	1 Groningen	186	5	2	2	3	0	3	1	0	4	1	0	207
	2 Friesland	23	200	3	3	0	1	1	0	0	0	0	0	231
	3 Drenthe	48	0	91	16	1	0	1	3	1	4	2	0	167
	4 Overijssel	2	2	3	265	2	0	2	5	0	1	1	0	283
	5 Gelderland	1	2	0	51	516	58	5	5	0	20	2	0	660
	6 Utrecht	1	2	0	3	26	323	23	1	1	2	0	0	382
	7 Noord-Holland	0	3	2	1	10	11	673	11	2	6	12	0	731
	8 Zuid-Holland	2	3	1	3	6	19	13	683	0	6	4	0	740
	9 Zeeland	0	0	0	0	0	0	2	3	80	8	1	0	94
	10 Noord-Brabant	1	0	0	0	23	4	9	14	0	491	6	0	548
	11 Limburg	0	0	0	1	7	0	4	1	1	3	300	1	318
	12 Flevoland	1	13	0	22	5	3	6	3	0	0	0	27	80
	Total	700	816	369	1034	2122	1284	2756	2458	236	1730	1169	213	14887

(*continued*)

Table 2 (*continued*)

Year		*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*Total*
2013	Missing value	392	534	372	857	1696	870	2643	2286	324	1475	815	154	12418
	1 Groningen	53	0	0	0	0	0	0	1	1	2	1	0	58
	2 Friesland	12	77	0	1	0	1	1	0	0	0	0	0	92
	3 Drenthe	18	0	36	8	1	1	0	0	0	0	1	0	65
	4 Overijssel	0	0	0	180	1	0	0	2	0	0	2	0	185
	5 Gelderland	0	0	0	37	313	30	2	2	0	5	2	0	391
	6 Utrecht	0	0	0	0	13	178	15	1	1	3	2	0	213
	7 Noord-Holland	2	0	0	1	3	7	492	3	0	0	1	1	510
	8 Zuid-Holland	1	0	1	2	4	8	14	439	1	6	1	2	479
	9 Zeeland	0	0	0	0	0	0	0	8	57	5	0	0	70
	10 Noord-Brabant	1	1	2	0	19	1	3	11	0	293	3	0	334
	11 Limburg	0	0	0	2	14	0	0	3	1	3	141	0	164
	12 Flevoland	1	2	0	10	2	2	15	0	0	0	0	22	54
	Total	480	614	411	1098	2066	1098	3185	2756	385	1792	969	179	15033

As we assume the missingness mechanism to be missingness at random, the distribution of missing values is related to Y_{21}:

$$P(R_{11}=0|Y_{11,\mathrm{obs}}, Y_{11,\mathrm{mis}}, Y_{21}) = P(R_{11}=0|Y_{11,\mathrm{obs}}, Y_{21}). \tag{4}$$

If a value is missing in Y_{11}, it is by definition missing in Y_{12} as well, as unit missingness is considered here and both variables originate from the same data set. Furthermore, the mechanism that governs the probability of being missing is considered missing completely at random for the variable 'vehicle type observed in the hospital registry', Y_{21}. Here,

$$P(R_{21}=0|Y_{21,\mathrm{obs}}, Y_{21,\mathrm{mis}}, Y_{11,\mathrm{obs}}) = P(R_{21}=0). \tag{5}$$

The distribution of missing values in Y_{11} and Y_{12} is related to Y_{21}, which in itself also contains missing values. Generally this would mean that we are not dealing with a missingness at random mechanism for Y_{11} and Y_{12}. However, because of the special structure of our data set in which Y_{11} and Y_{12} never contain missing values if Y_{21} contains missing values and vice versa, we are still dealing with a missingness at random mechanism. Cases containing missing values on all abovementioned variables are by definition not included in the data set and are treated separately.

The latent class model gives different forms of relevant output. The first form of relevant output is the entropy R^2. Entropy can be formally defined as

$$\mathrm{EN}(\alpha) = -\sum_{j=1}^{N}\sum_{x=1}^{X} \alpha_{jx}\log(\alpha_{jx}), \tag{6}$$

where α_{jx} is the probability that observation j is a member of class x, X the number of classes and N is the number of units in the combined data set. Rescaled to values between 0 and 1, entropy R^2 is measured by

$$R^2 = 1 - \frac{\mathrm{EN}(\alpha)}{N\log(X)}, \tag{7}$$

where 1 means perfect prediction (Dias and Vermunt, 2008). Boeschoten *et al.* (2017) showed that the performance of the MILC method is closely related to the entropy R^2 of the corresponding latent class model.

A second form of relevant output is the conditional response probabilities. They provide us with the probability of obtaining a specific response on the indicator conditionally on belonging to a certain latent class. These values can be used to investigate the relationships between the indicator variables and the latent variables in detail. For example, they show us the probability of having the score M–car on the indicator originating from the police registry given that the model assigned a case to the latent class M–car, but also the probability of having the score M–bicycle on the indicator given that the model assigned a case to the latent class M–car. Here, the former should be much higher compared with the latter. By comparing the conditional response probabilities with the cross-table between the variables measuring vehicle type originating from the police registry and the hospital registry (as seen in Table 1), it can be investigated whether the latent classes that are identified as certain categories of vehicle type are related to other categories in the indicator variables in a comparable way with that in the observed data. In this way, it is checked whether the latent class model reflects the main relationships that are found in the observed data, which is an important indication of adequate imputations in the next step.

Third, the posterior membership probabilities represent the probability that a unit belongs to a latent class given its combination of scores on the indicators and covariates that are used in the latent class model. These values are used to create multiple imputations for the latent variables, and the exact procedure for this is described in the next section.

3.3. Creating multiple imputations

The posterior membership probabilities are used to create multiple imputations of the latent variables containing the true scores. The posterior membership probabilities can be estimated by applying the Bayes rule to the latent class model that is described in equation (1):

$$P(X_1=x_1, X_2=x_2|\mathbf{Y}=\mathbf{y}, \mathbf{Q}=\mathbf{q}) = \frac{P(X_1=x_1, X_2=x_2, \mathbf{Y}=\mathbf{y}|\mathbf{Q}=\mathbf{q})}{P(\mathbf{Y}=\mathbf{y}|\mathbf{Q}=\mathbf{q})}, \tag{8}$$

where

$$\begin{aligned} P(X_1=x_1, X_2=x_2, \mathbf{Y}=\mathbf{y}|\mathbf{Q}=\mathbf{q}) = {} & \prod_{l_1=1}^{2} P(Y_{l_1,1}=y_{l_1,1}|X_1=x_1) \prod_{l_2=1}^{2} P(Y_{l_2,2}=y_{l_2,2}|X_2=x_2) \\ & \times P(X_1=x_1, X_2=x_2|\mathbf{Q}=\mathbf{q}), \end{aligned} \tag{9}$$

and $P(\mathbf{Y}=\mathbf{y}|\mathbf{Q}=\mathbf{q})$ is defined in equation (1).

Since two latent variables are specified in this model, the joint posterior membership probabilities are obtained which represent the probability that a unit is a member of a specific latent class in the latent variable vehicle type, and a member of a specific latent class in the latent variable region of accident. Since vehicle type has seven classes and region of accident has 12 classes, there are 84 posterior membership probabilities which add up to 1, and there is a different set of posterior membership probabilities for each combination of scores on the indicators and covariates. Parameter estimation was constrained in such a way that, if a case had an observed score on region of accident in the police registry, this score is automatically assigned to the latent variable as well. In those cases, there are only seven posterior membership probabilities with a value larger than 0 (those representing the different classes for vehicle type in combination with that specific region); all other posterior membership probabilities are exactly 0.

For each case in the original data set, the posterior membership probabilities corresponding to its combination of scores on the indicators and covariates are used as a multinomial distribution to draw a joint score on both latent variables. These joint scores are then used to create separate imputations for vehicle type and region of accident.

By drawing multiple times from the posterior membership probabilities, multiple imputations for both latent variables are created. The scores that are assigned to the latent variables can be different for the different imputations. The differences between them reflect the uncertainty due to the missing and conflicting values in the indicator variables. Boeschoten *et al.* (2017) concluded that a low number of imputations, such as 5, is already sufficient for a correct estimation of the standard errors. However, in that simulation study the number of classes was much lower compared with the number of classes that is needed for this data set. To evaluate what the appropriate number of imputations would be, the number of imputations was gradually increased and the fraction of missing information was compared between the differing numbers of imputations (Graham *et al.*, 2007), resulting in 20 imputations. This is in line with the recommendations by Wang *et al.* (2005).

3.4. Pooling of the results

At this point, 20 imputations are created for vehicle type and region of accident for every unit in the combined data set. The goal is to obtain estimates of interest by using these imputed variables. This is done by obtaining the estimate of interest for every imputed variable, and pooling these estimates by using the pooling rules that were defined by Rubin (Rubin (1987), page 76). Although our context differs from the traditional statistical context for which the pooling rules were originally developed, the rules are considered appropriate for the context of multiple imputation for measurement error (Reiter and Raghunathan, 2007). For this specific research, the main estimates of interest are frequency tables.

The first step is to calculate a pooled frequency table. In other words, we take the average over the imputations for every cell in the frequency table. This can be for the imputed variable vehicle type, for the imputed variable region of accident or for a cross-table between (one of) these variables and covariate(s). A pooled cell count is obtained by

$$\hat{\theta}_j = \frac{1}{m} \sum_{i=1}^{m} \hat{\theta}_{ij}, \tag{10}$$

where θ refers to a cell count, j refers to a specific cell in the frequency table, i refers to one imputation and m refers to the total number of imputations.

Next, an estimate of the uncertainty around these frequencies is of interest. Therefore, the pooled frequencies need to be transformed into pooled proportions:

$$\hat{p}_j = \frac{(1/m) \sum_{i=1}^{m} \hat{\theta}_{ij}}{\sum_{j=1}^{s} (1/m) \sum_{i=1}^{m} \hat{\theta}_{ij}}, \tag{11}$$

where s refers to the number of cells in the frequency table.

Since we work with a multiply imputed data set, an estimate of the variance is obtained that is a combination of sampling uncertainty and uncertainty due to missing and conflicting values in the data set. This is the total variance that consists of a 'within-imputation' and 'between-imputation' component:

$$\mathrm{VAR}_{\mathrm{total}_j} = \overline{\mathrm{VAR}}_{\mathrm{within}_j} + \mathrm{VAR}_{\mathrm{between}_j} + \frac{\mathrm{VAR}_{\mathrm{between}_j}}{m}. \tag{12}$$

$\overline{\mathrm{VAR}}_{\mathrm{within}_j}$ is the within-imputation variance of $\hat{p}_j$ calculated by

$$\overline{\mathrm{VAR}}_{\mathrm{within}_j} = \frac{1}{m} \sum_{i=1}^{m} \mathrm{VAR}_{\mathrm{within}_{ij}}, \tag{13}$$

where $\mathrm{VAR}_{\mathrm{within}_{ij}}$ is estimated as the variance of $\hat{p}_{ij}$:

$$\frac{\hat{p}_{ij}(1-\hat{p}_{ij})}{N}, \tag{14}$$

where N is the total size of the observed data set and $\hat{p}_{ij}$ is estimated as

$$\hat{p}_{ij} = \frac{\hat{\theta}_{ij}}{\sum_{j=1}^{s} \hat{\theta}_{ij}}. \tag{15}$$

$\mathrm{VAR}_{\mathrm{between}_j}$ is calculated by

$$\mathrm{VAR}_{\mathrm{between}_j} = \frac{1}{m-1} \sum_{i=1}^{m} (\hat{p}_{ij} - \hat{p}_j)(\hat{p}_{ij} - \hat{p}_j)'. \tag{16}$$

When $\mathrm{VAR}_{\mathrm{total}_j}$ is estimated, it can be used to estimate the standard error of $\hat{p}_j$:

$$\mathrm{SE}(\hat{p}_j) = \surd \mathrm{VAR}_{\mathrm{total}_j}. \tag{17}$$

From here, the confidence interval around $\hat{p}_j$ can be estimated by

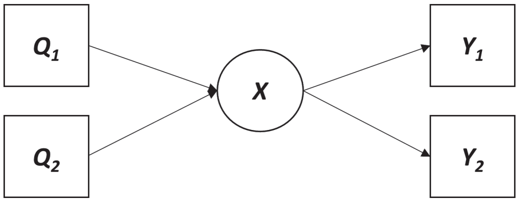


Fig. 3. Graphical overview of the latent class model used for the simulation study

$$\hat{p}_j \pm Z_{0.975}\,\mathrm{SE}(P_j), \tag{18}$$

where 0.975 corresponds to the $(1-\alpha/2)$-quantile of a standard normal distribution for $\alpha = 0.05$. The values that are obtained here can simply be multiplied by N to obtain the 95% confidence intervals around the observed frequencies $\hat{\theta}_j$. Note that a standard normal distribution is assumed so problems can be encountered when dealing with very small proportions.

3.5. Performance of the multiple imputation of latent classes method

Boeschoten *et al.* (2017) introduced the MILC method and evaluated the method under a range of conditions in terms of data quality. In addition, Boeschoten *et al.* (2018b) extended the method for situations with longitudinal data and Boeschoten *et al.* (2018a) extended the method in such a way that covariates can be included at later time points. All research performed on the MILC method so far has shown a strong relationship between the performance of the method and the entropy R^2-value of the latent class model. To investigate how the MILC method performs in comparison with the hierarchical assignment procedure that is traditionally used by the SWOV (Bos *et al.*, 2017), an illustrative simulation study is performed using a latent class model as shown in Fig. 3.

In the theoretical population that is used for this simulation study, latent variable X has two categories with probabilities 0.6 for $X = 1$ and 0.4 for $X = 2$. The probability distribution of $P(X, Q_1)$ is

$$\begin{array}{c c} & \begin{array}{cc} Q_1 = 1 & Q_1 = 2 \end{array} \\ \begin{array}{c} X = 1 \\ X = 2 \end{array} & \begin{pmatrix} 0.48 & 0.12 \\ 0.32 & 0.08 \end{pmatrix} \end{array} \tag{19}$$

and the probability distribution of $P(X, Q_2)$ is

$$\begin{array}{c c} & \begin{array}{ccc} Q_2 = 1 & Q_2 = 2 & Q_2 = 3 \end{array} \\ \begin{array}{c} X = 1 \\ X = 2 \end{array} & \begin{pmatrix} 0.36 & 0.18 & 0.06 \\ 0.24 & 0.12 & 0.04 \end{pmatrix} \end{array} \tag{20}$$

From this population structure, 1000 samples are drawn. In each sample, indicator Y_1 of X is created with 5% misclassification and a missingness at random mechanism, where the probability of being missing is related to a person's score on the Q_2-covariate:

$$Q_2 = 1,\ P(Y_1 = \mathrm{NA}) = 0.20; \tag{21}$$

$$Q_2 = 2,\ P(Y_1 = \mathrm{NA}) = 0.15; \tag{22}$$

$$Q_2 = 3,\ P(Y_1 = \mathrm{NA}) = 0.10. \tag{23}$$

Table 3. Results of a simulation study where the hierarchical assignment procedure is compared with the MILC method, which is performed with and without a non-parametric bootstrap†

	Bias	*Coverage*	*Confidence interval width*	*se/sd*	*RMSE*
Hierarchical assignment					
$W_{\text{ass}}=1$	−0.0134	0.2180	0.0193	0.9981	0.0143
$W_{\text{ass}}=2$	0.0134	0.2180	0.0193	0.9981	0.0143
$W_{\text{ass}}=1\times Q_1=1$	−0.0106	0.4220	0.0196	0.9894	0.0117
$W_{\text{ass}}=2\times Q_1=1$	−0.0028	0.8380	0.0126	0.9964	0.0043
$W_{\text{ass}}=1\times Q_1=2$	0.0107	0.3590	0.0184	0.9963	0.0117
$W_{\text{ass}}=2\times Q_1=2$	0.0027	0.8380	0.0108	1.0191	0.0038
$W_{\text{ass}}=1\times Q_2=1$	0.0012	0.3560	0.0134	0.9433	0.0052
$W_{\text{ass}}=2\times Q_2=1$	−0.1676	0.6390	0.0107	1.0053	0.1677
$W_{\text{ass}}=1\times Q_2=2$	−0.2910	0.7770	0.0066	0.9898	0.2910
$W_{\text{ass}}=2\times Q_2=2$	−0.1115	0.3050	0.0121	0.9702	0.1116
$W_{\text{ass}}=1\times Q_2=3$	−0.2261	0.5920	0.0092	1.0201	0.2261
$W_{\text{ass}}=2\times Q_2=3$	−0.3022	0.7990	0.0056	1.0552	0.3022
MILC method, bootstrap excluded					
$W=1$	−0.0317	0.1300	0.0216	0.1425	0.042
$W=2$	0.0317	0.1300	0.0216	0.1425	0.042
$W=1\times Q_1=1$	−0.0252	0.1660	0.0213	0.1751	0.0335
$W=2\times Q_1=1$	−0.0066	0.4410	0.0132	0.3912	0.0093
$W=1\times Q_1=2$	0.0253	0.1660	0.0205	0.1683	0.0336
$W=2\times Q_1=2$	0.0064	0.3980	0.0118	0.3628	0.0089
$W=1\times Q_2=1$	−0.0191	0.2270	0.0201	0.2151	0.0257
$W=2\times Q_2=1$	−0.0095	0.3470	0.0157	0.3278	0.0131
$W=1\times Q_2=2$	−0.0031	0.5820	0.0096	0.5341	0.0048
$W=2\times Q_2=2$	0.0191	0.1980	0.0188	0.2029	0.0255
$W=1\times Q_2=3$	0.0095	0.3150	0.0142	0.2980	0.0131
$W=2\times Q_2=3$	0.0031	0.5510	0.0085	0.4962	0.0046
MILC method including bootstrap					
$W=1$	−0.0304	0.8880	0.1790	1.5797	0.0420
$W=2$	0.0304	0.8880	0.1790	1.5797	0.0420
$W=1\times Q_1=1$	−0.0241	0.8950	0.1439	1.5811	0.0335
$W=2\times Q_1=1$	−0.0063	0.9050	0.0383	1.4324	0.0093
$W=1\times Q_1=2$	0.0243	0.8940	0.1437	1.5744	0.0336
$W=2\times Q_1=2$	0.0062	0.9160	0.0378	1.4887	0.0089
$W=1\times Q_2=1$	−0.0183	0.8880	0.1087	1.5375	0.0257
$W=2\times Q_2=1$	−0.0091	0.9020	0.0560	1.5192	0.0131
$W=1\times Q_2=2$	−0.0030	0.9290	0.0205	1.4125	0.0048
$W=2\times Q_2=2$	0.0183	0.8910	0.1085	1.5562	0.0255
$W=1\times Q_2=3$	0.0092	0.9050	0.0555	1.5085	0.0131
$W=2\times Q_2=3$	0.0030	0.9280	0.0200	1.4670	0.0046

†Results are shown for the imputed mixture variable, denoted by W, and of the relationship of W with covariates Q_1 and Q_2. Results are given in terms of bias, coverage of the 95% confidence interval, confidence interval width, the average standard error se of the estimate divided by the standard deviation sd over the estimates and the root-mean-squared error RMSE.

A second indicator Y_2 of X is created with 15% misclassification and 5% missing cases which are missing completely at random. The latent class models had an entropy R^2-value of approximately 0.75.

The MILC method as described in Sections 3.1, 3.2 and 3.3 is applied to the sample data sets, where five bootstrap samples are drawn and subsequently five imputations of X are created. As an illustration, the MILC method is also applied without the bootstrap procedure, with one

Table 4. Entropy R^2-values for the latent variables vehicle type and region of accident for the years 1994, 2009 and 2013

Year	*Vehicle type*	*Region of accident*
1994	0.8219	0.9050
2009	0.7444	0.8267
2013	0.8031	0.8077

latent class model directly estimated on the observed data and five imputations drawn from one single set of posterior membership probabilities. Furthermore, the hierarchical assignment procedure as used by the SWOV is also applied. At the SWOV, the score that is observed in the police registry, Y_1, is assigned if it is observed. Otherwise, the score that is observed in the hospital registry, Y_2, is assigned.

The imputations are evaluated in terms of bias, coverage of the 95% confidence interval, confidence interval width, average standard error of the estimates divided by the standard deviation over the estimates and the root-mean-squared error RMSE. Furthermore, the proportion of correctly classified cases is evaluated for imputation and hierarchical assignment.

To evaluate the methods, the marginals of the imputed latent variable W are compared with the hierarchically assigned variable W_{ass}. In addition the estimated relationships of the latent variable with covariates, $W \times Q_1$, $W_{\text{ass}} \times Q_1$, $W \times Q_2$ and $W_{\text{ass}} \times Q_2$, are examined.

In Table 3 the results of the simulation study comparing the MILC method (with and without the bootstrap) and the hierarchical assignment procedure are shown. We first discuss the performance of the MILC method in comparison with the hierarchical assignment method. The results that were obtained with hierarchical assignment especially show substantial amounts of bias for $W_{\text{ass}} \times Q_2$ compared with both implementations of the MILC method. For the unbiased parameters that were obtained when applying hierarchical assignment, RMSE is in general lower and more stable compared with the RMSE of the MILC method. The fact that, with hierarchical assignment, bias is especially found in the results relating to Q_2 can be explained by the fact that the missingness mechanism of Y_1 is defined by Q_2.

A comparison of the MILC method with and without the bootstrap shows clearly that standard errors are very much underestimated when no bootstraps are performed, i.e. coverage rates are too low and the ratios between the average standard error and the standard deviation across replications are far below 1. In contrast, these ratios are larger than 1 when the bootstrap is included in the MILC method, meaning that the standard errors are somewhat overestimated. The large difference between the two approaches is caused by the fact that the statistics that we are interested in are tables containing the latent variable X. By not applying the bootstrap, one seriously underestimates the uncertainty about the latent class proportions. The fact that the bootstrap procedure yields slightly too large standard errors can be considered to be less problematic than having (much) too small standard errors.

The percentage of incorrectly classified cases is 4.5% for $X=1$ and 10.1% for $X=2$ when hierarchical assignment is applied (these results are not shown in Table 3). When the MILC method (including the bootstrap) is applied, the percentage of incorrectly classified cases is 8.6% for $X=1$ and 20.5% for $X=2$. With hierarchical assignment, the score on one indicator variable is used per case, and the misclassification corresponds to the misclassification that is specified in these variables. When the MILC method is applied, two indicator variables are used

Table 5. Class-specific response probabilities for latent variable vehicle type for the years 1994, 2009 and 2013

Vehicle type	*Results for 1994*		*Results for 2009*		*Results for 2013*	
	Hospital	*Police*	*Hospital*	*Police*	*Hospital*	*Police*
1 M–car	0.8226	0.9782	0.8004	0.9742	0.9590	0.8973
2 M–moped	0.8458	0.9781	0.7194	0.9786	0.9693	0.8848
3 M–bicycle	0.7393	0.9170	0.7635	0.9620	0.9263	0.7376
4 M–motorcycle	0.8353	0.9686	0.8876	0.9129	0.0774	0.7577
5 M–other	0.6890	0.0578	0.5276	0.2629	0.0000	0.4243
6 M–pedestrian	0.7132	0.8213	0.8758	0.8104	0.5358	0.6412
7 N–all	0.9920	0.6162	0.9916	0.5273	0.9931	0.3897

Table 6. Class-specific response probabilities for latent variable region of accident for the years 1994, 2009 and 2013

Region of accident	*Results for 1994*		*Results for 2009*		*Results for 2013*	
	Region of hospital	*Region of accident*	*Region of hospital*	*Region of accident*	*Region of hospital*	*Region of accident*
1 Groningen	0.9351	1	0.8798	1	0.9167	1
2 Friesland	0.9063	1	0.8740	1	0.8433	1
3 Drenthe	0.7338	1	0.5897	1	0.6556	1
4 Overijssel	0.9103	1	0.9290	1	0.9675	1
5 Gelderland	0.7551	1	0.7961	1	0.8119	1
6 Utrecht	0.8292	1	0.8259	1	0.8149	1
7 Noord-Holland	0.9378	1	0.9267	1	0.9673	1
8 Zuid-Holland	0.9240	1	0.9248	1	0.9094	1
9 Zeeland	0.8506	1	0.8248	1	0.7941	1
10 Noord-Brabant	0.9084	1	0.9055	1	0.8884	1
11 Limburg	0.9397	1	0.9466	1	0.8725	1
12 Flevoland	0.7771	1	0.5374	1	0.4694	1

to generate the variables that are under evaluation here. When assigning scores, maintaining the relationships with other variables is apparently considered more important than correctly classifying individual cases. Including interaction terms in the latent class model may possibly lead to more accurate results for the MILC method. Whether this really is so remains to be examined, though.

4. Results

First, results in terms of relevant model output will be discussed. Second, substantial results that were obtained after creating multiple imputations for the latent variables are given.

4.1. Latent class model output

The first relevant model output from the latent class models comes in terms of the entropy R^2. A separate entropy R^2-value is estimated for the two latent variables and for each year. The results are shown in Table 4. These results are obtained after applying a latent class model on the original data set. Here it can be seen that the entropy R^2-value in 2013 increased compared with 2009 for vehicle type. Pankowska *et al.* (2017) showed in their simulation studies that, when a latent class model is used to correct for misclassification in combined data sets, the model also treats inconsistencies due to incorrect linkage as misclassification and thereby corrects for it in a similar way. This implies that the increase in terms of entropy R^2 in 2013 in comparison with 2009 for the latent variable vehicle type makes sense as the police improved their registration system in 2013. This improvement caused an increase in the number of correctly linked cases and therefore also improved the entropy R^2. The higher entropy R^2-values that were found for 1994 are likely to be caused by the fact that registration was performed more carefully and thoroughly by the police at that period, which also resulted in the lower amount of missing values, as can be seen in Table 1.

In Tables 5 and 6 the probability of correct classification for the indicators of both latent variables are shown, for the three different time points, obtained after applying a latent class model to the original data set. Class-specific response probabilities indicate the probability of having a score on the indicator variable that is equal to the latent class. A high probability of correct classification indicates that, when a specific case belongs to a certain latent class, the probability is large that this same score was obtained on an indicator variable. For example, the probability of correct classification of the 1994 indicator variable ‘hospital’ for the latent class ‘vehicle type ≡ M–car’ is 0.8226. This means that the probability of having scored M–car on the indicator variable ‘vehicle type measured by hospital’ is 0.8226 given that this case truly belongs to the latent class M–car.

When looking at the probabilities of correct classification for a specific latent class, the two probabilities corresponding to the two indicators are often not equal. This may be due to differences in the quality of the data. A low probability of correct classification can be caused by the fact that, for this specific latent class, this category is observed many times in one indicator (here this is often the indicator hospital), whereas, in the other indicator (‘police’), these cases are often missing. This can clearly be seen for the latent class N–all. Conditionally on truly belonging in this latent class, the probability of obtaining this score on the hospital indicator was 0.9920 in 1994. In other words, almost everyone who is assigned to this class by the model obtained this score in the hospital registry as well. However, the probability of obtaining this score by the police is only 0.6162. A substantial part of the cases belonging to this latent class obtained another score or no score at all by the police.

In general, it can be seen that the probabilities of correct classification for the police indicator in 1994 and 2009 are larger compared with the hospital indicator for all motorized classes except the class M–other and the ‘all non-motorized’ category. However, in 2013 all probabilities of correct classification are higher for the hospital indicator compared with the police indicator. This result might be related to the improvement in the linking in 2013. An exception is the category M–motorcycle, which is the only category with a probability of correct classification below 0.90 in the hospital registry. This is caused by the fact that some of the hospitals used a different registration system, that categorizes both motorcycles and mopeds in the motorcycle category.

When investigating the probabilities of correct classification for the latent variable region of accident, it can be seen that they are all exactly 1 for the indicator variable region of accident. Conditionally on being in a specific class in the latent variable region of accident, the probability of obtaining the same score on the indicator variable region of accident is 1. This restriction was imposed on the latent class model. The probabilities of correct classification of the indicator variable region of hospital now show us the probability that, conditionally on an accident truly happening in a specific region, what is the probability of also going to a hospital in that same

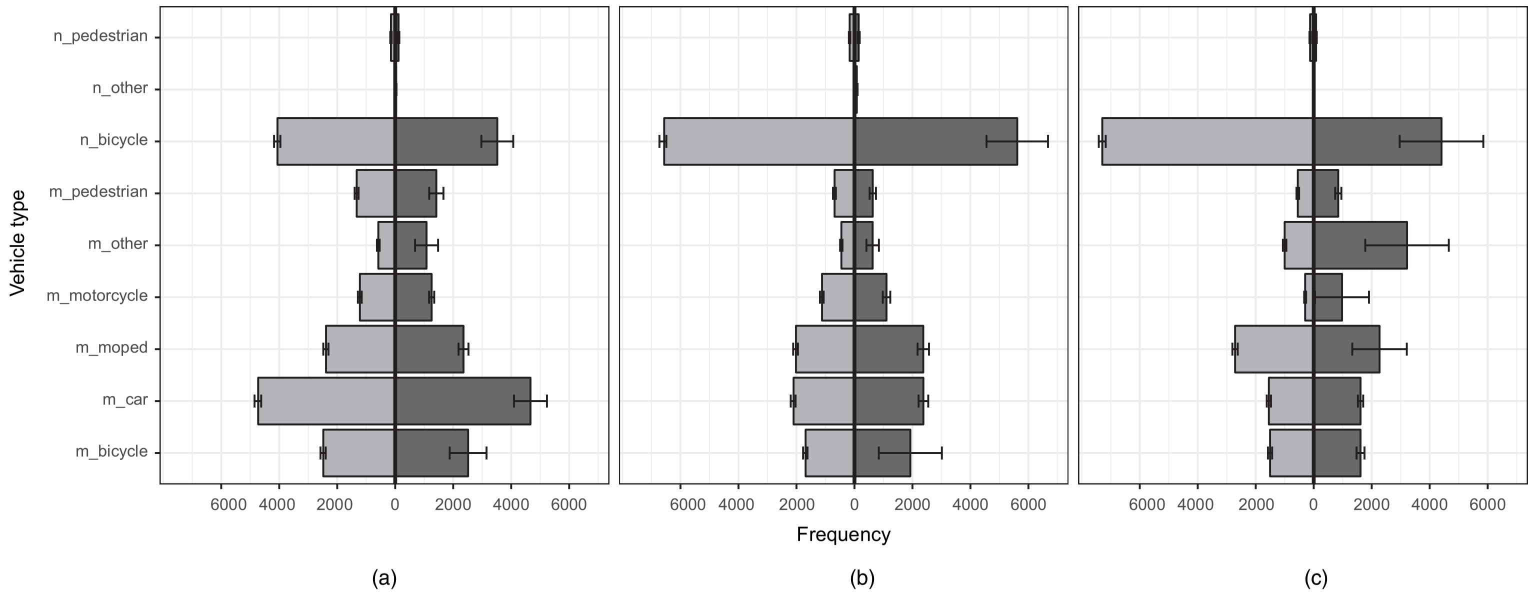


Fig. 4. Results obtained for (a) 1994, (b) 2009 and (c) 2013: on the left-hand side of each graph, the number of serious road injuries per vehicle type and corresponding 95% confidence intervals are shown when the hierarchical assignment procedure is applied (▨); on the right-hand side of each graph, pooled frequencies and 95% confidence intervals are shown when the extended MILC method is applied (■); note that the results that are presented in this paper by using the hierarchical assignment procedure are not necessarily exactly equal to official statistics produced by the SWOV

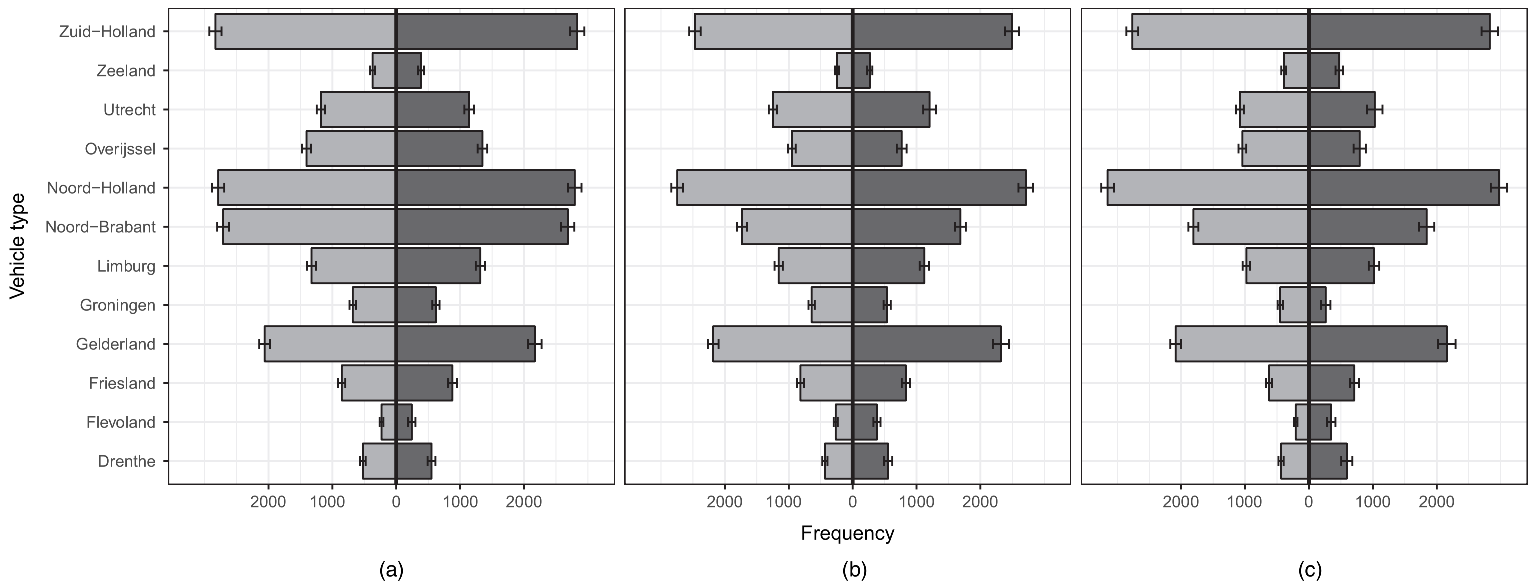


Fig. 5. Results obtained for (a) 1994, (b) 2009 and (c) 2013: on the left-hand side of each graph, frequencies of serious road injuries per region and corresponding 95% confidence intervals are shown when the hierarchical assignment procedure is applied (▨); on the right-hand side of each graph, pooled frequencies and 95% confidence intervals are shown when the extended MILC method is applied (■)

region? These probabilities are generally quite high and stable over the different time points. The regions Drenthe and Flevoland stand out because the probability of going to a hospital in these regions when having a serious road accident in this region is somewhat lower compared with other regions.

4.2. Pooled results output

In Fig. 4, the number of serious road injuries per vehicle type is shown for the three years that were investigated. For every year, the results that were obtained after applying the hierarchical assignment procedure are compared with results obtained when the extended MILC method is applied. Here, it can be seen that in general the frequencies that are obtained after applying the extended MILC method are quite similar compared with the results that are obtained after applying the hierarchical assignment procedure. When the extended MILC method is applied, the number of cases that are assigned to the category M–other is larger whereas the number of cases that were assigned to the category N–bicycle is smaller compared with the hierarchical assignment procedure, particularly in 2013. This corresponds to a large amount of missing cases for N–bicycle and a substantial amount of cases differently categorized by the police and hospital. Furthermore, in 2013 the number of cases categorized as M–other by the hospital increased, whereas this category was often classified differently by the police (see Table 1). At last, it can be seen that the widths of the 95% confidence intervals are substantially larger for all categories when the extended MILC method is applied. This directly results from the misclassification between the hospital and the police registry. Because of this misclassification, the latent class model is less certain about which value to assign to a specific case, resulting in differences between imputations and a larger estimate of the total variance. Note also that hierarchical assignment assumes that values that are observed in the police register are error free. Since this assumption is unlikely to be correct, uncertainty in the hierarchical assignment procedure is underestimated.

In Fig. 5, the number of serious road injuries per region is shown for the three years that were investigated. For every year, the results that were obtained after applying the hierarchical assignment procedure are compared with results obtained when the extended MILC method was applied, which are very similar. The 95% confidence intervals are larger when the extended MILC method was applied compared with the hierarchical assignment procedure, but the difference is not as substantial as for vehicle type in Fig. 4.

5. Discussion

In this paper, an extension of the MILC method was developed and applied to estimate the number of serious road injuries per vehicle type and to stratify this number in relevant subgroups. Information on serious road injuries was found in registries from both police and hospitals, which are both incomplete and contain misclassification. These variables were used as indicators of a latent variable of which it can be said that it contains the true scores. Posterior membership probabilities that were obtained from this latent class model were then used to create multiple imputations of these true scores. Simultaneously, multiple imputations were created for the missing values in region of accident by using this variable as a perfectly measured indicator of the latent variable region of accident and supplementing it by specifying region of hospital as an imperfectly measured indicator.

Multiple imputations were created for vehicle type and for region of accident. All variables are now fully imputed for every case in the data set. Descriptive statistics of these variables, or estimates of relationships with other variables, can now be investigated in a straightforward manner.

The extended MILC method was applied on data sets for the years 1994, 2009 and 2013. The quality of the data for these years was very different, which can be seen in the number of observations per registry per year and which is reflected in the entropy R^2 of the corresponding latent class model. In general the quality of the data was sufficient for applying the MILC method. The results of the extended MILC method were compared with the results that were obtained when the hierarchical assignment procedure was applied (traditionally used to generate these statistics). A clear difference was that the extended MILC method generated wider 95% confidence interval widths. Based on the results that were obtained from the simulation study performed in Section 3.5, it can be concluded that these wider confidence interval widths were indeed necessary to obtain nominal coverage rates.

Some issues are worth reflecting on a little further. First, it is important to note that our results heavily depend on the model assumptions that are made. In particular, the assumption is made that the classification errors are independent of covariates. Furthermore, the assumption is made that the covariate variables are free of error. Violating this assumption does not necessarily have to be an issue if these errors are random. However, there is currently no literature on this topic, so more research in this specific area is needed to be able to adapt the model. A more crucial assumption is that the missingness is at random. Although from a theoretical perspective this assumption is likely to hold, it could, however, lead to substantial bias in cases where this assumption is violated.

A second issue is how the extended MILC method dealt with non-motorized vehicles. This was an *ad hoc* procedure to handle an issue that could not be handled by the latent class model. This *ad hoc* procedure turned out to be useful. It can be investigated whether a comparable procedure could be applied to handle a moped or motorcycle issue in the 2013 data set and whether there are other issues that can be solved like this.

This particular data set contained several issues, of which a substantial part has been investigated by means of a simulation study. The results of this simulation study made clear that the extended MILC method could handle the missing values in the indicator variables and that the non-parametric bootstrap was required to obtain nominal coverage rates. It is, however, not investigated whether and how large numbers of categories influence the results. Therefore, the number of imputations was increased and evaluated by using methods to evaluate the number of imputations for missing values. A more thorough investigation could provide insight into whether these methods are suitable to evaluate the number of imputations that are needed when the MILC method is applied, and how many imputations are needed to evaluate data sets with larger numbers of categories.

Furthermore, in the initial model that was proposed by Boeschoten *et al.* (2017), bootstrap samples were taken of the original data to incorporate parameter uncertainty in the estimate of the total variance. This appeared to be problematic for larger models with many interactions than those used in our application, because not all parameters can be estimated for every bootstrap sample. Alternatives to incorporate parameter uncertainty can be Bayesian Markov chain Monte Carlo sampling or a parametric bootstrap. However, it should also be investigated whether such a step is still necessary for larger sample sizes as parameter uncertainty can become minimal in such cases. As the simulation study showed that it was necessary to incorporate parameter uncertainty when creating imputations for this specific case, a model with only main effects was used to enable estimation of all parameters.

Lastly, it is important to note that missing values in the combined data set and classification errors in the observed data are not the only issues when estimating the total number of serious road injuries per vehicle type. There are also serious road injuries that are neither observed by the hospital nor by the police. Weighting and capture–recapture methods are typically used to obtain an estimate of the total number of serious road injuries; approaches which can easily be combined with the MILC method by applying the methods on the imputations separately. A variance estimate would then include uncertainty about the total number of injuries which is typically estimated by making use of bootstrapping. This can also be applied separately to every imputation before pooling of the results is applied (Gerritse *et al.*, 2016).

By creating multiple imputations using a latent class model, multiply imputed versions of variables that contained missing values and/or classification errors are created. These can be used to provide frequencies easily, to divide these frequencies further into relevant subgroups or to create statistical figures. This application showed that the initial MILC method can be extended to handle problems that are data set specific. Furthermore, this application highlighted various new problems that one may need to deal with when applying the MILC approach. In future research, these will be investigated more thoroughly to exploit the potential of the MILC method fully for dealing with classification error problems.

Acknowledgements

The authors thank Niels Bos and Jacques Commandeur from the Institute of Road Safety Research for providing us with the data and for their useful comments on earlier versions of this paper.

Appendix A: Latent GOLD syntax

```
options
   maxthreads=all;
  algorithm
   tolerance=1e-008 emtolerance=0.01 emiterations=20000  nriterations=0;
  startvalues
    seed=0 sets=200 tolerance=1e-005 iterations=500;
  bayes
    categorical=1 variances=1 latent=1 poisson=1;
  missing
    includeall;
  output
    profile;
  outfile
    'posteriors1.dat' classification
  keep
    LRM2, BRON2, wfactor;
  variables
    caseweight b1;
    dependent LRM nominal 7, BRON nominal 7, prov_hosp nominal 12, prov_acc
      nominal 12;
    independent ernst nominal, external nominal, gender nominal, age
      nominal;
    latent X nominal 7, Xacc nominal 12;
  equations
    LRM       <- 1 | X;
    BRON      <- 1 | X;
    prov_acc  <- (a~wei)Xacc;
    prov_hosp <- 1 | Xacc;
    X         <- 1 | ernst + external + gender + age;
    Xacc      <- 1 | ernst + external + gender + age;
    X <-> Xacc;
```

```
a={1 0 0 0 0 0 0 0 0 0 0 0
   0 1 0 0 0 0 0 0 0 0 0 0
   0 0 1 0 0 0 0 0 0 0 0 0
   0 0 0 1 0 0 0 0 0 0 0 0
   0 0 0 0 1 0 0 0 0 0 0 0
   0 0 0 0 0 1 0 0 0 0 0 0
   0 0 0 0 0 0 1 0 0 0 0 0
   0 0 0 0 0 0 0 1 0 0 0 0
   0 0 0 0 0 0 0 0 1 0 0 0
   0 0 0 0 0 0 0 0 0 1 0 0
   0 0 0 0 0 0 0 0 0 0 1 0
   0 0 0 0 0 0 0 0 0 0 0 1};
```

To ensure convergence and to minimize the probability of obtaining local maxima, the number of random start sets is set to 200 with 500 iterations each. The use of Newton–Raphson iterations is suppressed and the number of expectation–maximization iterations is increased to 20000, following the suggestions by Vermunt *et al.* (2008).

To reduce computational time, the storing of parameters and the computation of standard errors is suppressed, since conditional and posterior response probabilities are of main interest.

To ensure that in the latent variable region of accident (`Xacc` in the Latent GOLD syntax) the value that is observed in the indicator variable region of accident (`prov_acc` in the Latent GOLD syntax) is assigned in cases where this variable is observed, the relationship between `Xacc` and `prov_acc` is restricted by using the matrix denoted by '`a`' in the Latent GOLD syntax.

References

Boeschoten, L., Oberski, D. and de Waal, T. (2017) Estimating classification errors under edit restrictions in composite survey-register data using multiple imputation latent class modelling (MILC). *J. Off. Statist.*, **33**, 921–962.

Boeschoten, L., Oberski, D. L., Waal, T. D. and Vermunt, J. K. (2018a) Updating latent class imputations with external auxiliary variables. *Structrl Equn Modlng*, **25**, 750–761.

Boeschoten, L., Varriale, R. and Filipponi, D. (2018b) Combining multiple imputation and hidden Markov modeling to obtain consistent estimates of 'true employment status'. *Unpublished*. Tilburg University, Tilburg.

Bolck, A., Croon, M. and Hagenaars, J. (2004) Estimating latent structure models with categorical variables: one-step versus three-step estimators. *Polit. Anal.*, **12**, 3–27.

Bos, N., Stipdonk, H. and Commandeur, J. (2017) Ernstig verkeersgewonden 2016. Instituut voor Wetenschappelijk Onderzoek Verkeersveiligheid, The Hague. (Available from `https://www.swov.nl/publicatie/ernstig-verkeersgewonden-2016`.)

Dias, J. G. and Vermunt, J. K. (2008) A bootstrap-based aggregate classifier for model-based clustering. *Computnl Statist.*, **23**, 643–659.

Gerritse, S. C., Bakker, B. F., de Wolf, P.-P. and van der Heijden, P. G. (2016) Undercoverage of the population register in the Netherlands, 2010. *Discussion Paper*. Centraal Bureau voor de Statistiek, The Hague.

Graham, J. W., Olchowski, A. E. and Gilreath, T. D. (2007) How many imputations are really needed?: Some practical clarifications of multiple imputation theory. *Prevn Sci.*, **8**, 206–213.

Pankowska, P., Bakker, B., Oberski, D. and Pavlopoulos, D. (2017) Estimating employment mobility using linked data from different sources: does linkage error matter? *Conf. New Techniques and Technologies for Statistics*. (Available from `https://www.conference-service.com/NTTS2017/documents/agenda/abstracts/abstract_138.html`.)

Reiter, J. P. and Raghunathan, T. E. (2007) The multiple adaptations of multiple imputation. *J. Am. Statist. Ass.*, **102**, 1462–1471.

Reurings, M. C. B. and Bos, N. M. (2012) Ernstig verkeersgewonden in de jaren 2009 en 2010: update van de cijfers. *Report*. Stichting Wetenschappelijk Onderzoek Verkeersveiligheid, The Hague. (Available from `https://www.swov.nl/sites/default/files/publicaties/rapport/r-2012-07.pdf`.)

Reurings, M. C. B. and Stipdonk, H. L. (2009) Ernstig gewonde verkeersslachtoffers in Nederland in 1993-2008. Stichting Wetenschappelijk Onderzoek Verkeersveiligheid, The Hague. (Available from `https://www.swov.nl/publicatie/ernstig-gewonde-verkeersslachtoffers-nederland-1993-2008`.)

Reurings, M. C. B. and Stipdonk, H. L. (2011) Estimating the number of serious road injuries in the Netherlands. *Ann. Epidem.*, **21**, 648–653.

Rubin, D. B. (1987) *Multiple Imputation for Nonresponse in Surveys*. New York: Wiley.

Vermunt, J. K. and Magidson, J. (2015) *Upgrade Manual for Latent GOLD 5.1*. Belmont: Statistical Innovations.

Vermunt, J. K., Van Ginkel, J. R., Der Ark, V., Andries, L. and Sijtsma, K. (2008) Multiple imputation of incomplete categorical data using latent class analysis. *Sociol. Methodol.*, **38**, 369–397.

Wang, C.-P., Brown, C. H. and Bandeen-Roche, K. (2005) Residual diagnostics for growth mixture models. *J. Am. Statist. Ass.*, **100**, 1054–1076.

Wong, E. (2011) Abbreviated injury scale. In *Encyclopedia of Clinical Neuropsychology* (eds J. S. Kreutzer, J. DeLuca and B. Caplan), pp. 5–6. New York: Springer.

DE GRUYTER
OPEN

Journal of Official Statistics, Vol. 33, No. 4, 2017, pp. 963–978, http://dx.doi.org/10.1515/JOS-2017-0045

How to Obtain Valid Inference under Unit Nonresponse?

Laura Boeschoten[1], *Gerko Vink*[2], *and Joop J.C.M. Hox*[2]

Weighting methods are commonly used in situations of unit nonresponse with linked register data. However, several arguments in terms of valid inference and practical usability can be made against the use of weighting methods in these situations. Imputation methods such as sample and mass imputation may be suitable alternatives, as they lead to valid inference in situations of item nonresponse and have some practical advantages. In a simulation study, sample and mass imputation were compared to traditional weighting when dealing with unit nonresponse in linked register data. Methods were compared on their bias and coverage in different scenarios. Both, sample and mass imputation, had better coverage than traditional weighting in all scenarios.

Imputation methods can therefore be recommended over weighting as they also have practical advantages, such as that estimates outside the observed data distribution can be created and that many auxiliary variables can be taken into account. The use of sample or mass imputation depends on the specific data structure.

Key words: Weighting; mass imputation; sample imputation; coverage.

1. Introduction

Missing data form a ubiquitous source of problems in survey research. A common research scenario occurs when respondents that are sampled from the population cannot be contacted, or when they are reluctant to conform to the survey. If no analysable information about the respondent is collected, we deem it unit nonresponse. In such a scenario, we can distinguish between two missing data problems. The first problem is that, when sampling from the population, not all units from the population are recorded (which is the usual process of sampling producing missing data by design). The second problem is that the sample is found to be incomplete. The severity of these problems is related to the probability each data point has of being missing.

The mechanism that governs these probabilities is called the missing data mechanism (Rubin 1976). To describe these mechanisms, we assume to have a data set consisting of an incomplete target variable Y and a fully observed covariate X. The incomplete target variable Y has two parts: an observed part Y_{obs} and a missing part Y_{mis}. An indicator variable R can be defined that scores a 0 when Y is missing and a 1 when Y is observed.

[1] Tilburg School of Social and Behavioral Sciences, Tilburg University, Warandelaan 2, 5037 AB Tilburg, The Netherlands. Email: L.Boeschoten@tilburguniversity.edu
[2] Department of Methodology & Statistics, Utrecht University, Padualaan 14, 3584 CH Utrecht, The Netherlands. Email: G.Vink@uu.nl and J.Hox@uu.nl

Acknowledgments: The authors would like to thank the anonymous associate editor and three referees for their constructive feedback on an earlier version of this manuscript. Furthermore, the authors would like to thank Barry Schouten and Sander Scholtus for their valuable input.

If the data are Missing Completely At Random (MCAR, Rubin 1976), the response probability for the respondents and nonrespondents is equal. This can be formally defined as:

$$P(R=0|Y_{obs},Y_{mis},X)=P(R=0) \tag{1}$$

"An example of MCAR is a weighing scale that ran out of batteries. Some of the data will be missing simply because of bad luck" (Van Buuren 2012, 7). If the data are Missing At Random (MAR, Rubin 1976), the distribution of the missing values is related to other observed values, formally defined:

$$P(R=0|Y_{obs},Y_{mis},X)=P(R=0|Y_{obs},X) \tag{2}$$

"For example, when placed on a soft surface, a weighing scale may produce more missing values than when placed on a hard surface. Such data are thus not MCAR. If, however, we know surface type and if we can assume MCAR within the type of surface, then the data are MAR" (Van Buuren 2012, 7). If the distribution of the missing values relates to unobserved values, it is called Missing Not At Random (MNAR, Rubin 1976), formally defined:

$$P(R=0|Y_{obs},Y_{mis},X)=P(R=0|Y_{obs},Y_{mis},X) \tag{3}$$

"For example, the weighing scale mechanism may wear out over time, producing more missing data as time progresses, but we fail to note this. If the heavier objects are measured later in time, then we obtain a distribution of the measurements that will be distorted" (Van Buuren 2012, 7).

Sometimes register data is available with information about the characteristics of the respondents and the nonrespondents that can be linked to the survey data (Bethlehem et al. 2011, 211). If there is a relationship between the selection mechanism and the survey variables, the estimators will systematically over- or under-represent the population characteristics. Such deviations can be corrected by weighting the observed data to conform to the known population parameters. If done properly, both distinct missing data problems can in theory be solved. However, there are several arguments against the use of weighting techniques to handle nonresponse. We list them in no particular order:

1. Weighting ignores the uncertainty about the missing data. This may result in too little variation about the estimates (Bethlehem et al. 2011, 184).
2. Weighting methods cannot create estimates that lie outside the observed data distribution. Although some researchers might view this as an advantage of weighting and would worry when a method could yield estimates outside the observed data distribution, an example given by Rubin illustrates when this could be problematic: "Consider dealing with censored data by weighting – data beyond or approaching the censoring point have zero or very small probabilities of being observed, and so either cannot be dealt with by weighting or imply a few observations with dominant weights. Weighting by inverse probabilities cannot create estimates outside the convex hull of the observed data, and estimates involving weights near the boundary have extremely large variance" (Rubin 1996, 486).

3. Uncertainty about the weights is ignored when weights are estimated from the data and thereby treated as fixed, given that the data conform to sampling variance. When taking additional measures, such as combining jackknife procedures with calibration, or by using design based analysis, weights can be treated as random.
4. Weighting has difficulties with handling large numbers of auxiliary variables, which are potentially needed to make the nonresponse ignorable (Rubin 1987, 155). Additional measures should then be taken, such as dimension reduction or propensity score estimation.
5. Weighting can have difficulties with creating sensible weights when more auxiliary information is incorporated. As a result, it is possible that the score on a target variable of an individual is used to represent a large group in the population. An illustrative example from the United States of America 2016 presidential elections show how one man heavily influenced the outcome of a poll due to extreme weights being given to his demographic category (Cohn 2016).
6. Some weighting methods cannot handle continuous variables.
7. Weighting cannot handle partial response. It is an all or nothing approach and may thereby discard valuable information (Van Buuren 2012, 22).

Because of arguments 1 and 3, we expect weighting to create too little variance and therefore to yield invalid inference (with confidence validity as defined by Rubin (1996)). We expect multiple imputation (MI) to be a good alternative method to correct for unit nonresponse, since it takes sampling variability as well as uncertainty due to missing values into account (Rubin 1987, 76). Furthermore, with MI there is no limit to the use of auxiliary information: continuous variables or the number of variables are less likely to pose problems, as the likelihood of the observed data given the unobserved data is taken into account. In cases of large numbers of variables or nonlinear associations, principal component analysis can be used (Howard 2012). In addition, item and unit non-response can be handled simultaneously with MI.

The goal of this article is to investigate whether MI is a suitable alternative for weighting when correcting for unit nonresponse. In this article, we distinguish between sample and mass imputation. With sample imputation, both item and unit nonresponse (occuring both in the sample) can be imputed. If the sample is a simple random sample without replacement ($\mathrm{SRS}_{\mathrm{WOR}}$) auxiliary information is only needed for the sample. However, sometimes registers with information about the whole population can be linked on a unit level to sample data sets. This is for example the case at Statistics Netherlands where complete population registers were used in the 2011 Dutch census (Schulte Nordholt et al. 2014). If this is the case, the nonsampled units can be imputed as well (besides the item and unit nonresponse within the sample). Mass imputation can then be applied with $\mathrm{SRS}_{\mathrm{WOR}}$ or complex samples.

Our definition of mass imputation should not be confused with the approach of Zhou et al. (2016), who generate a synthetic data set based on known population totals. A benefit of mass imputation is that every source of (linked) auxiliary information can be used for imputation. This means that a MNAR missing mechanism can become MAR, leading to more efficient estimation of (population) parameters.

We investigate the performance of weighting and both sample and mass imputation. As a reference, we also investigate complete case analysis (CCA), where no correction for unit nonresponse is made. With performance, summarized as ‘valid inference’ in the title, we mean obtaining unbiased parameter estimates and unbiased variance estimates.

2. Methodology

In this article, we distinguish between multiple auxiliary variables **X** and a single target variable y, which we assume to be normally distributed with mean μ and variance σ^2. If we would take a $\mathrm{SRS_{WOR}}$, the estimate of the sample mean of a target variable y is:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} y_i, \tag{4}$$

where y_i is the observation on the i^{th} sampled unit with $i = 1, \ldots, n$, where n is the sample size. The estimate of the variance of the mean is:

$$\mathrm{VAR}(\hat{\mu}) = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \hat{\mu})^2 \frac{1}{n}\left(1 - \frac{n}{N}\right), \tag{5}$$

where N is the size of the (finite) population. This is how μ and $\mathrm{VAR}(\mu)$ are estimated when the sample is completely observed. We will now discuss different methods to estimate these parameters in case of unit nonresponse.

2.1. Complete Case Analysis

When CCA is applied, nonrespondents are completely removed from the sample. μ and $\mathrm{VAR}(\mu)$ are estimated with the same equations used for a completely observed sample, as in Equations 4 and 5. However, with unit nonresponse, not all values in y are observed, and only the observed values in y are used to estimate μ and $\mathrm{VAR}(\mu)$ of the target variables:

$$\hat{\mu} = \frac{1}{n_{obs}}\sum_{i=1}^{n_{obs}} y_{obs_i}, \tag{6}$$

$$\mathrm{VAR}(\hat{\mu}) = \frac{1}{n_{obs}-1}\sum_{i=1}^{n_{obs}}(y_{obs_i} - \hat{\mu}_{obs})^2 \frac{1}{n_{obs}}\left(1 - \frac{n_{obs}}{N}\right). \tag{7}$$

2.2. Weighting

The weighted mean of a target variable is defined as

$$\hat{\mu} = \frac{\sum_{i=1}^{n_{obs}} w_i y_{obs_i}}{\sum_{i=1}^{n_{obs}} w_i} \tag{8}$$

where w_i is the weight corresponding to the i^{th} observation (Biemer and Christ 2008, 318) and μ is a vector quantity (and is so throughout the remainder of the article). The weights, w_i, can be estimated with different methods, such as poststratification, linear weighting, multiplicative weighting and propensity weighting. A full description of how to apply the different methods can be found in Chapter 8 of Bethlehem et al. (2011). De Waal et al. (2011, 237–244) show that under certain conditions, linear weighting and mass imputation yield the same estimate. Therefore, it would be interesting to use this method to estimate the weights, and investigate whether these methods also yield the same inference. For this reason, we use linear weighting to estimate w_i.

Linear weighting is a calibration method, and is thoroughly discussed by, among others, Deville and Särndal (1992) and Särndal et al. (1992). When estimating weights, it is important to note first that these weights (w_i) consist of two parts:

$$w_i = d_i \delta_i, \tag{9}$$

where d_i are the sampling design weights. For a $\mathrm{SRS_{WOR}}$, N and n are fixed numbers, d_i is constant and does not need to be estimated:

$$d_i = N/n. \tag{10}$$

δ_i is the adjustment factor. Our goal is to find a δ_i which makes w_i as close as possible to d_i, while respecting the calibration equation

$$\sum_{i=1}^{n_{obs}} w_i \mathbf{X}_i = \mathbf{t_X}, \tag{11}$$

where $\mathbf{X}$ represents the auxiliary variables and $\mathbf{t_X}$ are the population totals of $\mathbf{X}$. Minimizing the function

$$\sum_{i=1}^{n_{obs}} (w_i - d_i)^2 / d_i \tag{12}$$

leads to what is also known as linear weighting, which is a special case of calibration. We derive new weights here that modify as little as possible to the original sampling design weights d_i by minimizing the conditional value of the distance, given the realized observed sample n_{obs}. This leads to the calibrated weight

$$w_i = d_i(1 + \mathbf{X}_i'\lambda) \tag{13}$$

where λ is a vector of Lagrange multipliers determined from Equation 12:

$$\lambda = \mathbf{T}_{n_{obs}}^{-1}(\mathbf{t_X} - \hat{\mathbf{t}}_{\mathbf{X}\pi}). \tag{14}$$

The inverse of $\mathbf{T}_{n_{obs}}$ is

$$\mathbf{T}_{n_{obs}}^{-1} = \left(\sum d_i \mathbf{X}_i \mathbf{X}_i'\right)^{-1} \tag{15}$$

and $\hat{\mathbf{t}}_{\mathbf{X}\pi}$ is the Horvitz-Thompson (Horvitz and Thompson 1952) estimator for $\mathbf{X}$:

$$\hat{\mathbf{t}}_{\mathbf{X}\pi} = \sum_{i=1}^{n_{obs}} d_i \mathbf{X_i} \tag{16}$$

(Deville and Särndal 1992). The variance of a weighted mean can be approximated with methods such as Taylor linearization or Jacknife resampling (Stapleton 2008, 355). We use Taylor linearization and we assume for convenience that there is a vector of constants γ, such that $\gamma'\mathbf{X}_i = 1$ for all i. In that case, $\sum_{i=1}^{n_{obs}} w_i = N$. Then, the variance of a weighted mean can be approximated by:

$$\mathrm{VAR}(\hat{\mu}) = \frac{1}{N^2}\sum_{i=1}^{n_{obs}}\sum_{h=1,h\neq i}^{n_{obs}} \frac{\pi_{ih} - \pi_i\pi_h}{\pi_{ih}}\left(\delta_i\frac{e_i}{\pi_i}\right)\left(\delta_h\frac{e_h}{\pi_h}\right) \tag{17}$$

where π_i and π_h are the first order and π_{ih} the corresponding second order inclusion probabilities of observations i and h, and e_i (and e_h) are defined as:

$$e_i = y_i - \mathbf{X}_i'\mathbf{T}_{n_{obs}}^{-1}\sum_{l=1}^{n_{obs}}\mathbf{X}_l y_l d_l \tag{18}$$

(Särndal et al. 1992, 225–236).

2.3. Sample Imputation

With MI, each missing datapoint is imputed $m \geq 2$ times, resulting in m completed data sets. At least two imputations are needed to reflect the uncertainty about the imputations, although performing more imputations is often advisable. The m data sets can then be analyzed by standard procedures and the analyses combined into a single inference. A clear introduction to multiple imputation and different methods to impute the missing datapoints is given in Van Buuren (2012, Chapter 2).

With sample imputation, we only impute the nonrespondents in the sample. Because the imputation theory aims at inference about the population, sampling uncertainty is taken into account and we can use the standard rules for pooling.

The pooled estimate of μ is obtained by

$$\bar{\mu} = \frac{1}{m}\sum_{j=1}^{m}\hat{\mu}_j\,, \tag{19}$$

where m is the number of imputations with $j = 1, \ldots, m$ and $\hat{\mu}_j$ is the $\hat{\mu}$ of the j th imputed sample. $\overline{\mathrm{VAR}(\hat{\mu})}$ consists of multiple components (we therefore name it $\overline{\mathrm{VAR}(\hat{\mu})}_{\text{total}}$) and is estimated

$$\overline{\mathrm{VAR}(\hat{\mu})}_{\text{total}} = \overline{\mathrm{VAR}(\hat{\mu})}_{\text{within}} + \mathrm{VAR}(\hat{\mu})_{\text{between}} + \frac{\mathrm{VAR}(\hat{\mu})_{\text{between}}}{m}, \tag{20}$$

where $\overline{\mathrm{VAR}(\hat{\mu})}_{\text{within}}$ is the within imputation variance and $\mathrm{VAR}(\hat{\mu})_{\text{between}}$ is the between imputation variance. $\overline{\mathrm{VAR}(\hat{\mu})}_{\text{within}}$ is calculated by

$$\overline{\mathrm{VAR}(\hat{\mu})}_{\text{within}} = \frac{1}{m}\sum_{j=1}^{m}\mathrm{VAR}(\hat{\mu})_{\text{within}_j} \tag{21}$$

and $\mathrm{VAR}(\hat{\mu})_{\text{between}}$ is calculated by

$$\mathrm{VAR}(\hat{\mu})_{\text{between}} = \frac{1}{m-1}\sum_{j=1}^{m}(\hat{\mu}_j - \bar{\mu})(\hat{\mu}_j - \bar{\mu})'. \tag{22}$$

2.4. Mass Imputation

With mass imputation, the estimate of μ is also obtained by Equation 19, although $\hat{\mu}_j$ now corresponds to the j th imputed version of the population instead of the the j th imputed sample.

Because we impute the population, there is no variance due to sampling. Therefore, $\overline{\mathrm{VAR}(\hat{\mu})}_{\mathrm{within}} = 0$ and we can adjust Equation 20 to

$$\overline{\mathrm{VAR}(\hat{\mu})}_{\mathrm{total}} = \mathrm{VAR}(\hat{\mu})_{\mathrm{between}} + \frac{\mathrm{VAR}(\hat{\mu})_{\mathrm{between}}}{m}. \tag{23}$$

For a thorough description of making multiply imputed inference when sampling variance is not of interest see Vink and Van Buuren (2014).

3. Simulation Approach

To empirically evaluate the performance of the different analysis methods, we conducted a simulation study using R (R Core Team 2015, version 3.2.2). The properties we manipulate in the simulation design can be summarized as follows:

- The correlation between the auxiliary variables and the target variables: 0.30; 0.50.
- The amount of missingness: 25%; 50%.
- The missingness mechanism: MCAR; left-tailed MAR.
- The analysis method: CCA; lineair weighting (calibration); Bayesian normal linear imputation of the sample; Bayesian normal linear imputation of the population.

We now discuss the properties of the simulation design in more detail.

3.1. The Correlation Structure

We start by creating a large but finite population of 100,000 units with two auxiliary (X_1 and X_2) and two target variables (Y_1 and Y_2). The population data is multivariate normally distributed with $\boldsymbol{\mu}$ and Σ:

$$\begin{pmatrix} X_1 \\ X_2 \\ Y_1 \\ Y_2 \end{pmatrix} = MVN(\boldsymbol{\mu}, \Sigma),$$

where $\boldsymbol{\mu}$ is:

$$\boldsymbol{\mu} = \begin{array}{c} X_1 \\ X_2 \\ Y_1 \\ Y_2 \end{array} \begin{pmatrix} 3 \\ 2 \\ 0 \\ 170 \end{pmatrix}$$

and Σ is either:

$$\Sigma = \begin{array}{c c} & \begin{array}{cccc} X_1 & X_2 & Y_1 & Y_2 \end{array} \\ \begin{array}{c} X_1 \\ X_2 \\ Y_1 \\ Y_2 \end{array} & \begin{pmatrix} 1.00 & 0.08 & 1.34 & 1.90 \\ 0.08 & 0.25 & 0.67 & 0.95 \\ 1.34 & 0.67 & 20.00 & 4.24 \\ 1.90 & 0.95 & 4.24 & 40.00 \end{pmatrix} \end{array}$$

when the correlations between the target variables and the auxiliary variables are 0.30, and

$$\Sigma = \begin{array}{c c} & \begin{array}{cccc} X_1 & X_2 & Y_1 & Y_2 \end{array} \\ \begin{array}{c} X_1 \\ X_2 \\ Y_1 \\ Y_2 \end{array} & \begin{pmatrix} 1.00 & 0.08 & 2.24 & 3.16 \\ 0.08 & 0.25 & 1.12 & 1.58 \\ 2.24 & 1.12 & 20.00 & 4.24 \\ 3.16 & 1.58 & 4.24 & 40.00 \end{pmatrix} \end{array}$$

when the correlations between the target variables and the auxiliary variables are 0.50. The target variables X_1 and X_2 are transformed into categorical variables with respectively six and four categories, because auxiliary register information is in practice often categorical.

3.2. The Amount of Missingness and the Missingness Mechanism

From the population of size 100,000, a random sample of size 5,000 is drawn. In each sample, either 25% or 50% missingness is induced in the Y_1 and Y_2 variables.

The missingness in the target variables follow MCAR or left-tailed MAR mechanisms conform the procedure described by Van Buuren (2012, 63). With a left-tailed MAR mechanism, the probability of having missing values in the target variables is larger for smaller values on the auxiliary variables. For example, consider the number of employees of a company to be the auxiliary variable on which the missingness depends and working conditions of the company as target variables. In this situation, it is likely that more missing values are found at the companies with fewer employees. The first reason for this is that smaller companies are often less well organized. However, researchers are also probably more interested in larger companies, and are more likely to re-contact these in cases of nonresponse. If you sort companies on an axis with number of employees, you find more missing values on the left side of this axis, where the smaller companies are found.

3.3. The Analysis Method

We estimate $\hat{\mu}$ and VAR($\hat{\mu}$) of the target variables by making use of CCA, weighting, sample imputation and mass imputation. There are slight differences between the simulation setup within the different methods. For CCA, 96.25% or 97.50% of the 100,000 population values could be deleted directly from the target variables using MAR or MCAR to come to a sample of 5,000 with 25% or 50% missing values. The estimates of the incomplete sample can be compared directly to the population values.

Table 1. Smallest and largest adjustment factor per simulated condition.

		MCAR		MARleft	
cor.	% mis	min	max	min	max
	25	1.2305	1.4511	0.9682	4.2467
0.3	50	1.7472	2.3080	0.9419	13.8479
	25	1.2298	1.4513	0.9646	4.2426
0.5	50	1.7454	2.3128	0.9273	13.4502

For weighting, we first select randomly 5,000 cases from the population. Next, we create unit missingness following one of the missingness mechanisms. We weight the respondents to the total sample using the population totals. Weights are calculated using the survey package (Lumley 2014, version 3.30-3) in R (R Core Team 2015, version 3.2.2) with the `calibrate()` function. We evaluate the performance of weighting by comparing the results of the weighted sample to the population values. The design weights are $d_i = N/n = 100{,}000/5{,}000 = 20$. The adjustment factors δ_i can be found in Table 1, which can be used to compute the weights $w_i = d_i\delta_i$.

We are aware that some of the correction weights are considered large and that weighted estimates may be inefficient in such scenarios. An option would be to trim the weights to predefined boundaries. However, by not trimming the weights, we are able to investigate the performance of the method itself and its default options to other methods and their default options.

For sample imputation, we also 5,000 cases from the population and create unit missingness in the sample. Next, we multiply impute the sample and compare the results of the imputed sample to the population results.

For mass imputation, we can directly delete 96.25% or 97.50% of the values of the target variables and multiply impute the population. The results of the imputed population are compared to the original population results. Both sample and mass imputations are executed with `mice` (Van Buuren and Groothuis-Oudshoorn 2011) in `R` (R Core Team 2015) using Bayesian normal linear imputation (`mice.impute.norm()`) as the imputation method with five imputations and five iterations for the algorithm to converge.

3.4. *Performance Measures*

We estimate $\hat{\mu}$ and VAR($\hat{\mu}$) by using the previously discussed methods and replicate this procedure 1,000 times. In each replication, we investigate these estimates by looking at two performance measures. First, we look at the bias of $\hat{\mu}$ of the two target variables. This bias is equal to the difference between the average estimate over all replications and the population value. Next, we look at the coverage of the 95% confidence interval. This is equal to the proportion of times that the population value falls within the 95% confidence interval constructed around the $\hat{\mu}$'s of the two target variables over all replications.

3.5. *Expectations*

When CCA is applied and the missingness is MCAR, the probability of being missing is equal for every unit in the sample. Therefore, we do not expect biased estimates of $\hat{\mu}$.

However, with MAR, the probability of being missing is not equal for every unit, and we do expect bias. Since parameter uncertainty and uncertainty about the missing values is not taken into account when estimating the variance of the mean, we also expect undercoverage with MAR.

When weighting is applied, we expect unbiased estimates of $\hat{\mu}$ under both MCAR and MAR. The variance estimate takes the weights and parameter uncertainty into account, but not the uncertainty about the missing values. Therefore, we expect an estimate of the variance of the mean that is a bit too small, resulting in undercoverage under MAR.

For sample imputation we expect unbiased estimates and adequate coverage under both MCAR and MAR.

For mass imputation, we also expect unbiased estimates and adequate coverage under both MCAR and MAR.

4. Results

The simulation results are depicted in Table 2. Note that the results for CCA in terms of coverage and confidence interval width with correlation 0.30 and 0.50 look identical under MCAR. Small differences in the results were found, but these occur after the fourth decimal.

4.1. *The Missingness Mechanism*

The methods that aim to correct for the nonresponse show equivalent bias and coverage patterns under MCAR and left-tailed MAR missingness mechanisms. Naturally, the loss of observed information results in larger confidence interval widths under left tailed MAR missingness than under MCAR missingness mechanisms. CCA is unable to handle the estimation under left-tailed MAR missingness and yields large bias, zero coverage and confidence intervals that are, as expected, equally wide to those under MCAR.

4.2. *The Correlation Structure*

Larger correlations are often beneficial when solving incomplete data problems because the correlations give strong direction to the estimation procedure. This is clearly visible in all methods that aim to solve the missingness problem as confidence intervals tend to become smaller when the correlation between the target variables and the linked register data increases. Interestingly, the coverage rates for weighting are negatively impacted under large correlations. In this specific situation the bias remains roughly the same as under low-correlation simulations, while the confidence interval widths decrease. As a result, the simulations for weighting demonstrate lower coverage of the population mean.

4.3. *The Amount of Missingness*

In general, it can be said that when amounts of missingness become larger, incomplete data problems become more difficult. More specifically, the probability that you deal with a MNAR mechanism increases. None of the methods seem negatively impacted by the increased amount of missingness, when compared to the results under less missingness. However, the confidence intervals naturally tend to become wider as there is less information about the observed data.

Table 2. Simulation results. Depicted are the bias of the mean of Y_1 and Y_2, coverage of the 95% confidence interval and width of the 95% confidence interval for the four methods under varying simulation conditions.

				MCAR			MARleft		
Method	Correlation	% mis	*Y*	bias	coverage	CI width	bias	coverage	CI width
CCA	0.3	25	1	−0.0003	0.9620	0.2872	−1.1480	0.0000	0.2866
			2	0.0020	0.9580	0.4049	−1.6535	0.0000	0.4060
		50	1	−0.0003	0.9590	0.3516	−1.1728	0.0000	0.3479
			2	−0.0000	0.9620	0.4957	−1.6889	0.0000	0.4930
	0.5	25	1	−0.0002	0.9620	0.2872	−1.9291	0.0000	0.2848
			2	0.0020	0.9570	0.4049	−2.7578	0.0000	0.4024
		50	1	−0.0003	0.9590	0.3516	−1.9691	0.0000	0.3460
			2	−0.0000	0.9620	0.4957	−2.8181	0.0000	0.4888
Weighting	0.3	25	1	−0.0021	0.9310	0.2626	−0.0037	0.9370	0.2736
			2	0.0033	0.9400	0.3693	0.0007	0.9360	0.3835
		50	1	−0.0015	0.9340	0.3237	−0.0021	0.9390	0.3710
			2	0.0020	0.9370	0.4551	0.0014	0.9390	0.5184
	0.5	25	1	−0.0031	0.8820	0.2236	−0.0029	0.8950	0.2331
			2	0.0017	0.9060	0.3140	0.0004	0.9030	0.3266
		50	1	−0.0032	0.9070	0.2756	−0.0015	0.9180	0.3160
			2	−0.0004	0.9250	0.3868	0.0035	0.9080	0.4417

Table 2. *Continued.*

				MCAR			MARleft		
Method	Correlation	% mis	Y	bias	coverage	CI width	bias	coverage	CI width
Sample imputation	0.3	25	1	−0.0021	0.9650	0.2959	−0.0040	0.9520	0.3037
			2	0.0010	0.9460	0.4125	0.0076	0.9480	0.4293
		50	1	0.0004	0.9540	0.3422	−0.0062	0.9430	0.4281
			2	0.0005	0.9550	0.4879	0.0106	0.9540	0.6103
	0.5	25	1	−0.0016	0.9650	0.2824	−0.0014	0.9460	0.2905
			2	0.0013	0.9440	0.3954	0.0009	0.9450	0.4077
		50	1	0.0005	0.9540	0.3422	−0.0044	0.9540	0.3842
			2	0.0007	0.9550	0.4879	0.0098	0.9390	0.5402
Mass imputation	0.3	25	1	0.0003	0.9450	0.3857	−0.0200	0.9510	0.5419
			2	0.0005	0.9590	0.5480	0.0268	0.9460	0.7713
		50	1	−0.0008	0.9390	0.4772	−0.0237	0.9570	0.6636
			2	0.0030	0.9560	0.6752	0.0229	0.9440	0.9117
	0.5	25	1	−0.0001	0.9570	0.3289	−0.0051	0.9490	0.4663
			2	0.0007	0.9630	0.4603	0.0423	0.9400	0.6507
		50	1	−0.0033	0.9390	0.4033	−0.0051	0.9530	0.5743
			2	−0.0010	0.9470	0.5665	0.0438	0.9620	0.8202

Note that results of two target variables Y_1 and Y_2 are shown, which both have their own mean and variance, as illustrated in Subsection 3.1.

4.4. Overall Efficiency

We investigate efficiency of the methods in the sense that we investigate which methods have the smallest confidence interval widths under which conditions. When investigating the results, we see that CCA is an efficient method yielding valid inference under MCAR. There is no need for handling the nonresponse as the nonresponse is perfectly ignorable: the set of observed values can simply be analyzed to obtain unbiased estimates about the population. Even though the missingness is MCAR, treating the missingness can increase the statistical power of the analyses at hand. This is demonstrated by weighting and imputing the sample as the confidence intervals under these approaches are generally more narrow than under CCA. Mass imputation, on the other hand, does not show this result. This can simply be explained by the severity of the problem that is considered with mass imputation in our simulation setup. After all, under mass imputation we aim to solve at least a 96.25% missingness problem.

Even though mass imputation may yield less sharp inference than sample imputation and weighting, the inference is valid and exhibits correct variance properties under all simulation conditions. The same can be said of sample imputation, but with much sharper inference. The estimates obtained under weighting are unbiased, the intervals are among the smallest, but the coverage rates are somewhat low. Especially when larger correlations occur in the data, one could question the validity of inference obtained by weighting. Furthermore, it is surprising that these low coverage rates occur under both MCAR and MAR, indicating that the variance of a weighted mean estimated using Taylor linearization indeed ignores uncertainty about the missing data and possibly about the weights as well.

5. Discussion

We have demonstrated that weighting and imputation are practically equivalent when unbiased estimation is of interest. However, the inference obtained under weighting may be questionable in situations where multiple imputation approaches exhibit correct variance properties and well-covered population estimates. In general it holds that inferring about the population by imputing the sample yields efficient, unbiased estimates in all simulated conditions, which is in line with conclusions drawn by Peytchev (2012).

A main characteristic of our simulation approach is that it deals with a SRS_{WOR}. With more complex sampling approaches, it would not be sufficient to only impute the sample, since the complex sampling structure is then ignored. Although we did not investigate this, we do expect that mass imputation will lead to unbiased and efficient estimates when a more complex sample is drawn because the design of the complex sample is always based on observed information, so the missingness mechanism describing the sample to the population is always MAR. However, this is not included in this simulation study, and additional research should be done.

Furthermore, in this simulation we assume quite an ideal situation, where the sample is perfectly linked to a completely observed population register. Of course, this is not often the case in practice. In addition to the traditional Total Survey Error framework introduced by Groves et al. (2009), Zhang (2012) introduced a two-phase life cycle of integrated statistical micro data, which also discusses the errors that might be encountered when multiple data sets are combined, such as identification or comparability error. Furthermore, we also assume that our population register is perfectly observed. This is in practice also not often the case, although this is commonly assumed by many researchers. Recently, imputation methods have been developed to take misclassification in combined data sets into account, for example by assuming that a certain proportion of the data is misclassified (Manrique-Vallier and Reiter 2016) or by estimating the number of misclassified units by using information from multiple sources (Boeschoten et al. 2016).

It is clear that weighting does not include all sources of uncertainty. This limits the validity of the inference obtained under weighting. Theoretically, these sources of uncertainty could be added to the estimations that are obtained from weighted data sets. However, we have demonstrated that the imputation approaches take the sources of variations about the observed and missing data properly into account. Adjusting the weighted estimation to allow for valid inference under unit nonresponse would therefore be redundant as it is a complicated step to solve a problem that can be straightforwardly solved by another approach.

In addition, weighting cannot handle partial response (Van Buuren 2012, 22). Analyzing multivariate response data with partial responses will be particularly problematic when weighting is applied, and multiple imputation is a very suitable alternative in this setting.

It is known that complete case analysis yields valid inference under MCAR mechanisms and that its performance may be severely impaired under MAR missingness. The results of complete case analysis in simulations can be very informative, as it can act as a point of reference for the performance of other methods. At the same time, the validity of the simulation scheme can be assessed, because we know the theoretical properties under which complete case analysis can be applied. Failure to meet these expectations indicates a faulty simulation scheme. This is not the case.

The simulation study conducted in this article illustrated that multiple imputation methods lead to valid inference in situations of unit nonresponse and have practical advantages over weighting. Whether sample or mass imputation methods should be used depends on the specific data structure.

6. References

Bethlehem, J., F. Cobben, and B. Schouten. 2011. *Handbook of Nonresponse in Household Surveys,* volume 568 of *Wiley Handbooks in Survey Methodology*. John Wiley & Sons, Inc., Hoboken, New Jersey.

Boeschoten, L., D. Oberski, and T. de Waal. 2016. “Estimating Classification Error under Edit Restrictions in Combining Survey-Register Data.” *Journal of Official Statistics* 33:921–962. Doi: http://dx.doi.org/10.1515/JOS-2017-0044.

Cohn, N. 2016. “How One 19-Year-Old Illinois Man is Distorting National Polling Averages.” *The New York Times*. Available at: https://nyti.ms/2k5sB5z (accessed September 26, 2017).

De Waal, T., J. Pannekoek, and S. Scholtus. 2011. *Handbook of Statistical Data Editing and Imputation,* volume 563 of *Wiley Handbooks in Survey Methodology*. John Wiley & Sons, Inc., Hoboken, New Jersey.

Deville, J.-C. and C.-E. Särndal. 1992. "Calibration Estimators in Survey Sampling." *Journal of the American statistical Association* 87: 376–382.

Groves, R.M., F.J. Fowler, Jr, M.P. Couper, J.M. Lepkowski, E. Singer, and R. Tourangeau. 2009. *Survey Methodology,* volume 561 of *Wiley Series in Survey Methodology*. John Wiley & Sons, Inc., Hoboken, New Jersey.

Horvitz, D.G. and D.J. Thompson. 1952. "A Generalization of Sampling without Replacement from a Finite Universe." *Journal of the American Statistical Association* 47: 663–685.

Howard, W.J. 2012. *Using Principal Component Analysis (pca) to Obtain Auxiliary Variables for Missing Data in Large Data Sets*. University of Kansas. PhD Dissertation.

Lumley, T. 2014. *Analysis of Complex Survey Samples*. Available at: http://cran.r-project.org/web/packages/survey/survey.pdf (accessed September 26, 2017).

Manrique-Vallier, D. and J.P. Reiter. 2016. "Bayesian Simultaneous Edit and Imputation for Multivariate Categorical Data." *Journal of the American Statistical Association*. Doi: http://dx.doi.org/10.1080/01621459.2016.1231612.

Peytchev, A. 2012. "Multiple Imputation for Unit Nonresponse and Measurement Error." *Public Opinion Quarterly* 76: 214–237. Doi: https://doi.org/10.1093/poq/nfr065.

R Core Team. 2015. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rubin, D.B. 1976. "Inference and Missing Data." *Biometrika* 63: 581–592.

Rubin, D.B. 1987. *Multiple Imputation for Nonresponse in Surveys*. Wiley series in probability and mathematical statistics. John Wiley & Sons, New York, USA.

Rubin, D.B. 1996. "Multiple Imputation after 18+ Years." *Journal of the American Statistical Association* 91: 473–489.

Särndal, C., B. Swensson, and J. Wretman. 1992. *Model Assisted Survey Sampling*. Springer Series in Statistics. Springer-Verlag.

Schulte Nordholt, E., J. van Zeijl, and L. Hoeksma. 2014. *Dutch Census 2011, Analysis and Methodology*. The Hague/Heerlen. Available at: https://www.cbs.nl/NR/rdonlyres/5FDCE1B4-0654-45DA-8D7E-807A0213DE66/0/2014b57pub.pdf (accessed 26 September 2017).

Stapleton, L.M. 2008. "Analysis of Data from Complex Surveys." In *International Handbook of Survey Methodology*, edited by E.D. De Leeuw, J.J. Hox, and D. Dillman, 342–369. Psychology Press, Taylor & Francis Group, New York.

Van Buuren, S. 2012. *Flexible Imputation of Missing Data*. CRC press, Boca Raton, Florida.

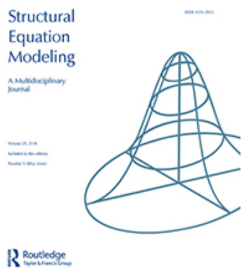
Van Buuren, S. and K. Groothuis-Oudshoorn. 2011. "mice: Multivariate Imputation by Chained Equations in R." *Journal of Statistical Software* 45: 1–67. Doi: http://dx.doi.org/10.18637/jss.v045.i03.

Vink, G. and S. van Buuren. 2014. "Pooling Multiple Imputations when the Sample Happens to be the Population." *arXiv preprint arXiv:1409.8542*. Available at: https://arxiv.org/pdf/1409.8542.pdf (accessed 26 September 2017).

Zhang, L.-C. 2012. “Topics of Statistical Theory for Register-Based Statistics and Data Integration.” *Statistica Neerlandica* 66: 41–63.

Zhou, H., M.R. Elliott, and T.E. Raghunathan. 2016. “A Two-Step Semiparametric Method to Accommodate Sampling Weights in Multiple Imputation.” *Biometrics* 72: 242–252.

Received November 2015
Revised January 2017
Accepted March 2017



Routledge
Taylor & Francis Group

Structural Equation Modeling: A Multidisciplinary Journal

ISSN: 1070-5511 (Print) 1532-8007 (Online) Journal homepage: https://www.tandfonline.com/loi/hsem20

Building Latent Class Growth Trees

Mattis van den Bergh & Jeroen K. Vermunt

To cite this article: Mattis van den Bergh & Jeroen K. Vermunt (2018) Building Latent Class Growth Trees, Structural Equation Modeling: A Multidisciplinary Journal, 25:3, 331-342, DOI: 10.1080/10705511.2017.1389610

To link to this article: https://doi.org/10.1080/10705511.2017.1389610

Published online: 21 Nov 2017.

Structural Equation Modeling: A Multidisciplinary Journal, 25: 331–342, 2018
ISSN: 1070-5511 print / 1532-8007 online
DOI: https://doi.org/10.1080/10705511.2017.1389610

Routledge
Taylor & Francis Group

Building Latent Class Growth Trees

Mattis van den Bergh and Jeroen K. Vermunt
Tilburg University

Researchers use latent class growth (LCG) analysis to detect meaningful subpopulations that display different growth curves. However, especially when the number of classes required to obtain a good fit is large, interpretation of the encountered class-specific curves might not be straightforward. To overcome this problem, we propose an alternative way of performing LCG analysis, which we call LCG tree (LCGT) modeling. For this purpose, a recursive partitioning procedure similar to divisive hierarchical cluster analysis is used: Classes are split until a certain criterion indicates that the fit does not improve. The advantage of the LCGT approach compared to the standard LCG approach is that it gives a clear insight into how the latent classes are formed and how solutions with different numbers of classes relate. The practical use of the approach is illustrated using applications on drug use during adolescence and mood regulation during the day.

Keywords: hierarchical clustering, latent class growth analysis, latent class growth trees, longitudinal data, mixture models

Longitudinal data are used by social scientists to study development of behaviors or other phenomena. The analysis will often be done with latent growth curve models (MacCallum & Austin, 2000), with the aim to assess interindividual differences in intraindividual change over time (Nesselroade, 1991). The typical growth model can be described as a multilevel model (Raudenbush & Bryk, 2002), in which the intercept and slopes of the time variables are allowed to vary across individuals. This heterogeneity is captured using random effects, which are continuous latent variables (Jung & Wickrama, 2008). This approach assumes that the growth trajectories of all individuals can be appropriately described by a single set of the growth parameters, and thus all individuals come from a single population. Growth mixture modeling relaxes this assumption by allowing for differences in growth parameters across unobserved subpopulations; that is, each latent class has a separate growth model. However, fully unrestricted growth mixture models are seldom used in practice, in part due to frequent estimation problems, as well as the preference for simpler, restricted models. The most widely used form of growth mixture modeling is latent class growth (LCG) analysis, whereby the variances and covariances of the growth factors within classes are fixed to zero (Jones, Nagin, & Roeder, 2001; Nagin & Land, 1993). This assumes that all individuals within a class follow the same trajectory and thus there is no residual heterogeneity within classes.

When an LCG model is applied, two key modeling decisions need to be made: the number of classes and the shape of the class-specific trajectories. In general, the decision on the number of classes is of more importance than the decision on the shape of the trajectory of each class as long as the shape is flexible enough (Nagin, 2005). Different fit statistics are available to handle the problem of model selection in LCG models, such as the Akaike's information criterion (AIC; Akaike, 1974), the Bayesian information criterion (BIC; Schwarz, 1978), the sample-size-adjusted BIC (Sclove, 1987), the Lo–Mendell–Rubin likelihood ratio test (Lo, Mendell, & Rubin, 2001), and the bootstrap likelihood ratio test (McLachlan & Peel, 2004). The benefits and limitations of these measures have been studied several times (e.g., Nylund, Asparouhov, & Muthén, 2007; Tofighi & Enders, 2008). However, these indexes rarely point to the same best model. Grimm, Ram, and Estabrook (2017) therefore recommended considering all available fit information when selecting a model and supplementing this information

Correspondence should be addressed to Mattis van den Bergh, Methodology and Statistics, Tilburg University, Warandelaan 2, 5037 AB, Tilburg, The Netherlands. E-mail: m.vdnbergh@uvt.nl

with substantive knowledge of the phenomena being studied (see also Muthén, 2003). Although there is nothing wrong with such a procedure, in practice it is often perceived as being problematic, especially when the model is applied with a large data set; that is, when the number of time points or the number of subjects is large. One problem occurring in such situations is that the selected number of classes could be rather large (Francis, Elliott, & Weldon, 2016). This causes the class trajectories to pick up very specific aspects of the data, which might not be interesting for the research question at hand. Moreover, these specific trajectories are hard to interpret substantively and compare to each other. This, combined with the fact that usually one would select a different number of classes depending on the model selection criterion used (e.g., AIC or BIC), means that one might wish to inspect multiple solutions, as each of them could reveal specific relevant features in the data. However, it is fully unclear how solutions with different numbers of classes are connected, making it impossible to see what a model with more classes adds to a model with fewer classes.

To circumvent the issues just mentioned, it is most convenient if models with differing numbers of classes are substantively related; in other words, a model with $K + 1$ classes is a refined version of a model with K classes, where one of the classes is split in two parts. Such an approach would result in a hierarchical structure, comparable to hierarchical cluster analysis (Everitt, Landau, Leese, & Stahl, 2011) or regression trees (Friedman, Hastie, & Tibshirani, 2001). Van Der Palm, Van Der Ark, and Vermunt (2016) developed an algorithm for hierarchical latent class analysis that can be used for this purpose. Although they focused on density estimation, with some adaptations their algorithm has also been used to build so-called latent class trees for substantive interpretation (van Den Bergh, Schmittmann, & Vermunt, 2017). In this article, this procedure is extended to the longitudinal framework to construct latent class growth trees (LCGT).

With LCGT analysis, a hierarchical structure is imposed on the latent classes by estimating one- and two-class models on a "parent" node, which initially comprised the full data. If the two-class model is preferred according to a certain information criterion, the data are split into child nodes and separate data sets are constructed for each of the child nodes. The split is based on the posterior class membership probabilities; hence, the data patterns in each new data set will be the same as the original data set, but with weights equal to the posterior class membership probabilities for the child class concerned. Subsequently, each new child node is treated as a parent and it is checked again whether a two-class model provides a better fit than a one-class model on the corresponding weighted data set. This procedure continues until no node is split up anymore. Because of this sequential algorithm, the classes at different levels of the tree can be substantively related, as child classes are subclasses of a parent class. Therefore, LCGT modeling allows for direct interpretation of the relationship between solutions with different numbers of classes, while still retaining the same statistical basis.

The remainder of the article is set up as follows. In the next section, we discuss the basic LCG model and show how it can be used to build an LCGT. Also split criteria and guidelines for deviating from a binary split at the root of the tree are discussed, together with an entropy measure for the post-hoc evaluation of the quality of splits. Two empirical data sets are used to illustrate LCGT analysis. The article concludes with final remarks by the authors.

METHOD

Latent Class Growth Models

Let y_{it} denote the response of individual i at time point t, T_i the number of measurements of person i, and $\mathbf{y}_i$ the full response vector of person i. Moreover, let X be the discrete latent class variable, k a particular latent class, and K the number of latent classes. An LCG model is, in fact, a regression model for the responses y_{it}, where time variables are used as predictors and where intercept and slope parameters differ across latent classes. We define the LCG model within the framework on the generalized linear model, which allows dealing with different scale types of the response variable (Muthén, 2004; Vermunt, 2007).

Let $E(y_{it}|X = k)$ denote the expected value of the response at time point t for latent class k. After an appropriate transformation $g(\cdot)$, which mainly depends on the measurement level of the response variable, $E(y_{it}|X = k)$ is modeled as a linear function of time variables. The most common approach is to use polynomial growth curves, which yields the following regression model for latent class k:

$$g[E(y_{it}|X = k)] = \beta_{0k} + \beta_{1k} \cdot t + \beta_{2k} \cdot t^2 + \ldots + \beta_{sk} \cdot t^s \qquad (1)$$

The choice of the degree of the polynomial (the value of s) is usually an empirical matter, although polynomials of a degree larger than three are seldom used. Recently, Francis et al. (2016) proposed an alternative approach involving the use of baseline splines in LCG models.

To complete the model formulation for the response vector $\mathbf{y}_i$, we have to define the form of the class-specific densities $f(y_{it}|X = k)$, which could be univariate normal for a continuous response, binomial for a binary response, and so on. The response density for class k is a function of the expected value $E(y_{it}|X = k)$ and for continuous variables also of the residual variance. The LCG model for $\mathbf{y}_i$ can now be defined as follows:

$$f(\mathbf{y}_i) = \sum_{k=1}^{K} P(X = k) \prod_{t=1}^{T_i} f(y_{it}|X = k), \qquad (2)$$

where the size of class k is represented by $P(X = k)$. A graphical representation of an LCG model with $K = 3$ can be seen in Figure 1.

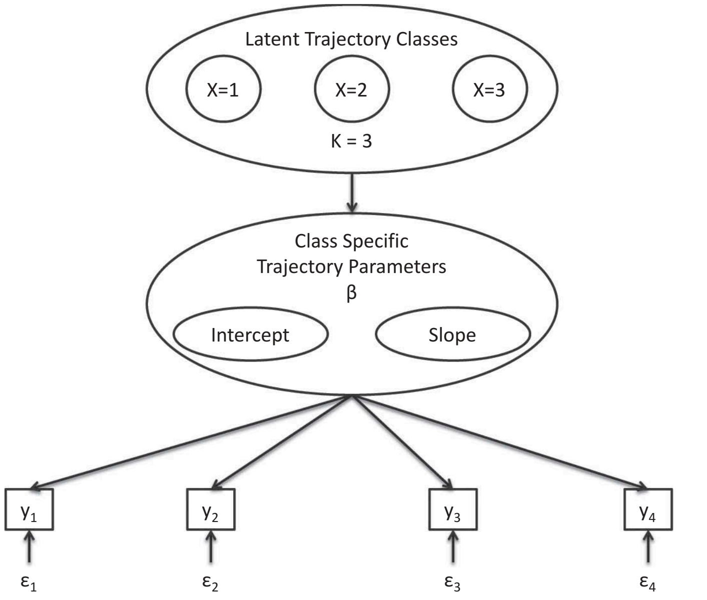


FIGURE 1 Graphical representation of a latent class growth model with three trajectory classes.

The model estimates (the β parameters and class sizes) can be obtained by maximizing the following log-likelihood function:

$$logL(\theta;\mathbf{y}) = \sum_{i=1}^{N} \log f(\mathbf{y}_i), \qquad (3)$$

where $f(\mathbf{y}_i)$ takes the form defined in Equation 2 and N denotes the total sample size. Maximization is usually achieved through an expectation maximization (EM) algorithm (Dempster, Laird, & Rubin, 1977), possibly combined with a Newton-type algorithm (Vermunt & Magidson, 2013).

After selecting a particular model, individuals could be assigned to latent classes based on their the posterior class membership probabilities. Using the Bayes theorem, these probabilities are obtained as follows:

$$P(X = k|\mathbf{y}_i) = \frac{P(X = k)\prod_{t=1}^{T_i} f(y_{it}|X = k)}{f(\mathbf{y}_i)}. \qquad (4)$$

Latent Class Growth Tree Models

Using an algorithm similar to the algorithm developed by Van der Palm et al. (2016) for divisive latent class analysis, an LCG model can also be constructed in a tree form. Such an LCGT has the advantages that increasing K classes to $K + 1$ classes results in directly related classes. This is because newly formed classes are obtained by splitting one of the K classes. Due to this direct relation, models with different numbers of classes can be substantively related, while still retaining the same statistical basis. In what follows, we first describe the algorithm for constructing an LCGT in more detail, and subsequently discuss various statistics that can be used during this process.

An LCGT consists of parent and child nodes. Every set of child nodes is based on one parent node and the first parent node consists of the root node containing the complete data set. At each parent node, standard LCG models are used and its child nodes are the classes assessed with the selected parent model. At the next level of the tree, these child nodes, in their turn, become parent nodes, and conditional on each new parent node a new set of LCG models is defined. This process continues until a stopping criterion is reached, for example, when the BIC no longer decreases when splitting.

The basic equations of the growth curves of an LCGT model do not differ from those of a standard LCG model (e.g., Equation 1). The fact is that the LCGT model is based on LCG models at parent nodes, which can be formulated as follows:

$$Pf(\mathbf{y}_i|X_{parent}) = \sum_{k=1}^{K} P(X_{child} = k|X_{parent}) \prod_{t=1}^{T} f(y_{it}|X_{child} = k, X_{parent}), \qquad (5)$$

where X_{parent} represents the parent class at level l and X_{child} represents one of the K possible newly formed classes at level $l + 1$, with in general K being 2. Furthermore, $P(X_{child} = k|X_{parent})$ represents the size of a class, given the parent node, and $f(y_{it}|X_{child} = k, X_{parent})$ represents the class-specific response density at time point t, given the parent class. In other words, as in a standard LCG analysis, a model for $\mathbf{y}_i$ is defined, but now conditioned on belonging to the parent class concerned.

Estimation of the LCG model at the parent node X_{parent} involves maximizing the following weighted log-likelihood function:

$$\log L(\theta; \mathbf{y}, X_{parent}) = \sum_{i=1}^{N} w_{i,X_{parent}} P(\mathbf{y}_i | X_{parent}), \tag{6}$$

where $w_{i,X_{parent}}$ is the weight for person i at the parent class, which equals this person's posterior probability of belonging to the parent class concerned. So, building an LCGT involves estimating a series of LCG models using weighted data sets.

To see how the weights $w_{i,X_{parent}}$ are constructed, let us first look at the posterior class membership probabilities for the child nodes, conditional on the corresponding parent node. Assuming a split is accepted, the posteriors are obtained as follows:

$$P(X_{child} = k | \mathbf{y}_i; X_{parent}) = \frac{P(X_{child} = k | X_{parent}) \prod_{t=1}^{T_i} f(y_{it} | X_{child} = k, X_{parent})}{P(\mathbf{y}_i | X_{parent})}. \tag{7}$$

As proposed by Van Der Palm et al. (2016), we use a proportional split based on these posterior class membership probabilities for the K child nodes conditional on the parent node, denoted by $k = 1, 2, ..., K$. If a split in two classes is performed, the weights for the two newly formed classes at the next level are obtained as follows:

$$w_{i,X_{child}=1} = w_{i,X_{parent}} P(X_{child} = 1 | \mathbf{y}_i; X_{parent}) \tag{8}$$

$$w_{i,X_{child}=2} = w_{i,X_{parent}} P(X_{child} = 2 | \mathbf{y}_i; X_{parent}). \tag{9}$$

In other words, a weight for individual i at a particular node equals the weight at the parent node times the posterior probability of belonging to the child node concerned conditional on belonging to the parent node. As an example, the weights $w_{i,X_1=2}$ used for investigating a possible split of class $X_1 = 2$ are constructed as follows:

$$w_{i,X_{12}} = w_{i,\, X=1} P(X_1 = 2 | \mathbf{y}_i, X = 1), \tag{10}$$

where in turn $w_{i,X=1} = P(X = 1 | \mathbf{y}_i)$. This implies:

$$w_{i,X_{12}} = P(X = 1 | \mathbf{y}_i) P(X_1 = 2 | \mathbf{y}_i, X = 1), \tag{11}$$

which shows that a weight at Level 2 is in fact a product of two posterior class membership probabilities.

Construction of an LCGT can be performed using standard software for latent class analysis, namely by running a series of latent class models with the appropriate weights. After each accepted split a new data set is constructed and the procedure repeats itself. We developed an R routine in which this process is fully automated. It calls the Latent GOLD program (Vermunt & Magidson, 2013) in batch mode to estimate one- and two-class models, evaluates whether a split should be made, and keeps track of the weights when a split is accepted. In addition, it creates various graphical displays, which facilitates the interpretation of the LCGT (see among others Figure 2). A novel graphical display is a tree depicting the class-specific growth curves for each of the classes in the tree and the newly formed child classes. In the trees, the name of a child class equals the name of the parent class plus an additional digit, a 1 or a 2. To prevent the structure of the tree from be affected by label switching resulting from the fact that the order of the newly formed classes depends on the random starting values, when building the LCGT we locate the larger class at the left branch with number 1 and the smaller class at the right branch with number 2.

Statistics for Building and Evaluating the LCGT

Different types of statistics can be used to determine whether a split should be accepted or rejected. Here, we will use the BIC (Schwarz, 1978), which is defined as follows:

$$BIC = -2\log L(\theta; \mathbf{y}, X_{parent}) + \log(N)P, \tag{12}$$

where $logL(.)$ represents the log-likelihood at the parent node concerned, N is the total sample size, and P is the number of parameters of the model at hand. Thus, a split is performed if at a parent node concerned the BIC for the two-class model is lower than the one of the one-class model. Note that using a less strict criterion (e.g., AIC) will yield the same splits as the BIC, but possible also additional splits, and thus a larger tree.

Special attention needs to be dedicated to the first split at the root node of the tree, in which one picks up the most dominant features in the data. In many situations, a binary split at the root might be too much of a simplification, and one would prefer allowing for more than two classes in the first split. For this purpose, we cannot use the usual criteria like AIC or BIC, as this would boil down to again using a standard LCG model. Instead, for the decision to use more than two classes at the root node, we propose looking at the relative improvement of fit compared to the improvement between the one- and two-class models. When using the log-likelihood value as the fit measure, this implies assessing the increase in log-likelihood between, say, the two- and three-class models and comparing it to the increase between the one- and two-class models. More explicitly, the relative improvement between models with K and K + 1 classes ($RI_{K,K+1}$) can be computed as:

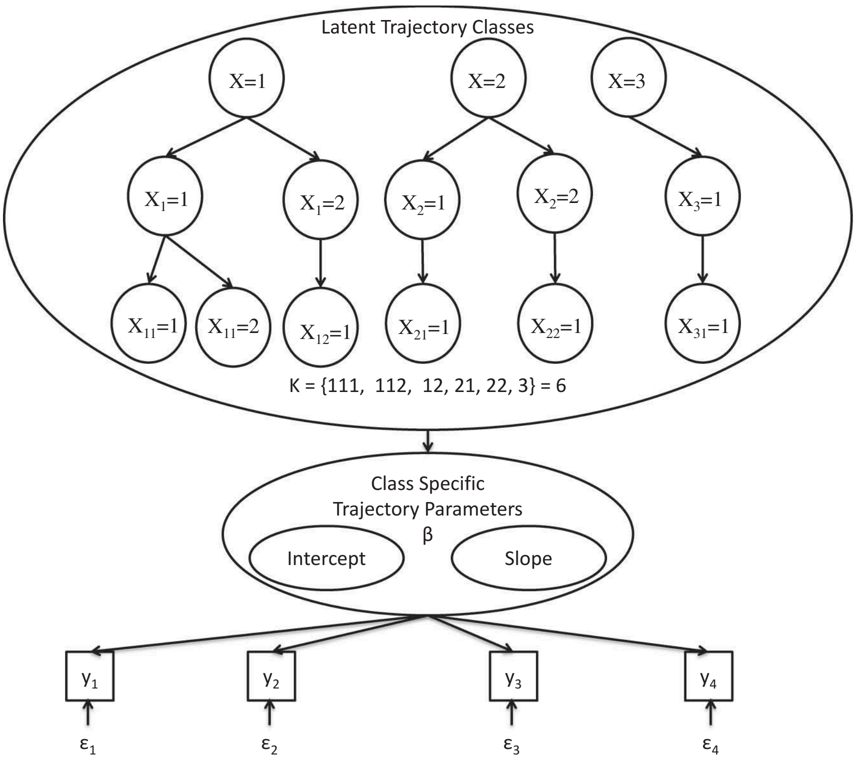


FIGURE 2 Graphical example of a latent class growth tree model with a root of three classes.

$$RI_{K,K+1} = \frac{\log L_{K+1} - \log L_K}{\log L_2 - \log L_1}, \tag{13}$$

which yields a number between 0 and 1, where a small value indicates that the *K*-class model can be used as the first split, and a larger value indicates that the tree might improve with an additional class at the root of the tree. Note that instead of an increase in log-likelihood, in Equation 13 one could use other measures of improvement of fit, such as the decrease of the BIC or the AIC. Scree plots depicting the difference in log-likelihood (or BIC or AIC) for models with one class difference can also be used to judge whether the relative improvement is large, as illustrated in the empirical examples presented later.

The *BIC* and $RI_{K,K+1}$ statistics are used to determine whether and how splits should be performed. However, often we are also interested in evaluating the quality of splits in terms of the amount of separation between the newly formed classes; that is, to determine how different the classes are. In other words, is a split substantively important?. This is also relevant if one would like to assign individuals to the classes resulting from an LCGT. Note that the assignment of individuals to the two child classes is more certain when the larger of the posterior probabilities $P(X_{child} = k|\mathbf{y}_i; X_{parent})$ is closer to 1. A measure to express this is the entropy; that is,

$$Entropy(X_{child}|\mathbf{y}) = \sum_{i=1}^{N} w_{i|X_{parent}} \sum_{k=1}^{2} -P(X_{child} = k|\mathbf{y}_i; X_{parent}) \\ logP(X_{child} = k|\mathbf{y}_i; X_{parent}). \tag{14}$$

Typically $Entropy(X_{child}|\mathbf{y})$ is rescaled to lie between 0 and 1 by expressing it in terms of the reduction compared to $Entropy(X_{child})$, which is the entropy computed using the unconditional class membership probabilities $P(X_{child} = k|X_{parent})$. This so-called $R^2_{Entropy}$ is obtained as follows:

$$R^2_{Entropy} = \frac{Entropy(X_{child}) - Entropy(X_{child}|\mathbf{y})}{Entropy(X_{child})} \tag{15}$$

The closer $R^2_{Entropy}$ is to one, the better the separation between the child classes in the split concerned.

EMPIRICAL EXAMPLES

The proposed LCGT methodology will be illustrated by the analyses of two longitudinal data sets. The data set in the first example contains a yearly dichotomous response on drug use collected using a panel design. The second data set contains an ordinal mood measure, recorded using an experience sampling design with eight measures per day during 1 week. The two

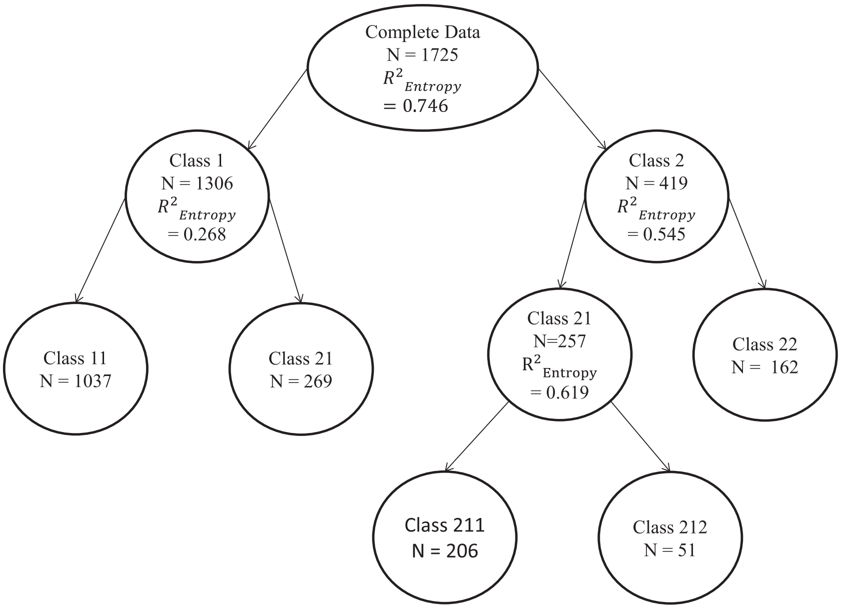


FIGURE 3 Layout, class sizes, and $R^2_{Entropy}$ of every split of a latent class growth tree with a root of two classes on drug use.

data sets illustrate LCGT analyses, differing in the number of classes at their root node. For both examples, the quality of the splits will also be evaluated using the entropy-based R^2.

Example 1: Drug Use

The first data set stems from the National Youth Survey (Elliot, Huizinga, & Menard, 1989). It contains nine waves, from 1976 to 1980 yearly and from 1980 to 1992 with 3-year intervals. The age at the first wave of the 1,725 respondents (53% men, 47% women) varied between 11 and 17 years. We use age at the panel wave concerned as the time variable, which takes on values ranging from age 11 to 33. Each respondent has been observed at most nine times (on average 7.93 times). Although the presence of (systematic) missing values could influence the results, as is typically done in growth modeling, in this illustrative example we assume the missing data to be missing at random. The dichotomous dependent variable of interest in our example will be whether the respondent used drugs or not during the past year.

The LCGT solution obtained with a third-order polynomial was very similar to the one with a second-order polynomial, both in terms of the shapes of the growth curves and the class assignments. In the LCGT starting with three classes, the class allocations differed for only 12 out of the 1,725 respondents. The tree structure and the class sizes at the splits[1] are presented in Figure 3. As can be seen, there are four binary splits, which result in a total of five latent classes at the end nodes.

To determine whether it would be better to increase the number of classes at the root of the tree, we can look at the relative improvement in fit of models with more than two classes according to the likelihood, BIC, and AIC as reported in Table 1. As can be seen, the relative improvement with a third class is around 10%. As this is quite low, we retain the tree with a binary split at the root.

To interpret the encountered classes, the growth curves can be plotted for the two newly formed classes at each node of the tree. This is displayed in Figure 4. As can be seen, the first split results in a class with a low probability to use drugs (Class 1) and a class with a high probability to use drugs (Class 2). Subsequently both of these classes are split further. Class 1 is split into Class 11 with a very low probability of using drugs (on average 0.01%) and Class 12 with a low probability during the first few years, but with

TABLE 1
Fit Statistics of a Traditional Latent Class Growth Model With One to Six Classes

	Log L	*P*	*BIC*	*AIC*	RI_{logL}	RI_{BIC}	RI_{AIC}
1	−5,089	3	10,200	10,183			
2	−4,246	7	8,543	8,505			
3	−4,156	11	8,394	8,334	0.106	0.090	0.102
4	−4,086	15	8,284	8,202	0.083	0.067	0.079
5	−4,046	19	8,233	8,129	0.048	0.031	0.043
6	−4,028	23	8,228	8,102	0.021	0.003	0.016

Note. BIC = Bayesian information criterion; AIC = Akiake's information criterion.

[1] Every split should sum up to the class size of its parent node. However, because the allocation is carried out on the basis of the posterior probabilities, the class sizes are not integers. For convenience, these numbers have been rounded, which causes slight deviations where the sum of two child nodes does not exactly add up to the parent node.

a slight increase from age 20 to 33. Class 2 is split into Class 21 and Class 22, which mainly differ in the moment at which the probability of drug use is the highest: Respondents of Class 21 start using drugs a few years earlier than respondents of Class 22. Finally, Class 21 is split further, where Class 211 has a moderate probability (around 0.6) of using drugs at an early age, but this probability also quickly declines. Class 212 has a very high probability (around 0.95) to start using drugs at an early age and this probability stays quite constant up to age 25.

The $R^2_{Entropy}$ values confirm what could also be seen from the depicted growth curves: The first split on the complete data set shows a large difference between the two classes with $R^2_{Entropy}$ of 0.746. Furthermore, Classes 11 and 12 are quite similar with $R^2_{Entropy}$ of 0.268, whereas the differences between Classes 21 and 22 and between Classes 211 and 212 are substantial (the $R^2_{Entropy}$ values are 0.545 and 0.619, respectively). Hence, after the first split, the branch of Class 2 contains more important additional differences than the one of Class 1. As an additional check we ran an LCGT using only the data from respondents with at least eight waves observed and obtained very similar results.

Example 2: Mood Regulation

The second data set stems from a momentary assessment study by Crayen, Eid, Lischetzke, Courvoisier, and Vermunt (2012). It contains eight mood assessments per day during a period of 1 week among 164 respondents (88 women and 76 men, with a mean age of 23.7, *SD* = 3.31). Respondents answered a small number of questions on a handheld device at pseudorandom signals during their waking hours. The delay between adjacent signals could vary between 60 and 180 min (*M* = 100.24 min, *SD* = 20.36, min = 62, max = 173). Responses had to be made within a 30-min time window after the signal, and were otherwise counted as missing. On average, the 164 participants responded to 51 (of 56) signals (*M* = 51.07 signals, *SD* = 6.05, min = 19, max = 56). In total, there were 8,374 nonmissing measurements. The missing data are assumed to be missing at random.

At each measurement occasion, participants rated their momentary mood on an adapted short version of the Multidimensional Mood Questionnaire (MMQ). Instead of the original monopolar mood items, a shorter bipolar version was used to fit the need for brief scales. Four items assessed pleasant or unpleasant mood (happy–unhappy, content–discontent, good–bad, and well–unwell). Participants rated how they momentarily feel on a 4-point bipolar intensity scale (e.g., *very unhappy, rather unhappy, rather happy, very happy*). For this analysis, we focus on the item well–unwell. Preliminary analysis of the response category frequencies showed that the lowest category (i.e., very unwell) was only chosen in approximately 1% of all occasions. Therefore the two lower categories were collapsed together into one unwell category. The following analysis is based on the recoded item with three categories (Crayen et al., 2012).

For the analysis, we used an LCG model based on an ordinal logit model. The time variable was the time during the day, meaning that we modeled the mood change during the day. Application of a traditional LCG model, using the BIC for model selection, resulted in seven classes with cubic growth curves (see Table 2). The cubic curves seemed to be flexible enough, as adding a quartic term did not improve model fit. The class-specific growth curves are displayed in Figure 5. As

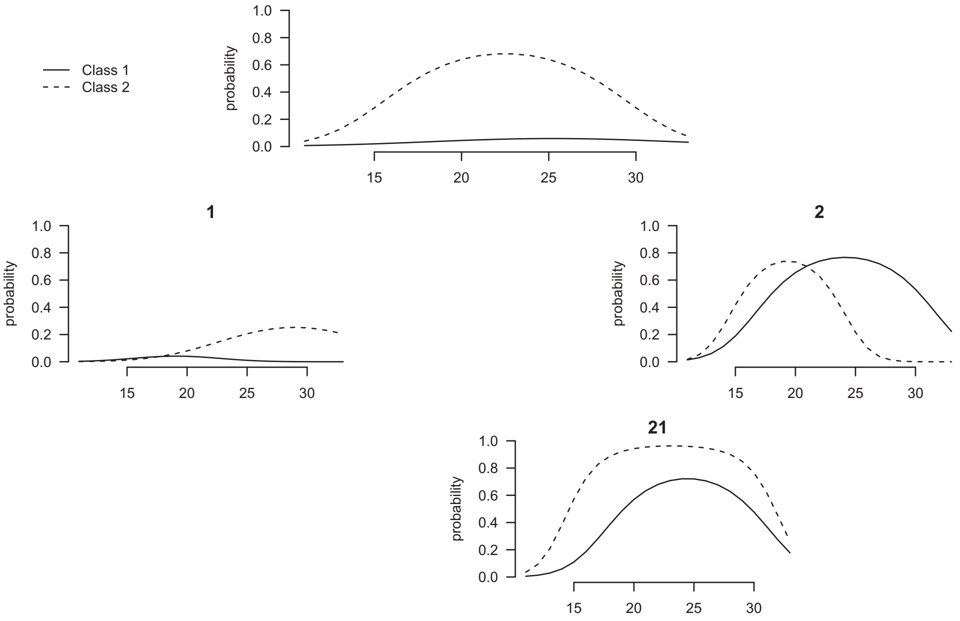


FIGURE 4 Latent class growth three with a root of two classes on drug use over age.

TABLE 2
Likelihood, Number of Parameters, Bayesian Information Criterion (BIC) and Relative Improvement of the Likelihood and BIC of a Traditional Latent Class Growth Model With One to Six Classes

	Log L	*P*	*BIC*	*AIC*	RI_{logL}	RI_{BIC}	RI_{AIC}
1	−7,199	4	14,424	14,408			
2	−6,741	9	13,538	13,504			
3	−6,578	14	13,244	13,191	0.355	0.333	0.347
4	−6,516	19	13,149	13,077	0.137	0.107	0.126
5	−6,471	24	13,091	13,001	0.097	0.065	0.085
6	−6,443	29	13,064	12,956	0.062	0.030	0.050
7	−6,424	34	13,058	12,931	0.040	0.007	0.028
8	−6,414	39	13,068	12,922	0.022	−0.012	0.009

Note. AIC = Akaike's information criterion.

can be seen, there is a clear high and a clear low class, whereas the remaining five "average" classes are rather similar to one another. This indicates that an LCGT might yield a simpler interpretation of the classes detected for this data set.

The LCGT model obtained with a root of two classes is quite large, with in total seven binary splits, resulting in a total of eight latent classes. A large tree already indicates that a larger number of classes at the root of the tree might be appropriate. Moreover, based on the relative improvement of the log-likelihood, BIC, and AIC (Table 2), it seems sensible to increase the number of classes at the root of the tree. A scree plot of the relative change in log-likelihood, BIC, and AIC also shows that after three classes the relative gain is quite small for both measures (Figure 6).

The layout and size of the LCGT with three root classes can be seen in Figure 7 and its growth curve plots in Figure 8. The growth plots show that at the root of the tree, the three different classes all improve their mood during the day. They differ in their overall mood level, with Class 3 having the lowest and Class 2 the highest overall score. Moreover, Class 1 seems to be more consistently increasing than the other two classes.

These three classes can be split further. Class 1 splits into two classes with both an average score around one, Class 11 just above and Class 12 just below. Moreover, the increase in Class 11 is larger than in Class 12. The split of Class 2 results in Class 21 consisting of respondents with a very good mood in the morning, a quick decrease until midday, and a subsequent increase. In general the mean score of Class 21 is high relative to the other classes. Class 22 starts with an average mean score and subsequently only increases. The splitting of Class 3 results in two classes with a below average mood. Both classes increase, Class 31 mainly in the beginning and Class 32 mainly at the end of the day.

The $R^2_{Entropy}$ of the different splits is quite high. The root of the tree has $R^2_{Entropy}$ of 0.889, and $R^2_{Entropy}$ of the subsequent splits is 0.734, 0.932, and 0.897, respectively. This indicates that the differences between Subclasses 21 and 22 are larger

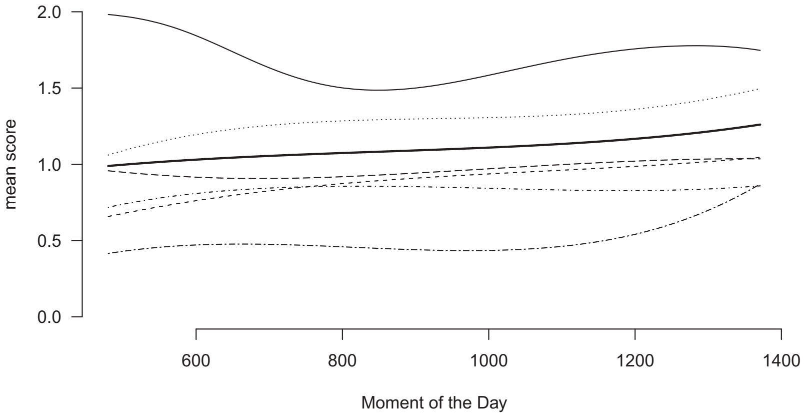


FIGURE 5 Profile plot of a latent class growth model on mood regulation with seven classes.

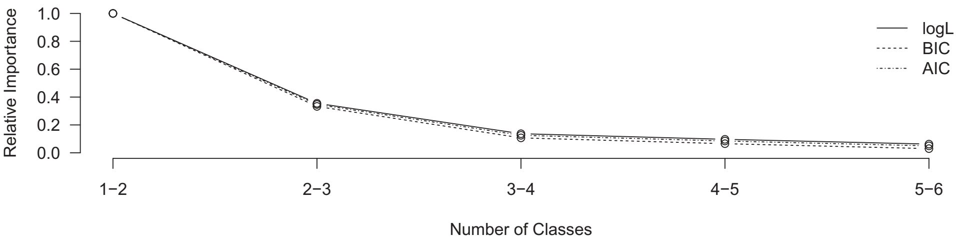


FIGURE 6 Scree plot of the difference in likelihood and Bayesian information criterion (BIC) of successive latent class growth models for the data on mood regulation. *Note*: logL = log-likelihood; AIC = Akaike's information criterion.

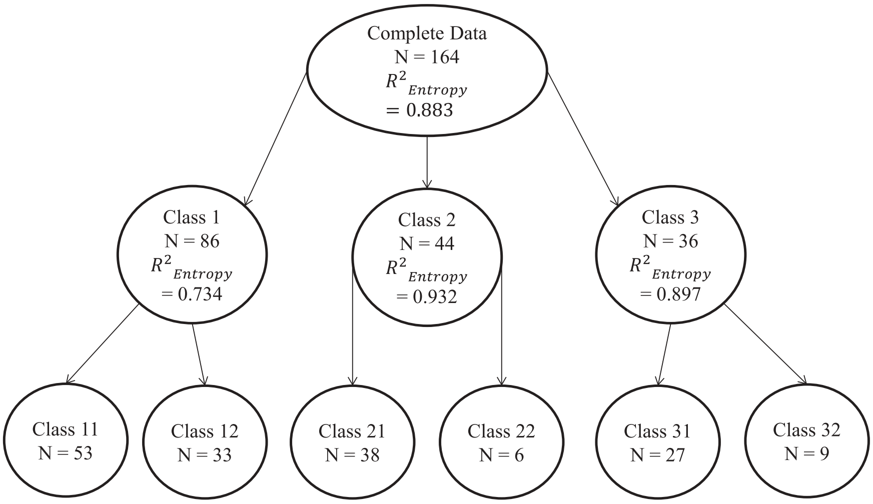


FIGURE 7 Layout, class sizes, and $R^2_{Entropy}$ of every split of a latent class growth three with a root of three classes on mood regulation during the day.

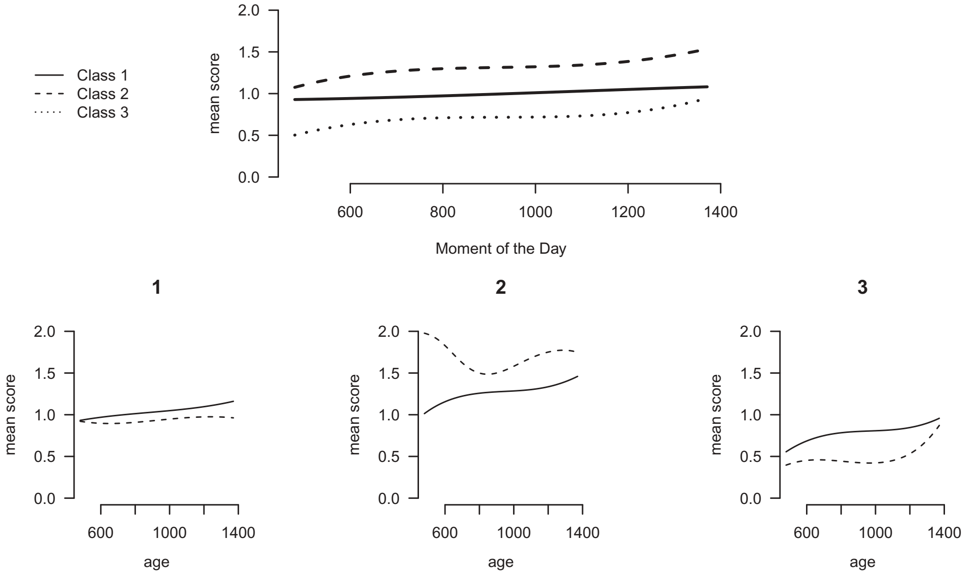


FIGURE 8 Latent class growth tree with a root of three classes of mood regulation during the day.

than those between Subclasses 31 and 32, whereas Classes 11 and 12 differ the least.

To compare the results obtained with the traditional LCG model with those of the LCGT we looked at the modal class assignments. Table 3 cross-tabulates the class allocations of the two methods. It can be seen that some classes are quite some similar; for instance, traditional Classes 3, 6, and 7 contain fairly the same respondents as the tree-based Classes 3, 4, and 6, respectively. This is not the case when using a tree with a root of two classes, as can be seen in Table 4. The differences between the remaining classes of an LCGT with a root of three classes and the seven-class LCG model are due to the first split of the tree. This can be seen in Table 5, which cross-tabulates class allocation from a seven-class and a three-class LCG model (the first level of the preferred LCGT). This also illustrates the problem of traditional LCG models, as the seven classes cannot be seen as resulting from a further splitting of these three classes. In contrast, this is exactly what is achieved with the LCGT approach, as can be seen in Table 6: Pairs of classes of the six-class model form a class of the three-class model.[2]

[2] This is true for all respondents, except for one that is in Class 1 at the first level of the tree and Class 5 at the second level of the tree. Respondents can still switch between branches of an LCGT, but this is much more restricted than in traditional LCG models.

TABLE 3
A Cross Table Showing the Differences and Similarities in Modal Assignment of a Traditional Latent Class Growth Model With Seven Classes (Rows) and a Latent Class Growth Tree With a Root of Three Classes (Columns)

	1	2	3	4	5	6
1	37	0	3	0	0	0
2	17	0	2	0	14	0
3	0	0	32	0	0	0
4	1	11	0	0	13	0
5	0	21	0	0	0	0
6	0	0	0	0	0	9
7	0	0	0	5	0	0

TABLE 4
A Cross Table Showing the Differences and Similarities in Modal Assignment of a Traditional Latent Class Growth Model With Seven Classes (Rows) and a Latent Class Growth Tree With a Root of Two Classes (Columns)

	1	2	3	4	5	6	7	8
1	13	13	0	0	0	0	14	0
2	14	0	0	16	0	3	0	0
3	0	0	0	0	0	25	3	4
4	8	0	8	9	0	0	0	0
5	1	0	20	0	0	0	0	0
6	0	0	0	2	7	0	0	0
7	0	0	0	0	0	0	0	5

TABLE 5
A Cross Table Showing the Differences and Similarities in Modal Assignment of a Latent Class Growth Tree With Six Classes (Rows) and a Traditional Latent Class Growth Model With Three Classes (Columns)

	1	2	3
1	37	3	0
2	17	2	14
3	0	32	0
4	13	0	12
5	21	0	0
6	0	0	9
7	0	5	0

TABLE 6
A Cross Table Showing the Differences and Similarities in Modal Assignment of a Traditional Latent Class Growth Model With Three Classes (Rows) and a Latent Class Growth Tree With a Root of Three Classes (Columns)

	1	2	3
1	55	0	0
2	32	0	0
3	0	37	0
4	0	5	0
5	1	0	26
6	0	0	9

DISCUSSION

LCG models are used by researchers who wish to identify (unobserved) subpopulations with different growth trajectories using longitudinal data. However, often the number of latent classes encountered is rather large, making interpretation of the results difficult. Moreover, because solutions with different numbers of classes are unrelated, a substantive comparison of models with different numbers of classes is not possible, which is especially problematic when different model selection criteria point at a different optimal number of classes. To resolve these issues, we proposed using LCGT models in which the identification of the latent classes is done in a sequential manner. The constructed hierarchical tree will show the most important distinctions in growth trajectories in the first splits, and more detailed distinctions in latter splits. Although we primarily used binary splits, we also showed how to decide about larger splits using relative improvement of fit measures. The latter is mainly of interest at the root of the tree. The proposed LCGT algorithm and graphical displays that are available as R code were illustrated with two empirical examples. The two illustrative examples showed that easily interpretable solutions are obtained using our new procedure. The fact that we impose a tree structure is both a strong point (it simplifies interpretation of the classes) and also a limitation in the sense the method will be less useful if there are no classes that are similar and that can thus be seen as children of the same parent. This is partially dealt with by allowing for a larger number of classes at the initial split, where the classes are not assumed to be hierarchically linked. In future research we want to investigate via simulation studies how well the LCGT modeling approach can pick up known class structures. As with other clustering methods, the LCG model is typically used as an exploratory clustering tool. In the context of exploratory clustering, it has been shown that even when the true class structure is not hierarchical, hierarchical clustering methods could perform very well (Ghattas, Michel, & Boyer, 2017).

The fact that we impose a tree structure is both a strong point (it simplifies interpretation of the classes) and a limitation in the sense the method will be less useful if there are no classes that are similar and that can thus be seen as children of the same parent. This is partially dealt with by allowing for a larger number of classes at the initial split, where the classes are not assumed to be hierarchically linked. In future research we want to investigate via simulation studies how well the LCGT modeling approach can pick up known class structures. As other clustering methods, the LCG model is typically used as an exploratory clustering tool. In the context of exploratory clustering, it has been shown that even when the true class structure is not hierarchical, hierarchical clustering methods might perform very well (Ghattas et al., 2017).

Various extensions and variants of the proposed procedure are possible and worth studying in more detail. Whereas in this article we restricted ourselves to LCGTs with only binary splits after the split at the root of the tree, and also at the second and next levels, it might be of interest to use larger split sizes, which could result in a tree with different split sizes within branches. Because the size of the splits could strongly affect the structure of the constructed LCGT, we recommend deciding this separately per split rather than using a fully automated procedure. Note that at lower branches of a tree there is also more substantive information available to guide the decision regarding the number of child classes.

The BIC was used to decide whether or not to split a class, as it is the most commonly used criterion and has been shown to perform well for standard latent class and LCG analysis (Nylund et al., 2007). However, other measures could be used as well, where their strictness will influence the likelihood of starting a new branch within a tree. For instance, the AIC is more lenient than the BIC and would therefore result in a larger tree with more splits, but also containing the splits of the BIC-based tree. It should be noted that the relative improvement in fit is rather similar for different fit measures, as the difference in penalty terms are canceled out. Hence, the decision criterion used mostly affects the bottom part of the tree and much less the decision regarding the number of initial classes. Therefore, the exact choice of a criterion depends on the required specificity of the encountered growth trajectories, where a less strict criterion could be used if one wishes to see more specific classes at the bottom of the tree. Note furthermore that other alterations are possible, such as a BIC with a sample size adjustment for every split (Sclove, 1987). Other criteria, such as the minimum class size, can be incorporated in the decision as to whether to perform a split. Note that sometimes classes with a very small size might point to the presence of outliers, and could thus be useful to detect.

Although LCG models are becoming very popular among applied researchers, the use of these models is not easy at all (Van De Schoot, Sijbrandij, Winter, Depaoli, & Vermunt, 2017). We hope that the proposed LCGT methodology will simplify the detection and interpretation of underlying growth trajectories. This, of course, does not mean that the standard LCG model is not useful anymore. In practice, a researcher might start with a standard LCG analysis, and switch to our LCGT approach when encountering difficulties in deciding about the number of classes or interpreting the differences between a possibly large number of classes.

FUNDING

This work was supported by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek [453-10-002].

REFERENCES

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

Crayen, C., Eid, M., Lischetzke, T., Courvoisier, D. S., & Vermunt, J. K. (2012). Exploring dynamics in mood regulation: Mixture latent Markov modeling of ambulatory assessment data. *Psychosomatic Medicine*, *74*, 366–376. doi:10.1097/PSY.0b013e31825474cb

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological), 39*, 1–38.

Elliott, D.S., Huizinga, D., and Menard, S.(1989). *Multiple problem youth: Delinquency, substance use, and mental health problems*. New York: Springer-Verlag

Everitt, B. S., Landau, S., Leese, M., & Stahl, D. (2011). Hierarchical clustering. In *Cluster analysis* (5th ed., pp. 71–110). Chichester: John Wiley & Sons.

Francis, B., Elliott, A., & Weldon, M. (2016). Smoothing group-based trajectory models through b-splines. *Journal of Developmental and Life-Course Criminology*, *2*, 113–133. doi:10.1007/s40865-016-0025-6

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1). Berlin, Germany: Springer.

Ghattas, B., Michel, P., & Boyer, L. (2017). Clustering nominal data using unsupervised binary decision trees: Comparisons with the state of the art methods. *Pattern Recognition*, *67*, 177–185. doi:10.1016/j.patcog.2017.01.031

Grimm, K. J., Ram, N., & Estabrook, R. (2017). *Growth modeling: Structural equation and multilevel modeling approaches*. New York, NY: Guilford.

Jones, B. L., Nagin, D. S., & Roeder, K. (2001). A SAS procedure based on mixture models for estimating developmental trajectories. *Sociological Methods & Research*, *29*, 374–393. doi:10.1177/0049124101029003005

Jung, T., & Wickrama, K. (2008). An introduction to latent class growth analysis and growth mixture modeling. *Social and Personality Psychology Compass*, *2*, 302–317. doi:10.1111/j.1751-9004.2007.00054.x

Lo, Y., Mendell, N. R., & Rubin, D. B. (2001). Testing the number of components in a normal mixture. *Biometrika*, *88*, 767–778. doi:10.1093/biomet/88.3.767

MacCallum, R. C., & Austin, J. T. (2000). Applications of structural equation modeling in psychological research. *Annual Review of Psychology*, *51*, 201–226. doi:10.1146/annurev.psych.51.1.201

McLachlan, G., & Peel, D. (2004). *Finite mixture models*. New York, NY: Wiley.

Muthén, B. (2003). Statistical and substantive checking in growth mixture modeling: Comment on Bauer and Curran (2003). *Psychological Methods*, *8*, 384–393.

Muthén, B. (2004). Latent variable analysis. In D. Kaplan (Ed.), *The Sage handbook of quantitative methodology for the social sciences* (pp. 345–368). Thousand Oaks, CA: Sage Publication.

Nagin, D. S. (2005). *Group-based modeling of development*. Cambridge, MA: Harvard University Press.

Nagin, D. S., & Land, K. C. (1993). Age, criminal careers, and population heterogeneity: Specification and estimation of a nonparametric, mixed Poisson model. *Criminology*, *31*, 327–362. doi:10.1111/crim.1993.31.issue-3

Nesselroade, J. R. (1991). Interindividual differences in intraindividual change. In L. M. Collins & J. L Horn (Eds.), *Best methods for the analysis of change: Recent advances, unanswered questions, future directions* (pp. 92–105). Washington, DC: American Psychological Association.

Nylund, K. L., Asparouhov, T., & Muthén, B. O. (2007). Deciding on the number of classes in latent class analysis and growth mixture modeling:

A Monte Carlo simulation study. *Structural Equation Modeling*, *14*, 535–569. doi:10.1080/10705510701575396

Raudenbush, S. W., & Bryk, A. S. (2002). *Hierarchical linear models: Applications and data analysis methods* (Vol. 1). Thousand Oaks, CA: Sage.

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

Sclove, S. L. (1987). Application of model-selection criteria to some problems in multivariate analysis. *Psychometrika*, *52*, 333–343. doi:10.1007/BF02294360

Tofighi, D., & Enders, C. K. (2008). Identifying the correct number of classes in growth mixture models. In G. R. Hancock & K. M. Samuelsen (Eds.), *Advances in latent variable mixture models* (pp. 317–341). Greenwich, CT: Information Age.

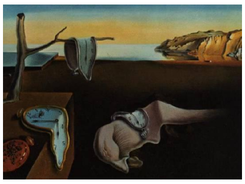
Van De Schoot, R., Sijbrandij, M., Winter, S. D., Depaoli, S., & Vermunt, J. K. (2017). The GRoLTS-checklist: Guidelines for reporting on latent trajectory studies. *Structural Equation Modeling*, *24*, 451–467.

van Den Bergh, M., Schmittmann, V. D., & Vermunt, J. K. (2017). Building latent class trees, with an application to a study of social capital. *Methodology,13*, 13–22.

van Der Palm, D. W., Van der Ark, L. A., & Vermunt, J. K. (2016). Divisive latent class modeling as a density estimation method for categorical data. *Journal of Classification, 33*(1), 52–72.

Vermunt, J. K. (2007). Growth models for categorial response variables: Standard, latent-class, and hybrid approaches. In K. Van Montfort, H. Oud, & A. Satorra (Eds.), *Longitudinal models in the behavioral and related sciences* (pp. 139–1580). Mahwah, NJ: Erlbaum.

Vermunt, J. K., & Magidson, J. (2013). *Technical guide for Latent GOLD 5.0: Basic, advanced, and syntax*. Belmont, MA: Statistical Innovations Inc.



Journal of Statistical Software

June 2011, Volume 42, Issue 10. http://www.jstatsoft.org/

poLCA: An R Package for Polytomous Variable Latent Class Analysis

Drew A. Linzer
Emory University

Jeffrey B. Lewis
University of California,
Los Angeles

Abstract

poLCA is a software package for the estimation of latent class and latent class regression models for polytomous outcome variables, implemented in the R statistical computing environment. Both models can be called using a single simple command line. The basic latent class model is a finite mixture model in which the component distributions are assumed to be multi-way cross-classification tables with all variables mutually independent. The latent class regression model further enables the researcher to estimate the effects of covariates on predicting latent class membership. **poLCA** uses expectation-maximization and Newton-Raphson algorithms to find maximum likelihood estimates of the model parameters.

Keywords: latent class analysis, latent class regression, polytomous, categorical, concomitant.

1. Introduction

Latent class analysis is a statistical technique for the analysis of multivariate categorical data. When observed data take the form of a series of categorical responses—as, for example, in public opinion surveys, individual-level voting data, studies of inter-rater reliability, or consumer behavior and decision-making—it is often of interest to investigate sources of confounding between the observed variables, identify and characterize clusters of similar cases, and approximate the distribution of observations across the many variables of interest. Latent class models are a useful tool for accomplishing these goals.

The latent class model seeks to stratify the cross-classification table of observed (or, "manifest") variables by an unobserved ("latent") unordered categorical variable that eliminates all confounding between the manifest variables. Conditional upon values of this latent variable, responses to all of the manifest variables are assumed to be statistically independent;

an assumption typically referred to as "conditional" or "local" independence. The model, in effect, probabilistically groups each observation into a "latent class," which in turn produces expectations about how that observation will respond on each manifest variable. Although the model does not automatically determine the number of latent classes in a given data set, it does offer a variety of parsimony and goodness of fit statistics that the researcher may use in order to make a theoretically and empirically sound assessment.

Because the unobserved latent variable is nominal (membership of a class), the latent class model is actually a type of finite mixture model. The component distributions in the mixture are cross-classification tables of equal dimension to the observed table of manifest variables, and, following the assumption of conditional independence, the frequency in each cell of each component table is simply the product of the respective class-conditional marginal frequencies (the parameters estimated by the latent class model are the proportion of observations in each latent class, and the probabilities of observing each response to each manifest variable, conditional on latent class). A weighted sum of these component tables forms an approximation (or, density estimate) of the distribution of cases across the cells of the observed table. Observations with similar sets of responses on the manifest variables will tend to cluster within the same latent classes. The model may also be fit to manifest variables that are ordinal, but they will be treated as nominal. In practice, this does not usually restrict analyses in any meaningful way.

An extension of this basic model permits the inclusion of covariates to predict latent class membership. Whereas in the basic model, every observation has the same probability of belonging to each latent class prior to observing the responses to the manifest variables, in the more general latent class "regression" model, these prior probabilities vary by individual as a function of some set of independent (or, "concomitant") variables.

Examples of latent class models in political science include McCutcheon (1985), Feick (1989), Breen (2000), Hill and Kriesi (2001a,b), Blaydes and Linzer (2008), and Linzer (2011). The latent class model is similar to the latent trait model widely used for the estimation of voter and legislator "ideal points" (e.g., Clinton, Jackman, and Rivers 2004) in that both models assume the presence of an underlying, unobserved latent variable to explain patterns among observed variables. Yet whereas the latent trait model assumes that the latent variable is continuous, the latent class model assumes that the latent variable is categorical. Moreover, unlike the ideal point model, the latent class model requires no assumptions about respondent utility functions, utility maximization, or rationality.

poLCA is the most complete and most user-friendly package for the estimation of latent class models and latent class regression models in R (Linzer and Lewis 2011; R Development Core Team 2010). The package is available from both the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=poLCA` and the **poLCA** project Web site at `http://userwww.service.emory.edu/~dlinzer/poLCA`. Other R functions for the estimation of latent class models include `lca` in package **e1071**, `gllm` in package **gllm**, and `randomLCA` in package **randomLCA**, but these can only estimate the basic model for dichotomous outcome variables (Duffy 2010; Beath 2011; Dimitriadou, Hornik, Leisch, Meyer, and Weingessel 2011). Package **flexmix** has the capacity to estimate latent class regression models, but not without considerable extra programming effort on the part of the user (Grün and Leisch 2008).

Note that there is occasionally some confusion over the term "latent class regression" (LCR); in practice it can have two meanings. In **poLCA**, LCR models refer to latent class models in which the probability of latent class membership is predicted by one or more covariates. In other contexts, however, LCR is used to refer to regression models in which the dependent variable is partitioned into latent classes as part of estimating the regression model. It is a way to simultaneously fit more than one regression to the data when the latent data partition is unknown. The `regmix` function in package **fpc** (Hennig 2010) will estimate this other type of LCR model, as will the `flexmix` function in package **flexmix** (Leisch 2004; Grün and Leisch 2008). Because of these terminology issues, the LCR models estimated using **poLCA** are sometimes termed "latent class models with covariates" or "concomitant-variable latent class analysis," both of which are accurate descriptions of this model.

2. Latent class models

The basic latent class model is a finite mixture model in which the component distributions are assumed to be multi-way cross-classification tables with all variables mutually independent. This model was originally proposed by Lazarsfeld (1950) under the name "latent structure analysis." Chapter 13 in Agresti (2002) details the connection between latent class models and finite mixture models.

2.1. Terminology and model definition

Suppose we observe J polytomous categorical variables (the "manifest" variables), each of which contains K_j possible outcomes, for individuals $i = 1, \ldots, N$. The manifest variables may have different numbers of outcomes, hence the indexing by j. Denote as Y_{ijk} the observed values of the J manifest variables such that $Y_{ijk} = 1$ if respondent i gives the kth response to the jth variable, and $Y_{ijk} = 0$ otherwise, where $j = 1, \ldots, J$ and $k = 1, \ldots, K_j$.

The latent class model approximates the observed joint distribution of the manifest variables as the weighted sum of a finite number, R, of constituent cross-classification tables. R is fixed prior to estimation on the basis of either theoretical reasons or model fit; this issue is addressed in greater detail in Section 2.4 below. Let π_{jrk} denote the class-conditional probability that an observation in class $r = 1, \ldots, R$ produces the kth outcome on the jth variable. Within each class, for each manifest variable, therefore, $\sum_{k=1}^{K_j} \pi_{jrk} = 1$. Further denote as p_r the R mixing proportions that provide the weights in the weighted sum of the component tables, with $\sum_r p_r = 1$. The values of p_r are also referred to as the "prior" probabilities of latent class membership, as they represent the unconditional probability that an individual will belong to each class before taking into account the responses Y_{ijk} provided on the manifest variables.

The probability that an individual i in class r produces a particular set of J outcomes on the manifest variables, assuming conditional independence of the outcomes Y given class memberships, is the product

$$f(Y_i; \pi_r) = \prod_{j=1}^{J} \prod_{k=1}^{K_j} (\pi_{jrk})^{Y_{ijk}}. \tag{1}$$

The probability density function across all classes is the weighted sum

$$\mathsf{P}(Y_i | \pi, p) = \sum_{r=1}^{R} p_r \prod_{j=1}^{J} \prod_{k=1}^{K_j} (\pi_{jrk})^{Y_{ijk}}. \tag{2}$$

The parameters estimated by the latent class model are p_r and π_{jrk}.

Given estimates $\hat{p}_r$ and $\hat{\pi}_{jrk}$ of p_r and π_{jrk}, respectively, the posterior probability that each individual belongs to each class, conditional on the observed values of the manifest variables, can be calculated using Bayes' formula:

$$\widehat{\mathsf{P}}(r_i|Y_i) = \frac{\hat{p}_r f(Y_i; \hat{\pi}_r)}{\sum_{q=1}^{R} \hat{p}_q f(Y_i; \hat{\pi}_q)}. \tag{3}$$

where $r_i \in \{1, \ldots, R\}$. Recall that the $\hat{\pi}_{jrk}$ are estimates of outcome probabilities *conditional on* class r.

It is important to remain aware that the number of independent parameters estimated by the latent class model increases rapidly with R, J, and K_j. Given these values, the number of parameters is $R \sum_j (K_j - 1) + (R - 1)$. If this number exceeds either the total number of observations, or one fewer than the total number of cells in the cross-classification table of the manifest variables, then the latent class model will be unidentified.

2.2. Parameter estimation

poLCA estimates the latent class model by maximizing the log-likelihood function

$$\ln L = \sum_{i=1}^{N} \ln \sum_{r=1}^{R} p_r \prod_{j=1}^{J} \prod_{k=1}^{K_j} (\pi_{jrk})^{Y_{ijk}} \tag{4}$$

with respect to p_r and π_{jrk}, using the expectation-maximization (EM) algorithm (Dempster, Laird, and Rubin 1977). This log-likelihood function is identical in form to the standard finite mixture model log-likelihood. As with any finite mixture model, the EM algorithm is applicable because each individual's class membership is unknown and may be treated as missing data (McLachlan and Krishnan 1997; McLachlan and Peel 2000).

The EM algorithm proceeds iteratively. Begin with arbitrary initial values of $\hat{p}_r$ and $\hat{\pi}_{jrk}$, and label them $\hat{p}_r^{old}$ and $\hat{\pi}_{jrk}^{old}$. In the expectation step, calculate the "missing" class membership probabilities using Equation 3, substituting in $\hat{p}_r^{old}$ and $\hat{\pi}_{jrk}^{old}$. In the maximization step, update the parameter estimates by maximizing the log-likelihood function given these posterior $\widehat{\mathsf{P}}(r_i|Y_i)$, with

$$\hat{p}_r^{new} = \frac{1}{N} \sum_{i=1}^{N} \widehat{\mathsf{P}}(r_i|Y_i) \tag{5}$$

as the new prior probabilities and

$$\hat{\pi}_{jr}^{new} = \frac{\sum_{i=1}^{N} Y_{ij} \widehat{\mathsf{P}}(r_i|Y_i)}{\sum_{i=1}^{N} \widehat{\mathsf{P}}(r_i|Y_i)} \tag{6}$$

as the new class-conditional outcome probabilities; see Everitt and Hand (1981), and Everitt (1984). In Equation 6, $\hat{\pi}_{jr}^{new}$ is the vector of length K_j of class-r conditional outcome probabilities for the jth manifest variable; and Y_{ij} is the $N \times K_j$ matrix of observed outcomes Y_{ijk} on that variable. The algorithm repeats these steps, assigning the new to the old, until the overall log-likelihood reaches a maximum and ceases to increment beyond some arbitrarily small value.

poLCA takes advantage of the iterative nature of the EM algorithm to make it possible to estimate the latent class model even when some of the observations on the manifest variables are missing. Although **poLCA** does offer the option to listwise delete observations with missing values before estimating the model, it is not necessary to do so. Instead, when determining the product in Equation 1 and the sum in the numerator of Equation 6, **poLCA** simply excludes from the calculation any manifest variables with missing observations. The priors are updated in Equation 3 using as many or as few manifest variables as are observed for each individual.

Depending on the initial values chosen for $\hat{p}_r^{old}$ and $\hat{\pi}_{jrk}^{old}$, and the complexity of the latent class model being estimated, the EM algorithm may only find a local maximum of the log-likelihood function, rather than the desired global maximum. For this reason, it is always advisable to re-estimate a particular model a couple of times when using **poLCA**, in an attempt to find the global maximizer to be taken as the maximum likelihood solution.

2.3. Standard error estimation

poLCA estimates standard errors of the estimated class-conditional response probabilities $\hat{\pi}_{jrk}$ and the mixing parameters $\hat{p}_r$ using the *empirical observed* information matrix (Meilijson 1989), which, following McLachlan and Peel (2000, 66), equals

$$\boldsymbol{I_e}(\hat{\Psi}; Y) = \sum_{i=1}^{N} \boldsymbol{s}(Y_i; \hat{\Psi}) \boldsymbol{s}^T(Y_i; \hat{\Psi}), \tag{7}$$

where $\boldsymbol{s}(Y_i; \hat{\Psi})$ is the score function with respect to the vector of parameters Ψ for the ith observation, evaluated at the maximum likelihood estimate $\hat{\Psi}$;

$$\boldsymbol{s}(Y_i; \Psi) = \sum_{r=1}^{R} \theta_{ir} \partial \{\ln p_r + \sum_{j=1}^{J} \sum_{k=1}^{K_j} Y_{ijk} \ln \pi_{jrk}\} / \partial \Psi \tag{8}$$

where $\theta_{ir} = \widehat{\mathsf{P}}(r_i | Y_i)$ is the posterior probability that observation i belongs to class r (Equation 3). The covariance matrix of the parameter estimates is then approximated by the inverse of $\boldsymbol{I_e}(\hat{\Psi}; Y)$.

Because of the sum-to-one constraint on the π_{jrk} across each manifest variable, it is useful to reparameterize the score function in terms of log-ratios $\phi_{jrk} = \ln(\pi_{jrk}/\pi_{jr1})$ for given outcome variable j and class r. Then, for the lth response on the hth item in the qth class,

$$\boldsymbol{s}(Y_i; \phi_{hql}) = \theta_{iq}(Y_{ihl} - \pi_{hql}). \tag{9}$$

Likewise, denoting $\omega_r = \ln(p_r/p_1)$, then for the log-ratio corresponding to the qth mixing parameter,

$$\boldsymbol{s}(Y_i; \omega_q) = \theta_{iq} - p_q. \tag{10}$$

To transform the covariance matrix of these log-ratios back to the original units of π and p, we apply the delta method. For the response probabilities, let $g(\phi_{jrk}) = \pi_{jrk} = e^{\phi_{jrk}} / \sum_l e^{\phi_{jrl}}$. Taking as $\mathsf{VAR}(\hat{\phi})$ the submatrix of the inverse of $\boldsymbol{I_e}(\hat{\Psi}; Y)$ corresponding to the ϕ parameters, hen

$$\mathsf{VAR}(g(\hat{\phi})) = g'(\phi) \mathsf{VAR}(\hat{\phi}) g'(\phi)^T$$

where $g'(\phi)$ is the Jacobian consisting of elements

$$\frac{\partial g(\phi_{jrk})}{\partial \phi_{hql}} = \begin{cases} 0 & \text{if } q \neq r \\ 0 & \text{if } q = r \text{ but } h \neq j \\ -\pi_{jrk}\pi_{jrl} & \text{if } q = r \text{ and } h = j \text{ but } l \neq k \\ \pi_{jrk}(1 - \pi_{jrk}) & \text{if } q = r \text{ and } h = j \text{ and } l = k. \end{cases}$$

For the mixing parameters, similarly let $h(\omega_r) = p_r = e^{p_r} / \sum_q e^{p_q}$. Taking as $\mathsf{VAR}(\hat{\omega})$ the submatrix of the inverse of $\boldsymbol{I_e}(\hat{\Psi}; Y)$ corresponding to the ω parameters, then

$$\mathsf{VAR}(h(\hat{\omega})) = h'(\omega)\mathsf{VAR}(\hat{\omega})h'(\omega)^T$$

where $h'(\omega)$ is the Jacobian consisting of elements

$$\frac{\partial h(\omega_r)}{\partial \omega_q} = \begin{cases} -p_r p_q & \text{if } q \neq r \\ p_r(1 - p_r) & \text{if } q = r. \end{cases}$$

Standard errors of each parameter estimate are equal to the square root of the values along the main diagonal of covariance matrices $\mathsf{VAR}(\pi)$ and $\mathsf{VAR}(p)$.

2.4. Model selection and goodness of fit criteria

One of the benefits of latent class analysis, in contrast to other statistical techniques for clustered data, is the variety of tools available for assessing model fit and determining an appropriate number of latent classes R for a given data set. In some applications, the number of latent classes will be selected for primarily theoretical reasons. In other cases, however, the analysis may be of a more exploratory nature, with the objective being to locate the best fitting or most parsimonious model. The researcher may then begin by fitting a complete "independence" model with $R = 1$, and then iteratively increasing the number of latent classes by one until a suitable fit has been achieved.

Adding an additional class to a latent class model will increase the fit of the model, but at the risk of fitting to noise, and at the expense of estimating a further $1 + \sum_j (K_j - 1)$ model parameters. Parsimony criteria seek to strike a balance between over- and under-fitting the model to the data by penalizing the log-likelihood by a function of the number of parameters being estimated. The two most widely used parsimony measures are the Bayesian information criterion, or BIC (Schwartz 1978) and Akaike information criterion, or AIC (Akaike 1973). Preferred models are those that minimize values of the BIC and/or AIC. Let Λ represent the maximum log-likelihood of the model and Φ represent the total number of estimated parameters. Then,

$$\text{AIC} = -2\Lambda + 2\Phi$$

and

$$\text{BIC} = -2\Lambda + \Phi \ln N.$$

poLCA calculates these parameters automatically when estimating the latent class model. The BIC will usually be more appropriate for basic latent class models because of their relative simplicity (Lin and Dayton 1997; Forster 2000).

Calculating Pearson's χ^2 goodness of fit and likelihood ratio chi-square (G^2) statistics for the observed versus predicted cell counts is another method to help determine how well a particular model fits the data (Goodman 1970). Let q_c denote the observed number of cases in the cth cell of the cross-classification table of the manifest variables, for cells $c = 1 \ldots C$, where $C = \prod K_j$. The expected percentage of the population in each cell of the fitted J-dimensional table is calculated by inserting estimates $\hat{p}_r$ and $\hat{\pi}_{jrk}$ into Equation 2. Denote as y_c the sequence of J outcomes corresponding to the cth cell in the fitted contingency table, such that $y_{cjk} = 1$ if cell c contains the kth response on the jth variable, and $y_{cjk} = 0$ otherwise. Then, the estimated probability mass function produced by the latent class model is

$$\tilde{P}(y_c) = \sum_{r=1}^{R} \hat{p}_r \prod_{j=1}^{J} \prod_{k=1}^{K_j} (\hat{\pi}_{jrk})^{y_{cjk}}. \tag{11}$$

The expected number of cases in each cell under a given model is $\tilde{Q}_c = N\tilde{P}(y_c)$. The two test statistics are

$$\chi^2 = \sum_{c=1}^{C} (q_c - \tilde{Q}_c)^2 / \tilde{Q}_c$$

and

$$G^2 = 2 \sum_{c=1}^{C} q_c \log(q_c / \tilde{Q}_c).$$

Like the AIC and BIC, these statistics are outputted automatically after calling `poLCA`.

Generally, the goal is to select models that minimize χ^2 or G^2 without estimating excessive numbers of parameters. Note, however, that the distributional assumptions for these statistics are not met if many cells of the observed cross-classification table contain very few observations. Common practice holds that no fewer than 10% to 20% of the cells should contain fewer than five observations if either chi-square test is to be used.

3. Latent class regression models

The latent class regression model generalizes the basic latent class model by permitting the inclusion of covariates to predict individuals' latent class membership (Dayton and Macready 1988; Hagenaars and McCutcheon 2002). This is a so-called "one-step" technique for estimating the effects of covariates, because the coefficients on the covariates are estimated simultaneously as part of the latent class model. An alternative estimation procedure that is sometimes used is called the "three-step" approach: estimate the basic latent class model, calculate the predicted posterior class membership probabilities using Equation 3, and then use these values as the dependent variable(s) in a regression model with the desired covariates. However, as demonstrated by Bolck, Croon, and Hagenaars (2004), the three-step procedure produces biased coefficient estimates. It is preferable to estimate the entire latent class regression model all at once.

Covariates are included in the latent class regression model through their effects on the priors p_r. In the basic latent class model, it is assumed that every individual has the same prior probabilities of latent class membership. The latent class regression model, in contrast, allows individuals' priors to vary depending upon their observed covariates.

3.1. Terminology and model definition

Denote the mixing proportions in the latent class regression model as p_{ri} to reflect the fact that these priors are now free to vary by individual. It is still the case that $\sum_r p_{ri} = 1$ for each individual. To accommodate this constraint, **poLCA** employs a generalized (multinomial) logit link function for the effects of the covariates on the priors (Agresti 2002).

Let X_i represent the observed covariates for individual i. **poLCA** arbitrarily selects the first latent class as a "reference" class and assumes that the log-odds of the latent class membership priors with respect to that class are linear functions of the covariates. Let $\boldsymbol{\beta}_r$ denote the vector of coefficients corresponding to the rth latent class. With S covariates, the $\boldsymbol{\beta}_r$ have length $S+1$; this is one coefficient on each of the covariates plus a constant. Because the first class is used as the reference, $\boldsymbol{\beta}_1 = 0$ is fixed by definition. Then,

$$\begin{aligned} \ln(p_{2i}/p_{1i}) &= X_i\boldsymbol{\beta}_2 \\ \ln(p_{3i}/p_{1i}) &= X_i\boldsymbol{\beta}_3 \\ &\vdots \\ \ln(p_{Ri}/p_{1i}) &= X_i\boldsymbol{\beta}_R \end{aligned}$$

Following some simple algebra, this produces the general result that

$$p_{ri} = p_r(X_i;\boldsymbol{\beta}) = \frac{e^{X_i\boldsymbol{\beta}_r}}{\sum_{q=1}^{R} e^{X_i\boldsymbol{\beta}_q}}. \tag{12}$$

The parameters estimated by the latent class regression model are the $R-1$ vectors of coefficients $\boldsymbol{\beta}_r$ and, as in the basic latent class model, the class-conditional outcome probabilities π_{jrk}. Given estimates $\hat{\boldsymbol{\beta}}_r$ and $\hat{\pi}_{jrk}$ of these parameters, the posterior class membership probabilities in the latent class regression model are obtained by replacing the p_r in Equation 3 with the function $p_r(X_i;\boldsymbol{\beta})$ in Equation 12:

$$\widehat{\mathsf{P}}(r_i|X_i;Y_i) = \frac{p_r(X_i;\hat{\boldsymbol{\beta}})f(Y_i;\hat{\pi}_r)}{\sum_{q=1}^{R} p_q(X_i;\hat{\boldsymbol{\beta}})f(Y_i;\hat{\pi}_q)}. \tag{13}$$

The number of parameters estimated by the latent class regression model is equal to $R\sum_j(K_j - 1) + (S+1)(R-1)$. The same considerations mentioned earlier regarding model identifiability also apply here.

3.2. Parameter estimation

The latent class regression model log-likelihood function is identical to Equation 4 except that the function $p_r(X_i;\boldsymbol{\beta})$ (Equation 12) takes the place of p_r:

$$\ln L = \sum_{i=1}^{N} \ln \sum_{r=1}^{R} p_r(X_i;\boldsymbol{\beta}) \prod_{j=1}^{J}\prod_{k=1}^{K_j} (\pi_{jrk})^{Y_{ijk}}. \tag{14}$$

To find the values of $\hat{\boldsymbol{\beta}}_r$ and $\hat{\pi}_{jrk}$ that maximize this function, **poLCA** uses a modified EM algorithm with a Newton-Raphson step, as set forth by Bandeen-Roche, Miglioretti, Zeger, and Rathouz (1997). This estimation process begins with initial values of $\hat{\boldsymbol{\beta}}_r^{old}$ and $\hat{\pi}_{jrk}^{old}$ that are used to calculate posterior probabilities $\widehat{\mathsf{P}}(r_i|X_i;Y_i)$ (Equation 13). The coefficients on the concomitant variables are updated according to the formula

$$\hat{\boldsymbol{\beta}}_r^{new} = \hat{\boldsymbol{\beta}}_r^{old} - \mathbf{H}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\nabla}_{\boldsymbol{\beta}} \tag{15}$$

where $\boldsymbol{\nabla}_{\boldsymbol{\beta}}$ is the gradient and $\mathbf{H}_{\boldsymbol{\beta}}$ the Hessian matrix of the log-likelihood function with respect to $\boldsymbol{\beta}$. The $\hat{\pi}_{jrk}^{new}$ are updated as

$$\hat{\pi}_{jr}^{new} = \frac{\sum_{i=1}^{N} Y_{ij}\widehat{\mathsf{P}}(r_i|X_i;Y_i)}{\sum_{i=1}^{N}\widehat{\mathsf{P}}(r_i|X_i;Y_i)}. \tag{16}$$

These steps are repeated until convergence, assigning the new parameter estimates to the old in each iteration. The formulas for the gradient and Hessian matrix are provided in Bandeen-Roche *et al.* (1997).

Because all of the concomitant variables must be observed in order to calculate p_{ri} (Equation 12), **poLCA** listwise deletes cases with missing values on the X_i before estimating the latent class regression model. However, missing values on the manifest variables Y_i can be accommodated in the latent class regression model, just as they were in the basic latent class model.

Note that when employing this estimation algorithm, different initial parameter values may lead to different local maxima of the log-likelihood function. To be more certain that the global maximum likelihood solution has been found, the `poLCA` function call should always be repeated a handful of times.

3.3. Standard error estimation

For latent class models with covariates, standard errors are obtained just as for models without covariates: using the empirical observed information matrix (Equation 7). First, we generalize the score function (Equation 8) so that

$$\boldsymbol{s}(X_i,Y_i;\Psi) = \sum_{r=1}^{R}\theta_{ir}\partial\{\ln p_r(X_i;\boldsymbol{\beta}) + \sum_{j=1}^{J}\sum_{k=1}^{K_j} Y_{ijk}\ln\pi_{jrk}\}/\partial\Psi. \tag{17}$$

As before, θ_{ir} denote posterior probabilities. Since this function is no different than Equation 8 in terms of the π parameters, the score function $\boldsymbol{s}(X_i,Y_i;\phi_{hql}) = \boldsymbol{s}(Y_i;\phi_{hql})$ (Equation 9), and the covariance matrix $\mathsf{VAR}(\pi)$ may be calculated in precisely the same way as for models without covariates.

Now, however, the priors p_{ri} are free to vary by individual as a function of some set of coefficients $\boldsymbol{\beta}$, as given in Equation 12. Letting q index classes and s index covariates,

$$s(X_i,Y_i;\beta_{qs}) = X_{is}(\theta_{iq} - p_{iq}). \tag{18}$$

The standard errors of the coefficients β are equal to the square root of the values along the main diagonal of the submatrix of the inverse of the empirical observed information matrix corresponding to the β parameters. (Note that when the model has no covariates, $X_i = 1$ and

$p_{iq} = p_q$ (that is, the priors do not vary by individual), so Equation 18 reduces to Equation 10 as expected.)

To obtain the covariance matrix of the mixing parameters p_r, which are the average value across all observations of the priors p_{ir}, we apply the delta method. Let

$$h(\beta_r) = p_r = \frac{1}{N}\sum_i \left(\frac{e^{X_i \boldsymbol{\beta_r}}}{\sum_{q=1}^{R} e^{X_i \boldsymbol{\beta_q}}} \right).$$

Then

$$\mathsf{VAR}(h(\hat{\beta})) = h'(\beta)\mathsf{VAR}(\hat{\beta})h'(\beta)^T$$

where $h'(\beta)$ is a Jacobian with elements

$$\frac{\partial h(\beta_r)}{\partial \beta_{qs}} = \begin{cases} \frac{1}{N}\sum_i X_{is}(-p_{ir}p_{iq}) & \text{if } q \neq r \\ \frac{1}{N}\sum_i X_{is}(p_{ir}(1-p_{ir})) & \text{if } q = r. \end{cases}$$

When estimating latent class models with covariates, `poLCA` will compute and automatically display t and p-values for the coefficient estimates, corresponding to a null hypothesis of $\beta_{qs} = 0$.

4. Using poLCA

The **poLCA** package makes it possible to estimate a wide range of latent class models in R using a single command line, `poLCA`. Also included in the package is the command `poLCA.simdata`, which enables the user to create simulated data sets that match the data-generating process assumed by either the basic latent class model or the latent class regression model. This functionality is useful for testing the **poLCA** estimator and for performing Monte Carlo-style analyses of latent class models.

4.1. Data input and sample data sets

Data are input to the `poLCA` function as a data frame containing all manifest and concomitant variables (if needed). *The manifest variables must be coded as integer values starting at one for the first outcome category, and increasing to the maximum number of outcomes for each variable.* If any of the manifest variables contain zeros, negative values, or decimals, `poLCA` will produce an error message and terminate without estimating the model. The input data frame may contain missing values.

poLCA also comes pre-installed with five sample data sets that are useful for exploring different aspects of latent class and latent class regression models.

`carcinoma`: Dichotomous ratings by seven pathologists of 118 slides for the presence or absence of carcinoma in the uterine cervix. Source: Agresti (2002, 542).

`cheating`: Dichotomous responses by 319 undergraduates to questions about cheating behavior. Also each student's GPA, which is useful as a concomitant variable. Source: Dayton (1998, 33 and 85).

`election`: Two sets of six questions with four responses each, asking respondents' opinions of how well various traits describe presidential candidates Al Gore and George W. Bush. Also potential covariates vote choice, age, education, gender, and party ID. Source: The National Election Studies (2000).

`gss82`: Attitudes towards survey taking across two dichotomous and two trichotomous items among 1202 white respondents to the 1982 General Social Survey. Source: McCutcheon (1987, 30).

`values`: Dichotomous measures of 216 survey respondents' tendencies towards "universalistic" or "particularistic" values on four questions. Source: Goodman (1974).

These data sets may be accessed using the command `data("carcinoma")`, for example. Examples of models and analyses using the sample data sets are included in the internal documentation for each.

4.2. poLCA command line options

To specify a latent class model, **poLCA** uses the standard, symbolic R model formula expression. The response variables are the manifest variables of the model. Because latent class models have multiple manifest variables, these variables must be "bound" as `cbind(Y1, Y2, Y3, ...)` in the model formula. For the basic latent class model with no covariates, the formula definition takes the form

```
R> f <- cbind(Y1, Y2, Y3) ~ 1
```

The `~ 1` instructs `poLCA` to estimate the basic latent class model. For the latent class regression model, replace the `~ 1` with the desired function of covariates, as, for example:

```
R> f <- cbind(Y1, Y2, Y3) ~ X1 + X2 * X3
```

Further assistance on formula specification in R can be obtained by entering `?formula` at the command prompt.

To estimate the specified latent class model, the default `poLCA` command is:

```
R> poLCA(formula, data, nclass = 2, maxiter = 1000, graphs = FALSE,
+    tol = 1e-10, na.rm = TRUE, probs.start = NULL, nrep = 1,
+    verbose = TRUE, calc.se = TRUE)
```

At minimum, it is necessary to enter a formula (as just described) and a data frame (as described in the previous subsection). The remaining options are:

`nclass`: The number of latent classes to assume in the model; R in the above notation. Setting `nclass = 1` results in `poLCA` estimating the loglinear independence model (Goodman 1970). The default is two.

`maxiter`: The maximum number of iterations through which the estimation algorithm will cycle. If convergence is not achieved before reaching this number of iterations, `poLCA` terminates and reports an error message. The default is 1000, but this will be insufficient for certain models.

`graphs`: Logical, for whether `poLCA` should graphically display the parameter estimates at the completion of the estimation algorithm. The default is `FALSE`.

`tol`: A tolerance value for judging when convergence has been reached. When the one-iteration change in the estimated log-likelihood is less than `tol`, the estimation algorithm stops updating and considers the maximum log-likelihood to have been found. The default is 1×10^{-10} which is a standard value; this option will rarely need to be invoked.

`na.rm`: Logical, for how `poLCA` handles cases with missing values on the manifest variables. If `TRUE`, those cases are removed (listwise deleted) before estimating the model. If `FALSE`, cases with missing values are retained. (As discussed above, cases with missing covariates are always removed.) The default is `TRUE`.

`probs.start`: A list of matrices of class-conditional response probabilities, π_{jrk}, to be used as the starting values for the EM estimation algorithm. Each matrix in the list corresponds to one manifest variable, with one row for each latent class, and one column for each possible outcome. The default is `NULL`, meaning that starting values are generated randomly. Note that if `nrep` > 1, then any user-specified `probs.start` values are only used in the first of the `nrep` attempts.

`nrep`: Number of times to estimate the model, using different values of `probs.start`. The default is one. Setting `nrep` > 1 automates the search for the global—rather than just a local—maximum of the log-likelihood function. `poLCA` returns only the parameter estimates corresponding to the model producing the greatest log-likelihood.

`verbose`: Logical, indicating whether `poLCA` should output to the screen the results of the model. If `FALSE`, no output is produced. The default is `TRUE`.

`calc.se`: Logical, indicating whether `poLCA` should calculate the standard errors of the estimated class-conditional response probabilities and mixing proportions. The default is `TRUE`; can only be set to `FALSE` if estimating a basic model with no concomitant variables specified in `formula`.

4.3. poLCA output

The `poLCA` function returns an object containing the following elements:

`y`: A data frame of the manifest variables.

`x`: A data frame of the covariates, if specified.

`N`: Number of cases used in the model.

`Nobs`: Number of fully observed cases (less than or equal to `N`).

`probs`: A list of matrices containing the estimated class-conditional outcome probabilities $\hat{\pi}_{jrk}$. Each item in the list represents one manifest variable; columns correspond to possible outcomes on each variable, and rows correspond to the latent classes.

`probs.se`: Standard errors of the estimated class-conditional response probabilities, in the same format as `probs`.

P: The respective size of each latent class; equal to the estimated mixing proportions $\hat{p}_r$ in the basic latent class model, or the mean of the priors in the latent class regression model.

P.se: The standard errors of P.

posterior: An $N \times R$ matrix containing each observation's posterior class membership probabilities. Also see the function poLCA.posterior.

predclass: A vector of length N of predicted class memberships, by modal assignment.

predcell: A table of observed versus predicted cell counts for cases with no missing values. Also see functions poLCA.table and poLCA.predcell.

llik: The maximum value of the estimated model log-likelihood.

numiter: The number of iterations required by the estimation algorithm to achieve convergence.

maxiter: The maximum number of iterations through which the estimation algorithm was set to run.

coeff: An $(S+1) \times (R-1)$ matrix of estimated multinomial logit coefficients $\hat{\boldsymbol{\beta}}_r$, for the latent class regression model. Rows correspond to concomitant variables X. Columns correspond to the second through Rth latent classes; see Equation 12.

coeff.se: Standard errors of the coefficient estimates, in the same format as coeff.

coeff.V: Covariance matrix of the coefficient estimates.

aic: Akaike Information Criterion.

bic: Bayesian Information Criterion.

Gsq: Likelihood ratio/deviance statistic.

Chisq: Pearson Chi-square goodness of fit statistic.

time: Length of time it took to estimate the model.

npar: The number of degrees of freedom used by the model (that is, the number of estimated parameters).

resid.df: The number of residual degrees of freedom, equal to the lesser of N and $(\prod_j K_j) - 1$, minus npar.

attempts: A vector containing the maximum log-likelihood values found in each of the nrep attempts to fit the model.

eflag: Logical, error flag. TRUE if estimation algorithm needed to automatically restart with new initial parameters, otherwise FALSE. A restart is caused in the event of computational/rounding errors that result in nonsensical parameter estimates. If an error occurs, poLCA outputs an error message to alert the user.

`probs.start`: A list of matrices containing the class-conditional response probabilities used as starting values in the EM estimation algorithm. If the algorithm needed to restart (see `eflag`), this contains the starting values used for the final, successful, run of the estimation algorithm.

`probs.start.ok`: Logical. `FALSE` if `probs.start` was incorrectly specified by the user, otherwise `TRUE`.

If `verbose=TRUE`, selected items from this list are displayed automatically once the latent class model has been estimated.

4.4. Predicted cell frequencies from the latent class model

The `poLCA` object contains an element `predcell` which enables quick comparisons of the observed cell counts to the cell counts predicted by the latent class model—but only for cells that were observed to contain at least one observation. To generate predicted cell counts for *any* combination of the manifest variables, including cells with zero observations, apply the `poLCA.table` function to the fitted latent class model stored in the `poLCA` object. This function post-processes the latent class model estimates to produce frequency distributions and two-way tables of predicted cell counts, holding the values of the other manifest variables fixed at a user-specified set of values.

As an example, consider a basic two-class latent class model fitted to the four survey variables in the `gss82` data set included in the **poLCA** package.

```
R> data("gss82")
R> f <- cbind(PURPOSE, ACCURACY, UNDERSTA, COOPERAT) ~ 1
R> gss.lc2 <- poLCA(f, gss82, nclass = 2)
```

Entering `gss.lc2$predcell` shows that of the 36 possible four-response sequences of responses ($3 \times 2 \times 2 \times 3$), only 33 are actually observed. One unobserved sequence is the combination `PURPOSE`=3, `ACCURACY`=1, `UNDERSTA`=2, `COOPERAT`=3. We produce the predicted frequency table for `COOPERAT` conditional on the specified values of the other three variables using the command

```
R> poLCA.table(formula = COOPERAT ~ 1,
+     condition = list(PURPOSE = 3, ACCURACY = 1, UNDERSTA = 2),
+     lc = gss.lc2)
```

Fitted values of 4.94 for `COOPERAT`=1 and 0.76 for `COOPERAT`=2 also appeared in the outputted `gss.lc2$predcell`; we may now also see that the fitted value for `COOPERAT`=3 is 0.16, close to the observed value of zero.

To make a two-way table, modify the specification of the `formula` argument to contain both a row (`COOPERAT`) and a column (`UNDERSTA`) variable, again using the `condition` argument to hold fixed the values of the other two variables:

```
R> poLCA.table(formula = COOPERAT ~ UNDERSTA,
+     condition = list(PURPOSE = 3, ACCURACY = 1),
+     lc = gss.lc2)
```

The second column of this table, corresponding to UNDERSTA=2, is identical to the conditional frequency just shown.

If the quantity of interest is not the predicted cell *counts*, but rather the cell *percentages* $\tilde{P}(y_c)$ estimated by the model for particular combinations of the manifest variables, apply the poLCA.predcell function. The fitted latent class model is a density estimate of the joint distribution of the manifest variables in the population, which may be represented as a large multi-way contingency table (see, e.g., Linzer 2011). The poLCA.predcell function calculates the value of the estimated probability mass function for specified cells in that table. For example, the latent class model density estimate of the percentage of people in the underlying population replying 1 to all four questions is 34%:

```
R> poLCA.predcell(lc = gss.lc2, y = c(1, 1, 1, 1))
```

Multiplying this percentage by the total number of observations in the data set, 1202, produces an expected cell frequency of 408.1, as may also be seen in the first row of gss.lc2$predcell.

4.5. Entropy of a fitted latent class model

Entropy is a measure of dispersion (or concentration) in a probability mass function. For multivariate categorical data, it is calculated $H = -\sum_c p_c \ln(p_c)$ where p_c is the share of the probability in the cth cell of the cross-classification table. A fitted latent class model produces a smoothed density estimate of the underlying distribution of cell percentages in the multi-way table of the manifest variables. The poLCA.entropy function calculates the entropy of that estimated probability mass function, setting $p_c = \tilde{P}(y_c)$ in the above notation.

4.6. Reordering the latent classes

Because the latent classes are unordered categories, the numerical order of the estimated latent classes in the model output is arbitrary, and is determined solely by the start values of the EM algorithm. If probs.start is set to NULL (the default) when calling poLCA, then the function generates the starting values randomly in each run. This means that repeated runs of poLCA will typically produce results containing the same parameter estimates (corresponding to the same maximum log-likelihood), but with reordered latent class labels.

To change the order of the latent classes, it is convenient to use the included function poLCA.reorder. Suppose you have estimated a three-class model and wish to reverse the second and third class labels in the output. After an initial call to poLCA, extract the outputted list of probs.start.

```
R> lc <- poLCA(f, dat, nclass = 3)
R> probs.start <- lc$probs.start
```

The poLCA.reorder function takes as its first argument the list of starting values probs.start, and as its second argument a vector describing the desired reordering of the latent classes. In this example, the vector c(1, 3, 2) instructs poLCA.reorder to keep the first class in its current position, but move the third class to the second, and the second class to the third.

```
R> new.probs.start <- poLCA.reorder(probs.start, c(1, 3, 2))
```

Maximum	Number of	Respondent type		
log-likelihood	occurrences	Ideal	Skeptics	Believers
-2754.545	258	0.621	0.172	0.207
-2755.617	14	0.782	0.150	0.067
-2755.739	57	0.796	0.162	0.043
-2762.005	70	0.508	0.392	0.099
-2762.231	101	0.297	0.533	0.170

Table 1: Results of 500 `poLCA` function calls for three-class model using `gss82` data set. Five local maxima of the log-likelihood function were found. Estimated latent class proportions $\hat{p}_r$ are reported for each respondent type at each local maximum.

Then run `poLCA` once more, this time using the reordered starting values in the function call.

```
R> lc <- poLCA(f, dat, nclass = 3, probs.start = new.probs.start)
```

The outputted class labels will now match the desired ordering.

4.7. Recognizing and avoiding local maxima

A well-known drawback of the EM algorithm is that depending upon the initial parameter values chosen in the first iteration, the algorithm may only find a local, rather than the global, maximum of the log-likelihood function (McLachlan and Krishnan 1997). To avoid these local maxima, a user should *always* either 1) call `poLCA` at least a couple of times; or 2) utilize the `nrep` argument to attempt to locate the parameter values that globally maximize the log-likelihood function.

We demonstrate this using a basic three-class latent class model to analyze the four survey variables in the `gss82` data set included in the **poLCA** package.

```
R> data("gss82")
R> f <- cbind(PURPOSE, ACCURACY, UNDERSTA, COOPERAT) ~ 1
```

We estimate this model 500 times, and after each function call, we record the maximum log-likelihood and the estimated population sizes of the three types of survey respondent. Following McCutcheon (1987), from whom these data were obtained, we label the three types *ideal*, *skeptics*, and *believers*. Among other characteristics, the ideal type is the most likely to have a good understanding of surveys, while the believer type is the least likely.

```
R> mlmat <- NULL
R> for (i in 1:500) {
+    gss.lc <- poLCA(f, gss82, nclass = 3, maxiter = 3000, verbose = FALSE)
+    o <- order(gss.lc$probs$UNDERSTA[, 1], decreasing = TRUE)
+    mlmat <- rbind(mlmat, c(gss.lc$llik, gss.lc$P[o]))
+  }
```

Results of this simulation are reported in Table 1. Of the five local maxima of the log-likelihood function that were found, the global maximum was obtained in only approximately half of the trials. At the global maximum, the ideal type is estimated to represent 62.1% of the population, with another 17.2% skeptics and 20.7% believers. In contrast, the second-most frequent local maximum was also the lowest of the local maxima, and the parameter estimates corresponding to that "solution" are substantially different: 29.7% ideal types, 53.3% skeptics, and 17.0% believers. This is why it is *essential* to run poLCA multiple times until you can be reasonably certain that you have found the parameter estimates that produce the global maximum likelihood solution.

To automate this search using the nrep argument, specify the model as

```
R> gss.lc <- poLCA(f, gss82, nclass = 3, maxiter = 3000, nrep = 10)
```

The latent class model will be estimated ten times using different initial parameter values, and will assign to gss.lc the results corresponding to the model with the greatest value of the log-likelihood function. Sample output will appear as follows.

```
Model 1: llik = -2762.231 ... best llik = -2762.231
Model 2: llik = -2755.739 ... best llik = -2755.739
Model 3: llik = -2754.545 ... best llik = -2754.545
Model 4: llik = -2754.545 ... best llik = -2754.545
Model 5: llik = -2754.545 ... best llik = -2754.545
Model 6: llik = -2762.005 ... best llik = -2754.545
Model 7: llik = -2755.739 ... best llik = -2754.545
Model 8: llik = -2754.545 ... best llik = -2754.545
Model 9: llik = -2754.545 ... best llik = -2754.545
Model 10: llik = -2754.545 ... best llik = -2754.545
```

In this example, the global maximum log-likelihood, -2754.545, is found in the third attempt at fitting the model.

4.8. Creating simulated data sets

The command poLCA.simdata will generate simulated data sets that can be used to examine properties of the latent class and latent class regression model estimators. The properties of the simulated data set are fully customizable, but poLCA.simdata uses the following default arguments in the function call.

```
R> poLCA.simdata(N = 5000, probs = NULL, nclass = 2, ndv = 4, nresp = NULL,
+    x = NULL, niv = 0, b = NULL, P = NULL, missval = FALSE, pctmiss = NULL)
```

These input arguments control the following parameters:

N: Total number of observations, N.

probs: A list of matrices of dimension nclass × nresp, containing, by row, the class-conditional outcome probabilities π_{jrk} (which must sum to 1) for the manifest variables. Each matrix represents one manifest variable. If probs is NULL (default) then the outcome probabilities are generated randomly.

nclass: The number of latent classes, R. If probs is specified, then nclass is set equal to the number of rows in each matrix in that list. If P is specified, then nclass is set equal to the length of that vector. Otherwise, the default is two.

`ndv`: The number of manifest variables, J. If `probs` is specified, then `ndv` is set equal to the number of matrices in that list. If `nresp` is specified, then `ndv` is set equal to the length of that vector. Otherwise, the default is four.

`nresp`: The number of possible outcomes for each manifest variable, K_j, entered as a vector of length `ndv`. If `probs` is specified, then `ndv` is set equal to the number of columns in each matrix in that list. If both `probs` and `nresp` are `NULL` (default), then the manifest variables are assigned a random number of outcomes between two and five.

`x`: A matrix of concomitant variables, of dimension `N` $\times$ `niv`. If `niv` > 0 but `x` is `NULL` (default) then the concomitant variable(s) will be generated randomly. If both `x` and `niv` are entered, then then the number of columns in `x` overrides the value of `niv`.

`niv`: The number of concomitant variables, S. Setting `niv` $= 0$ (default) creates a data set assuming no covariates. If `nclass` $= 1$ then `niv` is automatically set equal to 0. Unless `x` is specified, all covariates consist of random draws from a standard normal distribution and are mutually independent.

`b`: When using covariates, an `niv`+1 × `nclass`−1 matrix of (multinomial) logit coefficients, $\boldsymbol{\beta}_r$. If `b` is `NULL` (default), then coefficients are generated as random integers between -2 and 2.

`P`: A vector of mixing proportions of length `nclass`, corresponding to p_r. `P` must sum to 1. Disregarded if `niv`$>$ 1 because then `P` is, in part, a function of the concomitant variables. If `P` is `NULL` (default), then the p_r are generated randomly.

`missval`: Logical. If `TRUE` then a fraction `pctmiss` of the observations on the manifest variables are randomly dropped as missing values. Default is `FALSE`.

`pctmiss`: The percentage of values to be dropped as missing, if `missval = TRUE`. If `pctmiss` is `NULL` (default), then a value between 5% and 40% is chosen randomly.

Note that in many instances, specifying values for certain arguments will override other specified arguments. Be sure when calling `poLCA.simdata` that all arguments are in logical agreement, or else the function may produce unexpected results.

Specifying the list of matrices `probs` can be tricky; we recommend a command structure such as this for, for example, five manifest variables, three latent classes, and $K_j = (3, 2, 3, 4, 3)$.

```
R> probs <- list(
+    matrix(c(0.6, 0.1, 0.3,      0.6, 0.3, 0.1,      0.3, 0.1, 0.6),
+      ncol = 3, byrow = TRUE),
+    matrix(c(0.2, 0.8,           0.7, 0.3,           0.3, 0.7),
+      ncol = 2, byrow = TRUE),
+    matrix(c(0.3, 0.6, 0.1,      0.1, 0.3, 0.6,      0.3, 0.6, 0.1),
+      ncol = 3, byrow = TRUE),
+    matrix(c(0.1, 0.1, 0.5, 0.3, 0.5, 0.3, 0.1, 0.1, 0.3, 0.1, 0.1, 0.5),
+      ncol = 4, byrow = TRUE),
+    matrix(c(0.1, 0.1, 0.8,      0.1, 0.8, 0.1,      0.8, 0.1, 0.1),
+      ncol = 3, byrow = TRUE))
```

The object returned by `poLCA.simdata` is a list containing both the simulated data set *and* all of the parameters used to generate that data set. The elements listed here have the same characteristics and meanings as just described.

`dat`: A data frame containing the simulated variables X and Y. Variable names for manifest variables are `Y1`, `Y2`, ..., `Y`J. Variable names for concomitant variables are `X1`, `X2`, ..., `X`S.

`probs`: A list of matrices of dimension `nclass` × `nresp` containing the class-conditional outcome probabilities.

`nresp`: A vector containing the number of possible outcomes for each manifest variable.

`b`: A matrix containing the coefficients on the covariates, if used.

`P`: The mixing proportions corresponding to each latent class.

`pctmiss`: The percent of observations missing.

`trueclass`: A vector of length `N` containing the "true" class membership for each individual.

Examples of possible uses of `poLCA.simdata` are included in the poLCA internal documentation and may be accessed by entering `? poLCA.simdata` in R. One example demonstrates that even when the "true" data generating process involves a series of covariates—so that each observation has a different prior probability of belonging to each class—the posterior probabilities of latent class membership can still be recovered with high accuracy using a basic model specified without covariates. A second example confirms that in data sets with missing values, the `poLCA` function produces consistent estimates of the class-conditional response probabilities π_{jrk} regardless of whether the researcher elects to include or listwise delete the observations with missing values.

5. Two examples

To illustrate the usage of the **poLCA** package, we present two examples: a basic latent class model and a latent class regression model, using sample data sets included in the package.

5.1. Basic latent class modeling with the `carcinoma` data

The `carcinoma` data from Agresti (2002, 542) consist of seven dichotomous variables that represent the ratings by seven pathologists of 118 slides on the presence or absence of carcinoma. The purpose of studying these data is to model "interobserver agreement" by examining how subjects might be divided into groups depending upon the consistency of their diagnoses.

It is straightforward to replicate Agresti's published results (Agresti 2002, 543) using the series of commands:

```
R> data("carcinoma")
R> f <- cbind(A, B, C, D, E, F, G) ~ 1
R> lc2 <- poLCA(f, carcinoma, nclass = 2)
R> lc3 <- poLCA(f, carcinoma, nclass = 3, graphs = TRUE)
R> lc4 <- poLCA(f, carcinoma, nclass = 4, maxiter = 5000)
```

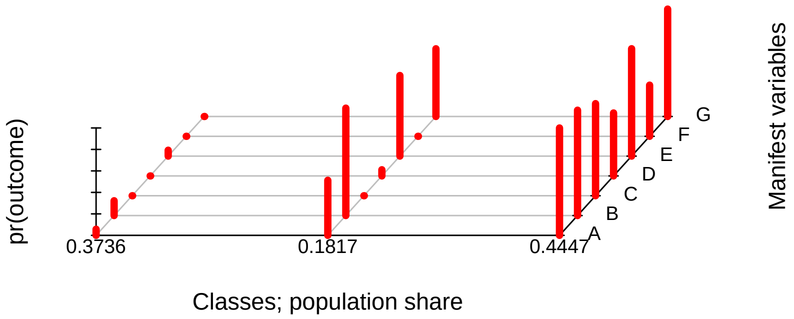


Figure 1: Estimation of the three-class basic latent class model using the `carcinoma` data; obtained by setting `graphs = TRUE` in the `poLCA` function call. Each group of red bars represents the conditional probabilities, by latent class, of being rated positively by each of the seven pathologists (labeled A through G). Taller bars correspond to conditional probabilities closer to 1 of a positive rating.

The four-class model will typically require a larger number of iterations to achieve convergence.

Figure 1 shows a screen capture of the estimation of model `lc3` with the `graphs` option set to `TRUE`. As Agresti describes, the three estimated latent classes clearly correspond to a pair of classes that are consistently rated negative (37%) or positive (44%), plus a third "problematic" class representing 18% of the population. In that class, pathologists B, E, and G tend to diagnose positive; C, D, and F tend to diagnose negative; and A is about 50/50.

The full output from the estimation of model `lc3` is given below. First, the estimated class-conditional response probabilities $\hat{\pi}_{jrk}$ are reported for pathologists A through G, with each row corresponding to a latent class, and each column corresponding to a diagnosis; negative in the first column, and positive in the second. Thus, for example, a slide belonging to the first ("negative") class has a 94% chance of being rated free from carcinoma by rater A, an 86% chance of the same from rater B, an 100% chance from rater C, and so forth.

Next, the output provides the estimated mixing proportions $\hat{p}_r$ corresponding to the share of observations belonging to each latent class. These are the same values that appear in Figure 1. An alternative method for determining the size of the latent classes is to assign each observation to a latent class on an individual basis according to its model posterior class membership probability. Values using this technique are reported directly below the estimated mixing proportions. Congruence between these two sets of population shares often indicates a good fit of the model to the data.

The next set of results simply reports the number of observations, the number of fully observed cases (for data sets with missing values and `na.rm = FALSE`), the number of estimated parameters, residual degrees of freedom, and maximum log-likelihood. It is always worth checking to ensure that the number of residual degrees of freedom is non-negative; `poLCA` will output a warning message if this is the case.

Finally, `poLCA` outputs a number of goodness of fit statistics as described in Section 2.4. For the `carcinoma` data, the minimum AIC and BIC criteria both indicate that the three-class model is most parsimonious: with two classes, the AIC is 664.5 and the BIC is 706.1; with three classes, the AIC decreases to 633.4 and the BIC decreases to 697.1; and with four classes, the AIC increases again to 641.6 and the BIC increases to 727.5.

```
Conditional item response (column) probabilities,
 by outcome variable, for each class (row)

$A
           Pr(1)  Pr(2)
class 1:  0.9427 0.0573
class 2:  0.4872 0.5128
class 3:  0.0000 1.0000

$B
           Pr(1)  Pr(2)
class 1:  0.8621 0.1379
class 2:  0.0000 1.0000
class 3:  0.0191 0.9809

$C
           Pr(1)  Pr(2)
class 1:  1.0000 0.0000
class 2:  1.0000 0.0000
class 3:  0.1425 0.8575

$D
           Pr(1)  Pr(2)
class 1:  1.0000 0.0000
class 2:  0.9424 0.0576
class 3:  0.4138 0.5862

$E
           Pr(1)  Pr(2)
class 1:  0.9449 0.0551
class 2:  0.2494 0.7506
class 3:  0.0000 1.0000

$F
           Pr(1)  Pr(2)
class 1:  1.0000 0.0000
class 2:  1.0000 0.0000
class 3:  0.5236 0.4764

$G
           Pr(1)  Pr(2)
class 1:  1.0000 0.0000
class 2:  0.3693 0.6307
class 3:  0.0000 1.0000

Estimated class population shares
 0.3736 0.1817 0.4447
```

```
Predicted class memberships (by modal posterior prob.)
 0.3729 0.1949 0.4322

=========================================================
Fit for 3 latent classes:
=========================================================
number of observations: 118
number of estimated parameters: 23
residual degrees of freedom: 95
maximum log-likelihood: -293.705

AIC(3): 633.41
BIC(3): 697.1357
G^2(3): 15.26171 (Likelihood ratio/deviance statistic)
X^2(3): 20.50336 (Chi-square goodness of fit)
```

5.2. Latent class regression modeling with the `election` data

In the `election` data set, respondents to the 2000 American National Election Study public opinion poll were asked to evaluate how well a series of traits—moral, caring, knowledgable, good leader, dishonest, and intelligent—described presidential candidates Al Gore and George W. Bush. Each question had four possible choices: (1) extremely well; (2) quite well; (3) not too well; and (4) not well at all.

Models with one covariate

A reasonable theoretical approach might suppose that there are three latent classes of survey respondents: Gore supporters, Bush supporters, and those who are more or less neutral. Gore supporters will tend to respond favorably towards Gore and unfavorably towards Bush, with the reverse being the case for Bush supporters. Those in the neutral group will not have strong opinions about either candidate. We might further expect that falling into one of these three groups is a function of each individual's party identification, with committed Democrats more likely to favor Gore, committed Republicans more likely to favor Bush, and less intense partisans tending to be indifferent. We can investigate this hypothesis using a latent class regression model.

Begin by loading the `election` data into memory, and specifying a model with 12 manifest variables and `PARTY` as the lone concomitant variable. The `PARTY` variable is coded across seven categories, from strong Democrat at 1 to strong Republican at 7. People who primarily consider themselves Independents are at 3-4-5 on the scale. Next, estimate the latent class regression model and assign those results to object `nes.party`. A call to the `poLCA.reorder` command, with a subsequent re-estimation of the model, ensures that the three latent classes are assigned the same category labels in each run.

```
R> data("election")
R> f.party <- cbind(MORALG, CARESG, KNOWG, LEADG, DISHONG, INTELG,
+    MORALB, CARESB, KNOWB, LEADB, DISHONB, INTELB) ~ PARTY
```

```
R> nes.party <- poLCA(f.party, election, nclass = 3, verbose = FALSE)
R> probs.start <- poLCA.reorder(nes.party$probs.start,
+    order(nes.party$P, decreasing = TRUE))
R> nes.party <- poLCA(f.party, election, nclass = 3,
+    probs.start = probs.start)
```

By examining the estimated class-conditional response probabilities, we confirm that the model finds that the three groups indeed separate as expected, with 27% in the favor-Gore group, 34% in the favor-Bush group, and 39% in the neutral group.

This example also illustrates a shortcoming of the χ^2 goodness of fit statistic, which is calculated to be over 34.5 billion. With only 1300 observations but nearly 17 million cells in the observed cross-classification table (that is, four responses to each of 12 questions, or 4^{12} cells), the vast majority of the cells will contain zero cases. For models such as this, using the χ^2 statistic to assess model fit is not advised.

In addition to the information outputted for the basic model, the poLCA output now also includes the estimated coefficients $\hat{\boldsymbol{\beta}}_r$ on the covariates, and their standard errors.

```
=========================================================
Fit for 3 latent classes:
=========================================================
2 / 1
            Coefficient  Std. error  t value  Pr(>|t|)
(Intercept)    -3.81813     0.31109  -12.274         0
PARTY           0.79327     0.06232   12.728         0
=========================================================
3 / 1
            Coefficient  Std. error  t value  Pr(>|t|)
(Intercept)     1.16155     0.17989    6.457         0
PARTY          -0.57436     0.06401   -8.973         0
=========================================================
```

Here, the neutral group is the first latent class, the favor-Bush group is the second latent class, and the favor-Gore group is the third latent class. Following the terminology in Section 3.1, the log-ratio prior probability that a respondent will belong to the favor-Bush group with respect to the neutral group is $\ln(p_{2i}/p_{1i}) = -3.82 + 0.79 \times \texttt{PARTY}$. Likewise, the log-ratio prior probability that a respondent will belong to the favor-Gore group with respect to the neutral group is $\ln(p_{3i}/p_{1i}) = 1.16 - 0.57 \times \texttt{PARTY}$. Equation 12 provides the formula for converting these log-ratios into predicted prior probabilities for each latent class.

To interpret the estimated generalized logit coefficients, we calculate and plot predicted values of p_{ri}, the prior probability of class membership, at varying levels of party ID. The R commands to do this are as follows, producing the graph in Figure 2.

```
R> pidmat <- cbind(1, c(1:7))
R> exb <- exp(pidmat %*% nes.party$coeff)
R> matplot(c(1:7), (cbind(1, exb)/(1 + rowSums(exb))),
+    main = "Party ID as a predictor of candidate affinity class",
```

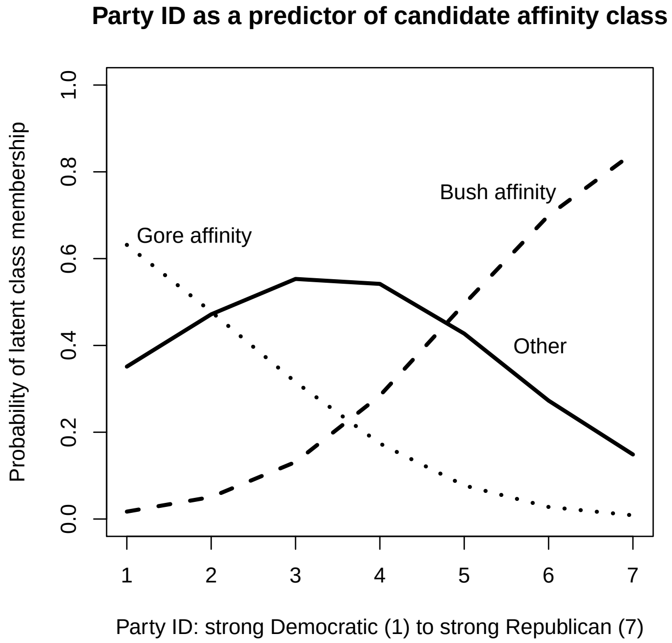


Figure 2: Predicted prior probabilities of latent class membership at varying levels of partisan self-identification. Results are from a three-class latent class regression model.

```
+    xlab = "Party ID: strong Democratic (1) to strong Republican (7)",
+    ylab = "Probability of latent class membership",
+    ylim = c(0, 1), type = "l", lwd = 3, col = 1)
R> text(5.9, 0.40, "Other")
R> text(5.4, 0.75, "Bush affinity")
R> text(1.8, 0.65, "Gore affinity")
```

Strong Democrats have over a 60% prior probability of belonging to the Gore affinity group, while strong Republicans have over an 80% prior probability of belonging to the Bush affinity group. The prior probability of belonging to the indifferent category, labeled "Other", is greatest for self-identified Independents (4) and Independents who lean Democratic (3).

Models with more than one covariate

It is straightforward to similarly investigate models with more than one covariate. Suppose we are interested in whether the effect of age modifies the effect of partisanship on candidate affinity. We specify the interaction model with three covariates:

```
R> f.3cov <- cbind(MORALG, CARESG, KNOWG, LEADG, DISHONG, INTELG,
+    MORALB, CARESB, KNOWB, LEADB, DISHONB, INTELB) ~ PARTY * AGE
R> nes.3cov <- poLCA(f.3cov, election, nclass = 3, verbose = FALSE)
R> probs.start <- poLCA.reorder(nes.3cov$probs.start,
+    order(nes.3cov$P, decreasing = TRUE))
R> nes.3cov <- poLCA(f.3cov, election, nclass = 3,
+    probs.start = probs.start)
```

This produces the following coefficient estimates, again with the neutral group as the first latent class, the favor-Bush group as the second latent class, and the favor-Gore group as the third latent class.

```
=========================================================
Fit for 3 latent classes:
=========================================================
2 / 1
            Coefficient  Std. error  t value  Pr(>|t|)
(Intercept)    -4.39452     0.85423   -5.144     0.000
PARTY           0.80682     0.17614    4.581     0.000
AGE             0.01314     0.01772    0.741     0.459
PARTY:AGE      -0.00020     0.00363   -0.054     0.957
=========================================================
3 / 1
            Coefficient  Std. error  t value  Pr(>|t|)
(Intercept)    -0.31445     0.56324   -0.558     0.577
PARTY          -0.39923     0.19990   -1.997     0.046
AGE             0.02967     0.01121    2.648     0.008
PARTY:AGE      -0.00310     0.00398   -0.778     0.437
=========================================================
```

To see the effects of age on the candidate affinity of strong partisans, we first specify a matrix of hypothetical values of the covariates: `strdems` for Democrats and `strreps` for Republicans. We then calculate and plot the predicted prior probabilities of latent class membership corresponding to each of these chosen hypothetical values (Figure 3).

```
R> strdems <- cbind(1, 1, c(18:80), (c(18:80) * 1))
R> exb.strdems <- exp(strdems %*% nes.3cov$coeff)
R> matplot(c(18:80), (cbind(1, exb.strdems)/(1+rowSums(exb.strdems))),
+    main = "Age and candidate affinity for strong Democrats",
+    xlab = "Age", ylab = "Probability of latent class membership",
+    ylim = c(0, 1), type = "l", col = 1, lwd = 3)
R> strreps <- cbind(1, 7, c(18:80), (c(18:80) * 7))
R> exb.strreps <- exp(strreps %*% nes.3cov$coeff)
R> matplot(c(18:80), (cbind(1, exb.strreps) / (1 + rowSums(exb.strreps))),
+    main = "Age and candidate affinity for strong Republicans",
+    xlab = "Age", ylab = "Probability of latent class membership",
+    ylim = c(0, 1), type = "l", col = 1, lwd = 3)
```

As expected, regardless of age, strong Democrats are very unlikely to belong to the Bush-affinity group, and strong Republicans are very unlikely to belong to the Gore-affinity group. However, it is interesting to observe that while strong Republicans in 2000 had extremely high levels of affinity for Bush at all ages, strong Democrats below the age of 30 tended to be just as (or more) likely to belong to the neutral group as to the Gore-affinity group.

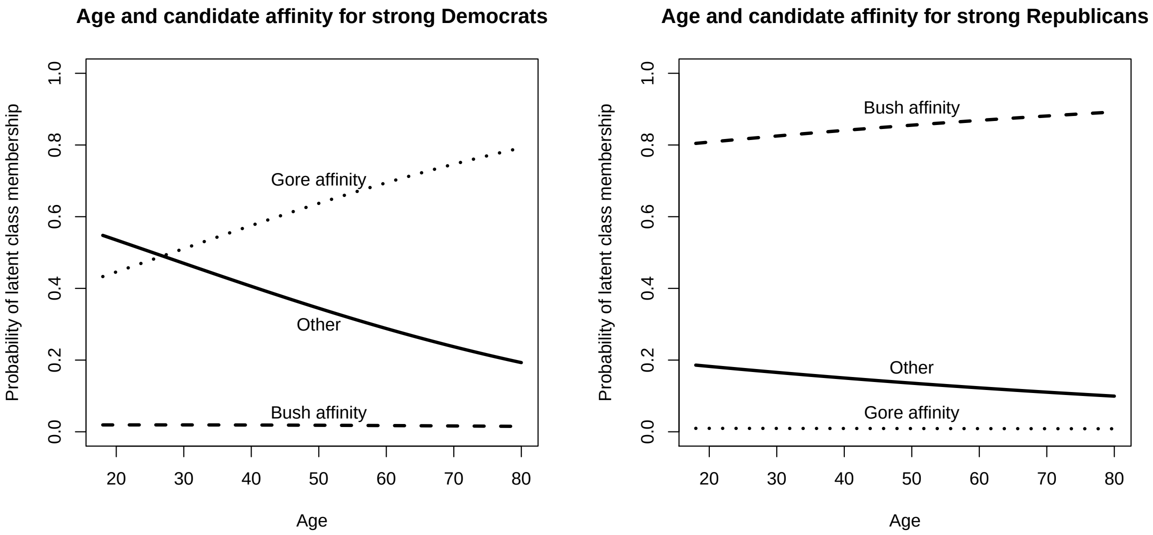


Figure 3: Predicted prior probabilities of latent class membership for strong Democrats (left) and strong Republicans (right) at ages 18-80.

6. Conclusion

The R package **poLCA** provides an easy-to-use framework for the estimation of latent class models and latent class regression models for the analysis of multivariate categorical data. The manifest variables may contain any number of possible (polytomous) outcomes. In the latent class regression model, covariates may be used to predict individual observations' latent class membership. **poLCA** also includes tools for visualizing model results, creating simulated multivariate categorical data sets with an unobserved latent categorical structure, and post-processing the model results to produce various other quantities of interest, including expected cell percentages and posterior probabilities of latent class membership for either observed or hypothetical cases.

poLCA is still undergoing active development. Planned extensions include flexibility to relax the assumption of local independence among selected manifest variables, explicit treatment of manifest variables as ordinal as well as nominal, incorporation of sampling weights, and accommodation of user-specified constraints on the class-conditional response probabilities π_{jrk} as a way to simplify models, achieve model identifiability, test substantive hypotheses, and analyze model fit. Such constraints might, for example, require selected response probabilities to be set equal to one another across different classes, across manifest variables within classes, or equal to fixed constant values, as in Goodman (1974). This extension would also permit the estimation of so-called "simultaneous" latent class models across multiple groups where the groups are already known (or at least theorized) to exist in the data (Clogg and Goodman 1986). The researcher would then include in the model a manifest variable measuring this known categorization, and specify that one of the grouping variable response probabilities be fixed at 1.0 in each class.

References

Agresti A (2002). *Categorical Data Analysis*. John Wiley & Sons, Hoboken.

Akaike H (1973). "Information Theory and an Extension of the Maximum Likelihood Principle." In B Petrov, F Csake (eds.), *Second International Symposium on Information Theory*, pp. 267–281. Akademiai Kiado, Budapest, Hungary.

Bandeen-Roche K, Miglioretti DL, Zeger SL, Rathouz PJ (1997). "Latent Variable Regression for Multiple Discrete Outcomes." *Journal of the American Statistical Association*, **92**(440), 1375–1386.

Beath K (2011). ***randomLCA**: Random Effects Latent Class Analysis*. R package version 0.7-4, URL http://CRAN.R-project.org/package=randomLCA.

Blaydes L, Linzer DA (2008). "The Political Economy of Women's Support for Fundamentalist Islam." *World Politics*, **60**(4), 576–609.

Bolck A, Croon M, Hagenaars J (2004). "Estimating Latent Structure Models with Categorical Variables: One-Step Versus Three-Step Estimators." *Political Analysis*, **12**(1), 3–27.

Breen R (2000). "Why Is Support for Extreme Parties Underestimated by Surveys? A Latent Class Analysis." *British Journal of Political Science*, **30**(2), 375–382.

Clinton J, Jackman S, Rivers D (2004). "The Statistical Analysis of Roll Call Data." *American Political Science Review*, **98**(2), 355–370.

Clogg CC, Goodman LA (1986). "On Scaling Models Applied to Data from Several Groups." *Psychometrika*, **51**(1), 123–135.

Dayton CM (1998). *Latent Class Scaling Analysis*. Sage Publications, Thousand Oaks, CA.

Dayton CM, Macready GB (1988). "Concomitant-Variable Latent-Class Models." *Journal of the American Statistical Association*, **83**(401), 173–178.

Dempster AP, Laird NM, Rubin DB (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society B*, **39**, 1–38.

Dimitriadou E, Hornik K, Leisch F, Meyer D, Weingessel A (2011). ***e1071**: Misc Functions of the Department of Statistics (e1071), TU Wien*. R package version 1.5-25, URL http://CRAN.R-project.org/package=e1071.

Duffy D (2010). ***gllm**: Generalised log-linear model*. R package version 0.33. C code in emgllmfitter by Andreas Borg, URL http://CRAN.R-project.org/package=gllm.

Everitt BS (1984). *An Introduction to Latent Variable Models*. Chapman and Hall, London.

Everitt BS, Hand DJ (1981). *Finite Mixture Distributions*. Chapman and Hall, London.

Feick LF (1989). "Latent Class Analysis of Survey Questions that Include Don't Know Responses." *Public Opinion Quarterly*, **53**(4), 525–547.

Forster MR (2000). "Key Concepts in Model Selection: Performance and Generalizability." *Journal of Mathematical Psychology*, **44**, 205–231.

Goodman L (1970). "The Multivariate Analysis of Qualitative Data: Interactions among Multiple Classifications." *Journal of the American Statistical Association*, **65**, 226–256.

Goodman L (1974). "Exploratory Latent Structure Analysis using both Identifiable and Unidentifiable Models." *Biometrika*, **61**, 315–231.

Grün B, Leisch F (2008). "FlexMix Version 2: Finite Mixtures with Concomitant Variables and Varying and Constant Parameters." *Journal of Statistical Software*, **28**(4), 1–35. URL http://www.jstatsoft.org/v28/i04/.

Hagenaars JA, McCutcheon AL (eds.) (2002). *Applied Latent Class Analysis*. Cambridge University Press, Cambridge.

Hennig C (2010). ***fpc**: Flexible Procedures for Clustering*. R package version 2.0-3, URL http://CRAN.R-project.org/package=fpc.

Hill JL, Kriesi H (2001a). "Classification by Opinion-Changing Behavior: A Mixture Model Approach." *Political Analysis*, **9**(4), 301–324.

Hill JL, Kriesi H (2001b). "An Extension and Test of Converse's 'Black-and-White' Model of Response Stability." *American Political Science Review*, **95**(2), 397–413.

Lazarsfeld PF (1950). "The Logical and Mathematical Foundations of Latent Structure Analysis." In SA Stouffer (ed.), *Measurement and Prediction*, pp. 362–412. John Wiley & Sons, New York.

Leisch F (2004). "FlexMix: A General Framework for Finite Mixture Models and Latent Class Regression in R." *Journal of Statistical Software*, **11**(8), 1–18. URL http://www.jstatsoft.org/v11/i08/.

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.

Linzer DA (2011). "Reliable Inference in Highly Stratified Contingency Tables: Using Latent Class Models as Density Estimators." *Political Analysis*, **19**(2).

Linzer DA, Lewis J (2011). ***poLCA**: Polytomous Variable Latent Class Analysis*. R package version 1.3, URL http://CRAN.R-project.org/package=poLCA.

McCutcheon AL (1985). "A Latent Class Analysis of Tolerance for Nonconformity in the American Public." *The Public Opinion Quarterly*, **49**(5), 474–488.

McCutcheon AL (1987). *Latent Class Analysis*. Sage Publications, Newbury Park.

McLachlan G, Peel D (2000). *Finite Mixture Models*. John Wiley & Sons, New York.

McLachlan GJ, Krishnan T (1997). *The EM Algorithm and Extensions*. John Wiley & Sons, New York.

Meilijson I (1989). "A Fast Improvement to the EM Algorithm on its Own Terms." *Journal of the Royal Statistical Society B*, **51**(1), 127–138.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Schwartz G (1978). "Estimating the Dimension of a Model." *The Annals of Statistics*, **6**, 461–464.

The National Election Studies (2000). *The 2000 National Election Study dataset*. University of Michigan, Center for Political Studies, producer and distributor, Ann Arbor, MI. URL http://www.electionstudies.org/.

Affiliation:

Drew A. Linzer
Department of Political Science
Emory University
327 Tarbutton Hall
1555 Dickey Drive
Atlanta, GA 30322, United States of America
E-mail: dlinzer@emory.edu
URL: http://userwww.service.emory.edu/~dlinzer/

Jeffrey B. Lewis
Department of Political Science
University of California, Los Angeles
Box 951472
4289 Bunche Hall
Los Angeles, CA 90095, United States of America
E-mail: jblewis@polisci.ucla.edu
URL: http://www.sscnet.ucla.edu/polisci/faculty/lewis/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 42, Issue 10 *Submitted:* 2007-02-15
June 2011 *Accepted:* 2007-08-02

Latent Class Analysis

Jeroen K. Vermunt & Jay Magidson

The basic idea underlying latent class (LC) analysis is a very simple one: some of the parameters of a postulated statistical model differ across unobserved subgroups. These subgroups form the categories of a categorical latent variable (see entry LATENT VARIABLE). This basic idea has several seemingly unrelated applications, the most important of which are clustering, scaling, density estimation, and random-effects modeling. Outside social sciences, LC models are often referred to as finite mixture models.

LC analysis was introduced in 1950 by Lazarsfeld, who used the technique as a tool for building typologies (or clustering) based on dichotomous observed variables. More than 20 years later, Goodman (1974) made the model applicable in practice by developing an algorithm for obtaining maximum likelihood estimates of the model parameters. He also proposed extensions for polytomous manifest variables and multiple latent variables, and did important work on the issue of model identification. During the same period, Haberman (1979) showed the connection between LC models and log-linear models for frequency tables with missing (unknown) cell counts. Many important extensions of the classical LC model have been proposed since then, such as models containing (continuous) covariates, local dependencies, ordinal variables, several latent variables, and repeated measures. A general framework for categorical data analysis with discrete latent variables was proposed by Hagenaars (1990) and extended by Vermunt (1997).

While in the social sciences LC and finite mixture models are conceived primarily as tools for categorical data analysis, they can be useful in several other areas as well. One of these is density estimation, in which one makes use of the fact that a complicated density can be approximated as a finite mixture of simpler densities. LC analysis can also be used as a probabilistic cluster analysis tool for continuous observed variables, an approach that offers many advantages over traditional cluster techniques such as K-means clustering (see LATENT PROFILE MODEL). Another application area is dealing with unobserved heterogeneity, for example, in regression analysis with dependent observations (see NON-PARAMETRIC RANDOM-EFFECTS MODEL).

The classical LC model for categorical indicators

Let X represent the latent variable and Y_ℓ one of the L observed or manifest variables, where $1 \leq \ell \leq L$. Moreover, let C be the number of latent classes and D_ℓ the number of levels of Y_ℓ. A particular LC is enumerated by the index x, $x = 1, 2, ..., C$, and a particular value of Y_ℓ by y_ℓ, $y_\ell = 1, 2, ..., D_\ell$. The vector notation $\mathbf{Y}$ and $\mathbf{y}$ is used to refer to a complete response pattern.

In order to make things more concrete, consider the following small data set obtained from the 1987 General Social Survey:

Y_1	Y_2	Y_3	Frequency	$P(X = 1\|\mathbf{Y} = \mathbf{y})$	$P(X = 2\|\mathbf{Y} = \mathbf{y})$
1	1	1	696	.998	.002
1	1	2	68	.929	.071
1	2	1	275	.876	.124
1	2	2	130	.168	.832
2	1	1	34	.848	.152
2	1	2	19	.138	.862
2	2	1	125	.080	.920
2	2	2	366	.002	.998

The three dichotomous indicators Y_1, Y_2, and Y_3 are the responses to the statements "allow anti-religionists to speak"(1 = allowed, 2 = not allowed), "allow anti-religionists to teach" (1 = allowed, 2 = not allowed), "remove anti-religious books from the library" (1 = do not remove, 2 = remove). By means of LC analysis it is possible to identify subgroups with different degrees of tolerance towards anti-religionists.

The basic idea underlying any type of LC model is that the probability of obtaining response pattern $\mathbf{y}$, $P(\mathbf{Y} = \mathbf{y})$, is a weighted average of the C class-specific probabilities $P(\mathbf{Y} = \mathbf{y}|X = x)$; that is,

$$P(\mathbf{Y} = \mathbf{y}) = \sum_{x=1}^{C} P(X = x) P(\mathbf{Y} = \mathbf{y}|X = x). \tag{1}$$

Here, $P(X = x)$ denotes the proportion of persons belonging to LC x.

In the classical LC model, this basic idea is combined with the assumption of LOCAL INDEPENDENCE. The L manifest variables are assumed to be mutually independent within each LC, which can be formulated as follows:

$$P(\mathbf{Y}=\mathbf{y}|X=x)=\prod_{\ell=1}^{L}P(Y_\ell=y_\ell|X=x). \tag{2}$$

After estimating the conditional response probabilities $P(Y_\ell = y_\ell|X = x)$, comparing these probabilities between classes shows how the classes differ from each other, which can be used to name the classes. Combining the two basic equations (1) and (2) yields the following model for $P(\mathbf{Y}=\mathbf{y})$:

$$P(\mathbf{Y}=\mathbf{y})=\sum_{x=1}^{C}P(X=x)\prod_{\ell=1}^{L}P(Y_\ell=y_\ell|X=x).$$

A two-class model estimated with the small example data set yielded the following results:

	$X=1$ (Tolerant)	$X=2$ (Intolerant)
$P(X=x)$	.62	.38
$P(Y_1=1\|X=x)$	.96	.23
$P(Y_2=1\|X=x)$	.74	.04
$P(Y_3=1\|X=x)$	.92	.24

The two classes contain 62 and 38 percent of the individuals, respectively. The first class can be named “Tolerant” because people belonging to that class have much higher probabilities of selecting the tolerant responses on the indicators than people belonging to the second “Intolerant” class.

Similarly to cluster analysis, one of the purposes of LC analysis might be to assign individuals to latent classes. The probability of belonging to LC x – often referred to as posterior membership probability – can be obtained by the Bayes rule,

$$P(X=x|\mathbf{Y}=\mathbf{y})=\frac{P(X=x)P(\mathbf{Y}=\mathbf{y}|X=x)}{P(\mathbf{Y}=\mathbf{y})}. \tag{3}$$

The most common classification rule is modal assignment, which amounts to assigning each individual to the LC with the highest $P(X=x|\mathbf{Y}=\mathbf{y})$. The class-membership probabilities reported in the first table show that people with at least two tolerant responses are classified into the “Tolerant” class.

Log-linear formulation of the LC model

Haberman (1979) showed that the LC model can also be specified as a LOG-LINEAR MODEL for a table with missing cell entries or, more precisely, as a model for the expanded table including the latent variable X as an additional dimension. The relevant log-linear model for $P(X = x, \mathbf{Y} = \mathbf{y})$ has the following form:

$$\ln P(X = x, \mathbf{Y} = \mathbf{y}) = \beta + \beta_x^X + \sum_{\ell=1}^{L} \beta_{y_\ell}^{Y_\ell} + \sum_{\ell=1}^{L} \beta_{x,y_\ell}^{X,Y_\ell}.$$

It contains a main effect, the one-variable terms for the latent variable and the indicators, and the two-variable terms involving X and each of the indicators. Note that the terms involving two or more manifest variables are omitted because of the local independence assumption.

The connection between the log-linear parameters and the conditional response probabilities is as follows:

$$P(Y_\ell = y_\ell | X = x) = \frac{\exp(\beta_{y_\ell}^{Y_\ell} + \beta_{x,y_\ell}^{X,Y_\ell})}{\sum_{r=1}^{D_\ell} \exp(\beta_r^{Y_\ell} + \beta_{x,r}^{X,Y_\ell})}.$$

This shows that the log-linear formulation amounts to specifying a logit model for each of the conditional response probabilities.

The type of LC formulation that is used becomes important if one wishes to impose restrictions. Although constraints on probabilities can sometimes be transformed into constraints on log-linear parameters and vice versa, there are many situations in which this is not possible.

Maximum likelihood estimation

Let I denote the total number of cells entries (or possible answer patterns) in the L-way frequency table, so that $I = \prod_{\ell=1}^{L} D_\ell$, and let i denote a particular cell entry, n_i the observed frequency in cell i, and $P(\mathbf{Y} = \mathbf{y}_i)$ the probability of having the response pattern of cell i.

The parameters of LC models are typically estimated by means of maximum likelihood (ML). The kernel of the log-likelihood function that is maximized equals

$$\ln \mathcal{L} = \sum_{i=1}^{I} n_i \ln P(\mathbf{Y} = \mathbf{y}_i)$$

Notice that only non-zero observed cell entries contribute to the log-likelihood function, a feature that is exploited by several more efficient LC software packages that have been developed within the past few years.

One of the problems in the estimation of LC models is that model parameters may be non-identified, even if the number of degrees of freedom is larger or equal to zero. Non-identification means that different sets of parameter values yield the same maximum of the log-likelihood function or, worded differently, that there is no unique set of parameter estimates. The formal identification check is via the information matrix which should be positive definite. Another option is to estimate the model of interest with different sets of starting values. Except for local solutions (see below), an identified model gives the same final estimates for each set of the starting values.

Although there are no general rules with respect to the identification of LC models, it is possible to provide certain minimal requirements and point at possible pitfalls. For an unrestricted LC analysis, one needs at least three indicators, but if these are dichotomous, no more than two latent classes can be identified. One has to watch out with four dichotomous variables, in which case the unrestricted three-class model is not identified, even though it has a positive number of degrees of freedom. With five dichotomous indicators, however, even a five-class model is identified. Usually, it is possible to achieve identification by constraining certain model parameters: for example, the restrictions $P(Y_\ell = 1|X = 1) = P(Y_\ell = 2|X = 2)$ can be used to identify a two-class model with two dichotomous indicators.

A second problem associated with the estimation of LC models is the presence of local maxima. The log-likelihood function of a LC model is not always concave, which means that hill-climbing algorithms may converge to a different maximum depending on the starting values. Usually, we are looking for the global maximum. The best way to proceed is, therefore, to estimate the model with different sets of random starting values. Typically, several sets converge to the same highest log-likelihood value, which can then be assumed to be the ML solution. Some software packages have automated the use of several sets of random starting values in order to reduce the probability of getting a local solution.

Another problem in LC modeling is the occurrence of boundary solutions, which are probabilities equal to zero (or one) or log-linear parameters equal to minus (or plus) infinity. These may cause numerical problems in the estimation algorithms, occurrence of local solutions, and complications in the computation of standard errors and number of degrees of freedom of the goodness-of-fit tests. Boundary solutions can be prevented by imposing constraints or by taking into account other kinds of prior information on the model parameters.

The most popular methods for solving the ML estimation problem are the Expectation-Maximization (EM) and Newton-Raphson (NR) algorithms. EM is a very stable iterative method for ML estimation with incomplete data. NR is a faster procedure that, however, needs good starting values to converge. The latter method makes use the matrix of second-order derivatives of the log-likelihood function, which is also needed for obtaining standard errors of the model parameters.

Model selection issues

The goodness-of-fit of an estimated LC model is usually tested by the Pearson or the likelihood-ratio chi-squared statistic (see CATEGORICAL DATA ANALYSIS). The latter is defined as

$$L^2 = 2\sum_{i=1}^{I} n_i \ln \frac{n_i}{N \cdot P(\mathbf{Y} = \mathbf{y}_i)},$$

where N denotes the total sample size. As in log-linear analysis, the number of degrees of freedom (df) equals the number of cells in the frequency table minus one, $\prod_{\ell=1}^{L} D_\ell - 1$, minus the number of independent parameters. In an unrestricted LC model,

$$df = \prod_{\ell=1}^{L} D_\ell - C \cdot \left[1 + \sum_{\ell=1}(D_\ell - 1)\right].$$

Although it is no problem to estimate LC models with 10, 20, or 50 indicators, in such cases the frequency table may become very sparse and, as a result, asymptotic p values can longer be trusted. An elegant, but somewhat time-consuming, solution to this problem is to estimate the p values by parametric bootstrapping. Another option is to assess model fit in lower-order marginal tables; for example, in the two-way marginal tables.

It is not valid to compare models with C and C+1 classes by subtracting their L^2 and df values because this conditional test does not have an asymptotic chi-squared distribution. This means that alternative methods are required for comparing models with different numbers of classes. One popular method is the use of information criteria such as BIC and AIC.

Another more descriptive method is a measure for the proportion of total association accounted for by a C-class model, $[L^2(1) - L^2(C)]/L^2(1)$, where the L^2 value of the one-class (independence) model, $L^2(1)$, is used as a measure of total association in the L-way frequency table.

Usually we are not only interested in goodness-of-fit, but also in the performance of the modal classification rule [see equation (3)]. The estimated proportion of classification errors under modal classification equals

$$E = \sum_{i=1}^{I} \frac{n_i}{N} \left\{ 1 - \max \left[P(X = x | \mathbf{Y} = \mathbf{y}_i) \right] \right\}.$$

This number can be compared to the proportion of classification errors based on the unconditional probabilities $P(X = x)$, yielding a reduction of errors measure λ:

$$\lambda = 1 - \frac{E}{\max \left[P(X = x) \right]}.$$

The closer this nominal R^2-type measure is to one, the better the classification performance of a model.

Extensions of the LC model for categorical indicators

Several extensions have been proposed of the basic LC model. One of the most important extensions is the inclusion of covariates or grouping variables which describe (predict) the latent variable X. This is achieved by specifying a multinomial logit model for the probability of belonging to LC x; that is,

$$P(X = x | \mathbf{Z} = \mathbf{z}) = \frac{\exp(\gamma_x^X + \sum_{k=1}^K \gamma_x^{X,Z_k} \cdot z_k)}{\sum_{r=1}^C \exp(\gamma_r^X + \sum_{k=1}^K \gamma_r^{X,Z_k} \cdot z_k)},$$

where z_k denotes a value of covariate k.

Another important extension is related to the use of information on the ordering of categories. Within the log-linear LC framework, ordinal constraints can be imposed via ASSOCIATION MODEL structures for the two-variable terms $\beta_{x,y_\ell}^{X,Y_\ell}$. For example, if Y_ℓ is an ordinal indicator, we can restrict $\beta_{x,y_\ell}^{X,Y_\ell} = \beta_x^{X,Y_\ell} \cdot y_\ell$. Similar constraints can be used for the latent variable (Heinen, 1996).

In the case that a C-class model does not fit the data, the local independence assumption fails to hold for one or more pairs of indicators. The common model fitting strategy in LC analysis is to increase the number of latent classes till the local independence assumption holds. Two extensions have been developed that make it possible to follow other strategies. Rather than increasing the number of latent classes, one alternative approach is to relax the local independence assumption by including direct effects between certain indicators – a straightforward extension to the log-linear LC model. Another alternative strategy involves increasing the number of latent variables instead of the number of latent classes. This so-called LC factor analysis approach (Magidson and Vermunt, 2001) is especially useful if the indicators measure several dimensions.

Other important extensions involve the analysis of longitudinal data (see LATENT MARKOV MODEL) and partially observed indicators. The most general model that contains all models discussed thus far as special cases is the structural equation model for categorical data proposed by Hagenaars (1990) and Vermunt (1997).

Other types of LC models

Thus far, we have focused on LC models for categorical indicators. However, the basic idea of LC analysis, that parameters of a statistical model differ across unobserved subgroups, can also be applied with variables of other scales types. In particular, there are three important types of applications of LC or finite mixture models that fall outside the categorical data analysis framework: clustering with continuous variables, density estimation, and random-effects modeling.

Over the past ten years, there has been a renewed interest in LC analysis as a tool for cluster analysis with continuous indicators. The LC model can be seen as a probabilistic or model-based variant of traditional non-hierarchical cluster analysis procedures such as the K-means method. It has been shown that such a LC-based clustering procedure outperforms the more ad hoc traditional methods. The method is known under names such as LATENT PROFILE MODEL, mixture-model clustering, model-based clustering, latent discriminant analysis, and LC clustering. The basic formula of this model is similar to one given in equation (1); that is,

$$f(\mathbf{Y}=\mathbf{y}) = \sum_{x=1}^{C} P(X=x) f(\mathbf{Y}=\mathbf{y}|X=x).$$

As shown by this slightly more general formulation, the probabilities $P(...)$ are replaced by densities $f(...)$. With continuous variables, the class-specific densities $f(\mathbf{Y} = \mathbf{y}|X = x)$ will usually be assumed to be (restricted) multivariate normal, where each LC has its own mean vector and covariance matrix. Note that this is a special case of the more general principle of density estimation by finite mixtures of simple densities.

Another important application of LC analysis is as a NONPARAMETRIC RANDOM-EFFECTS MODEL. The idea underlying this application is that the parameters of the regression model of interest may differ across unobserved subgroups. For this kind of analysis, often referred to as LC regression analysis, the LC variable serves the role of a MODERATING VARIABLE. The method is very similar to regression models for repeated measures or two-level data sets, with the difference that no assumptions are made about the distribution of the random coefficients.

Software

The first LC program, MLLSA, made available by Clifford Clogg in 1977, was limited to a relative small number of nominal variables. Today's program can handle many more variables, as well as other scale types. For example, the LEM program (Vermunt, 1997) provides a command language that can be used to specify a large variety of models for categorical data, including LC models. Mplus is a command language based structural equation modeling package that implements some kinds of LC models, but not for nominal indicators. In contrast to these command language programs, Latent GOLD is a program with an SPSS-like user interface that is especially developed for LC analysis. It implements the most important types of LC models, deals with variables of different scale types, and extends the basic model to include covariates, local dependencies, several latent variables, and partially observed indicators.

References

Goodman, L.A. (1974). The analysis of systems of qualitative variables when some of the variables are unobservable. Part I: A modified latent structure approach, *American Journal of Sociology*, 79, 1179-1259.

Haberman, S.J. (1979). *Analysis of Qualitative Data, Vol 2, New Developments.* New York: Academic Press.

Hagenaars, J.A. (1990). *Categorical Longitudinal Data - Loglinear Analysis of Panel, Trend and Cohort Data.* Newbury Park: Sage.

Hagenaars, J.A. and McCutcheon, A.L. (2002), *Applied Latent Class Analysis.* Cambridge University Press.

Heinen, T. (1996). *Latent Class and Discrete Latent Trait Models: Similarities and Differences.* Thousand Oakes: Sage Publications.

Lazarsfeld, P.F. (1950). The logical and mathematical foundation of latent structure analysis & The interpretation and mathematical foundation of latent structure analysis. S.A. Stouffer et al. (eds.), *Measurement and Prediction*, 362-472. Princeton, NJ: Princeton University Press.

Magidson, J., and Vermunt, J.K. (2001). Latent class factor and cluster models, bi-plots and related graphical displays, *Sociological Methodology*, 31, 223-264.

Vermunt, J.K. (1997). *Log-linear Models for Event Histories.* Thousand Oakes: Sage Publications.