Bayesian Analysis (2024) **TBA**, Number TBA, pp. 1–28

Domain Latent Class Models

Jesse Bowers* and Steve Culpepper†

Abstract. Latent Class Models (LCMs) are used to cluster multivariate categorical data (e.g. group participants based on survey responses). Traditional LCMs assume a property called conditional independence. This assumption can be restrictive, leading to model misspecification and overparameterization. To combat this problem, we developed a novel Bayesian model called a Domain Latent Class Model (DLCM), which permits conditional dependence. We verify identifiability of DLCMs. We also demonstrate the effectiveness of DLCMs in both simulations and real-world applications. Compared to traditional LCMs, DLCMs are effective in applications with time series, overlapping items, and structural zeroes.

Keywords: LCM, latent variable, clustering, categorical data analysis.

1 Introduction

1.1 Problem Statement

Latent Class Modeling (LCM) is a clustering technique for multivariate categorical data. LCMs are of interest in many areas including social, behavioral, health sciences, and record linkage. A common use is to group respondents based on their responses to a multiple choice survey and to interpret each of those groups.

Traditional LCMs break the respondents into $C \in \mathbb{N}$ groups called latent classes, and assume that respondents answer each question independently, conditional on class membership. Suppose a survey contains J multiple choice questions (items). Let the i'th person's response to item j be denoted $X_{ij} \in \mathbb{Z}_{Q_j} := \{0, 1, \ldots, Q_j - 1\}$ where $Q_j \in \mathbb{N}$ gives the total number of categorical values item j can take. Let $\rho_{cjx_j} = P(X_{ij} = x_j | c_i = c)$ be the probability that members of class c report $X_j = x_j$ for item j. LCMs assume that, given class membership, elements of the multivariate response vector $\underline{\boldsymbol{X}}_i = (X_{i0}, \ldots, X_{i,J-1})$ are conditionally independent. Consequently, if person i belongs to class $c_i = c$ then the probability of observing $\underline{\boldsymbol{X}}_i = \underline{\boldsymbol{x}}_i$ is given by:

$$P(\underline{\boldsymbol{X}}_i = \underline{\boldsymbol{x}} | c_i = c, \rho) = \prod_{j=0}^{J-1} \prod_{q=0}^{Q_j-1} \rho_{cjq}^{I(q=x_j)}, \tag{1}$$

where $I(\cdot)$ is the indicator function. Nominally the prior probability of the i'th subject belonging to class c is $P(c_i = c | \underline{\boldsymbol{\pi}}) = \pi_c$. Therefore, the responses to our survey follow

arXiv: 2205.08677

*Department of Statistics, University of Illinois Urbana Champaign, bowers.jesse+publications@gmail.com

†Department of Statistics, University of Illinois Urbana Champaign, sculpepp@illinois.edu

 https://doi.org/10.1214/24-BA1433

the distribution:

$$P(\underline{\boldsymbol{X}}_i = \underline{\boldsymbol{x}}|\rho, \underline{\boldsymbol{\pi}}) = \sum_{c=0}^{C-1} \pi_c \prod_{j=0}^{J-1} \prod_{q=0}^{Q_j-1} \rho_{cjq}^{I(q=x_j)}. \tag{2}$$

One challenge in traditional LCMs is the assumption of conditional independence. In practice, it is sometimes inappropriate to assume that, for a member of a class, each question is answered independently. For instance, two items might overlap, asking similar questions in different ways. Locally dependent questions also appear in time series data. Questions within the same time point may exhibit local dependence. Conversely, if the same question is asked across time points, then there may be local dependence between responses to the same question. This temporal dependence is notably present in pre-post testing with paired items.

1.2 Contribution to Past Work

A classical approach to address local dependence is via diagnostics and manual adjustments. One might fit a traditional LCM, check for local dependence, and tweak the model until the dependence disappears. There are a number of methods for detecting local dependence. Some classical methods include chi-squared tests and Fisher exact tests (Agresti, 2018). When dependence is found there are at least two techniques available to eliminate it.

The first approach is to increase the number of classes C (Bartholomew et al., 2011). With more and more groups composed of smaller and smaller populations, the groups become increasingly homogeneous and local dependence decreases. In principle, with enough latent classes local dependence disappears entirely. In a later illustrative example, we show how doubling the amount of classes accounts for the local dependence caused by two dependent questions (Appendix A, Bowers and Culpepper, 2024a). In general, local dependence disappears no later than $C = \prod_{j=0}^{J-1} Q_j$ classes, where there is one class for every possible response pattern. The weakness of removing local dependence by increasing the number of classes is that it tends to overfit. Furthermore, a large number of classes can be hard to interpret. Considering that a main objective of LCMs is to provide a parsimonious interpretation of data, increasing the number of classes to deal with local dependence is not always attractive. Given that the correct number of classes is not known, it is also easy to choose too few classes resulting in model misspecification.

The second approach is the 'Joint Variable' approach. The idea here is to transform the data itself to remove dependence. Suppose a pair of items are conditionally dependent. Those items correspond with a common *domain* and could be merged into a 'joint' variable (Goodman, 1974) as demonstrated next in Example 1.

Example 1. Consider a case with two binary items for $n = 6$ respondents. If the two binary items are put into the same domain they might be recoded as follows: $(0, 0) = 0$, $(0, 1) = 1$, $(1, 0) = 2$, and $(1, 1) = 3$.

When this technique is applied, paired items are removed and replaced with the joint variable in the dataset. This effectively removes the dependence by generating a new variable with one value for each possible response to the grouped items. The obstacle with either aforementioned correction is that they are manual, iterative, and rely on personal judgement. The manual nature requires time and effort. The iterative process means that early decisions will be made based on a biased model and personal judgement can be difficult to reproduce.

We propose a model called a Domain Latent Class Model (DLCM). DLCM is a Bayesian model which extends the joint variable approach. It is an exploratory algorithm which identifies locally dependent items and groups dependent items together into joint variables. The DLCM has several advantages. First, the model flexibly searches for local dependence, which is captured in a nonparametric way. We require no prior knowledge of which items are related nor information on the exact nature of that relationship. Second, we provide tools to readily interpret the local dependence recovered by the model. Third, we optionally permit different classes to have different dependence structures, a freedom we have not seen in competing models. Fourth, we provide rigorous identifiability conditions for DLCMs.

Recent research in the area of local dependence relies more on algorithms and less on human judgement. For instance there have been advances in the area of record linkage to handle local dependence (Daggy et al., 2014). Record linkage attempts to match records between two different data sources which lack a common key. The goal is to look at every pair of records and identify each pair as either a 'match' or 'mismatch' based on common items. In general, there are at least four broad approaches for solving local dependence:

1. Hierarchical models. Hierarchical models are typically described as a tree with the latent class up top, intermediary latent variables in the middle, and the observed responses at the end. Latent Tree Models (Chen et al., 2012) and Bayesian Pyramids (Gu and Dunson, 2023) are two examples of this. Hierarchical models are very flexible, but can be difficult to interpret. For instance latent tree models in particular suffer from the fact that any intermediary node can be interpreted as the head of tree. This leads to many competing interpretations for the same model.

2. Mixture Probit models. Asparouhov and Muthen (2011) build a mixture model assuming that binary responses $\underline{\boldsymbol{X}}_i$ are formed from latent $[\underline{\tilde{\boldsymbol{X}}}_i|c_i = c] \sim \text{Normal}(\underline{\boldsymbol{\mu}}_c, \boldsymbol{\Sigma}_c)$ under a cumulative link. Cagnone and Viroli (2012) developed a mixture of factor models to describe heterogeneity in multivariate binary response data. In record linkage, Daggy et al. (2014) fit a Gaussian mixture model with random effects to handle local dependence. When compared to DLCMs, dependence in mixture probit models is captured in a parametric way, typically with a linear relationship between items. This can be limiting.

3. Log-linear models. For every possible vector of item responses, a log-linear model will predict the number of matching observations. These models can either be used directly (Uebersax, 2009) or in a mixture (Daggy et al., 2014). Daggy et al. (2014)'s work is applied to record linkage with cross terms accounting for dependence between

known pairs of items. Log-linear models are parametric and require the investigator to hypothesize which relationships influence responses. In comparison, the DLCM requires no such prior knowledge.

4. Conditional Modes Model (CMM). Marbac et al. (2014) proposes a CMM using joint variables to solve local dependence. They use a Metropolis algorithm to search for conditionally dependent blocks of items and convert them to joint variables. Our Dependent Latent Class Models (DLCMs) also uses a Metropolis algorithm to search for dependent items, but differs from Marbac's CMMs in several ways. I. Our model includes regularization which prefers simpler models with less assumed conditional dependence. In comparison, CMM uses a uniform prior over the space of joint variables. In Monte Carlo simulations we provide evidence that our regularization more accurately uncovers the joint variable dependence structure than a uniform prior. II. We optionally permit different classes to use different joint variables where CMM assumes common joint variables across classes. III. We provide more complete identifiability conditions than Marbac. IV. Our approach is fully Bayesian whereas Marbac uses a mixture of frequentist and Bayesian techniques. A full Bayesian approach permits the use of standard inference techniques.

The remainder of the paper is organized as follows. In Section 2 we introduce the notation necessary to formalize DLCMs. Throughout the paper we customarily use examples to illustrate definitions. In Section 3, we establish sufficient conditions for generic identifiability of DLCMs. In Sections 4 and 5, we discuss the full conditional distribution of the DLCM parameters and the prior for domains. Section 6 describes the MCMC algorithm for approximating the posterior of our parameters. In Section 7 we validate the accuracy of DLCMs in simulation studies on artificial data. Section 8 showcases the power of DLCMs in real world applications. In Section 9 we provide closing thoughts. We provide an R package for running the DLCM. This is available on Github: `https://github.com/jessebowers/dependentLCM`. The package runs with respectable speed. In Section 7, we conduct some simulations, and 98% of simulations ran in two minutes[1] or less on a sample size of $n = 1{,}000$, $J = 24$ items, $C = 5$ classes, and $T = 6{,}000$ MCMC iterations. Runtime increases roughly in proportion to $n \times J \times C \times T$.

2 Domain Notation

Within a class, we segment items into conditionally independent groups called domains. A domain is a set of items which will be combined to form one joint variable. Conditional on being in class $c_i = c$, items within the same domain are dependent. However, items of one domain are conditionally independent, given class, of the items of all other domains. Given that items in the same domain are transformed into a joint variable, every possible response pattern to the grouped items is given an individual probability. With a Metropolis within Gibbs Markov chain Monte Carlo (MCMC) process, our DLCM actively searches for a promising way to recode our data to capture conditional independence.

[1]Simulations were conducted on a 2.1 GHz processor: Intel Xeon (Sapphire Rapids) 8468 CPU.

Within any class c, the items are partitioned into D disjoint sets: $J(c,1),\ldots,$ $J(c,D) \subseteq \mathbb{Z}_J := \{0,1,\ldots,J-1\}$. We allow for empty $J(c,d)$ with D typically, but not necessarily, much larger than our number of items J. In Section 5, D is used as a regularizing hyperparameter, with larger values of D promoting more finely partitioned items. Let $J_{(k)}(c,d)$ refer to the item with the k'th smallest label in domain (c,d), and in general let $S_{(k)}$ be the k'th smallest element of set S.

The DLCM domains are subject to certain restrictions. In the most restricted case, all classes are required to group items the same way and have the same domains: $J(c,d) = J(c',d)$ for all c, c', d. This is called a homogeneous DLCM. Conditional on a fixed domain structure, a homogeneous DLCM can be thought of as transforming our dataset $\boldsymbol{X}$ by merging dependent items, and then applying traditional LCM onto the transformed dataset. In the least restrictive case, each class is allowed to have different domains and group items differently. This is called a heterogeneous DLCM. In this case the recoding of responses varies from class to class.

The responses for items in the same domain and class need to be modeled jointly. We call the series of responses in a domain a 'pattern': $X_{i,J(c,d)}$. It is convenient to express these patterns as integer values. Let r_{icd} be the integer referring to the i'ths person's responses to the questions in domain (c,d). The multivariate responses to a domain are transformed to an integer using a mapping vector $\underline{\boldsymbol{V}}(S) = [V_1(S),\ldots,V_J(S)]^\top$ which takes set $S \subseteq \mathbb{Z}_J$ and produces a vector in $(\mathbb{N} \cup \{0\})^J$. Specifically, r_{icd} is defined as

$$r_{icd} := \underline{\boldsymbol{V}}(J(c,d))^\top \underline{\boldsymbol{X}}_i \in \mathbb{Z}_{R_{cd}}, \tag{3}$$

where element j of $\underline{\boldsymbol{V}}(S)$ is defined as

$$V_j(S) := \begin{cases} 0 & j \notin S, \\ 1 & j = S_{(1)}, \\ \prod_{w=1}^{m-1} Q_{S_{(w)}} & j = S_{(m)}, m > 1, \end{cases} \tag{4}$$

and

$$R_{cd} := \prod_{j \in J(c,d)} Q_j \tag{5}$$

is the total number of patterns for domain (c,d). For a given class, patterns can also be expressed as a vector $\underline{\boldsymbol{r}}_{ic}$ with values $[\underline{\boldsymbol{r}}_{ic}]_d := r_{icd}$. Variable δ_{jc} allows one to look up what domain the j'th item belongs to in class c. By definition $j \in J(c,\delta_{jc})$ and $J(c,d) = \{j : \delta_{jc} = d\}$. For a given class, the δ_{jc} can be expressed as a vector: $\underline{\boldsymbol{\delta}}_c := [\delta_{0,c},\ldots,\delta_{J-1,c}]^\top$. It can also be expressed as a $J \times C$ matrix, $\boldsymbol{\Delta}$, with elements $\boldsymbol{\Delta}_{jc} = \delta_{jc}$. Matrix $\boldsymbol{\Delta}$ completely specifies how the items are grouped into domains across all classes. We call $\boldsymbol{\Delta}$ the *domain structure*. We particularly care about nonempty domains and the patterns corresponding to these domains. Let set $\mathcal{D}_c := \{d : |J(c,d)| > 0\} \subseteq \mathbb{Z}_D$ identify the nonempty domains in class c. To illustrate this notation consider the following example:

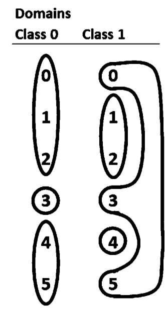
Example 2. Suppose we have a dataset with $J = 6$ items, $C = 2$ classes, and $D = 35$ domains. The first three items are binary ($Q_j = 2$), items four and five have five options ($Q_j = 5$), and the last item is binary ($Q_j = 2$). We show a possible domain structure

$$\boldsymbol{\Delta} = \begin{array}{cc} \begin{array}{cc} \underline{\boldsymbol{\delta}}_0 & \underline{\boldsymbol{\delta}}_1 \end{array} & \\ \begin{bmatrix} \delta_{0,0} & \delta_{0,1} \\ \delta_{1,0} & \delta_{1,1} \\ \delta_{2,0} & \delta_{2,1} \\ \delta_{3,0} & \delta_{3,1} \\ \delta_{4,0} & \delta_{4,1} \\ \delta_{5,0} & \delta_{5,1} \end{bmatrix} & \begin{array}{l} j=0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \end{array} = \begin{bmatrix} 6 & 2 \\ 6 & 7 \\ 6 & 7 \\ 4 & 2 \\ 8 & 6 \\ 8 & 2 \end{bmatrix}$$

(a)

$$\begin{aligned} &J(c,d) \\ &J(0,4) &&= \{3\} \\ &J(0,6) &&= \{0,1,2\} \\ &J(0,8) &&= \{4,5\} \\ &J(1,2) &&= \{0,3,5\} \\ &J(1,6) &&= \{4\} \\ &J(1,7) &&= \{1,2\} \end{aligned}$$

(b)



(c)

Figure 1: Domains in Example 2. (a) Identifier δ_{jc} indicates which domain item j belongs to under class c. Each identifier is in the range $\delta_{jc} \in \{0, \ldots, D-1\}$, with $D = 35$ in this example. (b) Set $J(c,d)$ lists all of the items belonging to domain d under class c. For all items j and classes c it follows that $j \in J(c, \delta_{jc})$. (c) Items in the same region are in the same domain. Remark: Some domains are empty: e.g. $J(0,1) = J(1,15) = \emptyset$. This is an intentional feature designed to promote regularization, described in Section 5.

$\boldsymbol{\Delta}$ in Figure 1a. The resulting domain items $J(c,d)$ are shown in Figure 1b and a visualization of the domains is shown in Figure 1c.

In general, for domain $(c,d) = (1,2)$ with $J(1,2) = \{0,3,5\}$, there are $R_{12} = (2)(5)(2) = 20$ possible patterns with $r_{i12} \in \{0,1,\ldots,19\}$. For example, for an individual who responded $\underline{\boldsymbol{X}}_i = [0,0,0,2,1,1]^\top$ the corresponding pattern r_{i12} is given by:

$$r_{i,c=1,d=2} = \underline{\boldsymbol{V}}(J(c=1,d=2))^\top \underline{\boldsymbol{X}}_i = [0,0,0,2,1,1] \cdot [1,0,0,2,0,2\cdot 5] = 14.$$

2.1 Pattern Probabilities

Patterns r_{icd} and $r_{icd'}$ are assumed to be conditionally independent for $d \neq d'$. For subject i in class c, a domain pattern has probability:

$$P(r_{icd} = r | c_i = c, \theta_{cdr}, \underline{\boldsymbol{\delta}}_c) = \theta_{cdr}, \tag{6}$$

$$P(\underline{\boldsymbol{r}}_{ic} | c_i = c, \theta_c, \underline{\boldsymbol{\delta}}_c) = \prod_{d \in \mathcal{D}_c} \prod_{r=0}^{R_{cd}-1} \theta_{cdr}^{I(r=r_{icd})}. \tag{7}$$

An empty domain d has $r_{icd} = 0$ and $\theta_{cd0} = 1$ everywhere, and is generally omitted. All probabilities for a given class can be represented as θ_c, and probabilities for a domain (c,d) can also be expressed as vector $\underline{\boldsymbol{\theta}}_{cd} = [\theta_{cd0}, \ldots, \theta_{c,d,R_{cd-1}}]^\top$. For a given class c with domains $\underline{\boldsymbol{\delta}}_c$ there is a one to one relationship between responses $\underline{\boldsymbol{X}}_i$ and patterns $\underline{\boldsymbol{r}}_{ic}$. It follows that: $P(\underline{\boldsymbol{X}}_i | c_i = c, \theta_c, \underline{\boldsymbol{\delta}}_c) = P(\underline{\boldsymbol{r}}_{ic} | c_i = c, \theta_c, \underline{\boldsymbol{\delta}}_c)$. When marginalized across class, the response probabilities are $P(\underline{\boldsymbol{X}}_i | \theta, \boldsymbol{\Delta}, \underline{\boldsymbol{\pi}}) = \sum_{c=0}^{C} \pi_c P(\underline{\boldsymbol{r}}_{ic} | c_i = c, \theta_c, \underline{\boldsymbol{\delta}}_c)$ with π_c representing the prior probability that subject i is in class c. The following example demonstrates the pattern probabilities of a class in practice:

Example 2 (continued). Continuing the previous example, suppose class 1 had the following probabilities:

$$\begin{aligned}\underline{\boldsymbol{\theta}}_{12} &= \frac{1}{100}[4,3,5,5,6,4,4,5,3,7,5,7,6,6,9,5,5,4,3,4]^\top,\\ \underline{\boldsymbol{\theta}}_{16} &= [0.20,0.28,0.18,0.25,0.09]^\top,\\ \underline{\boldsymbol{\theta}}_{17} &= [0.26,0.20,0.25,0.29]^\top.\end{aligned}$$

The probability that subject i in class 1 responds $\underline{\boldsymbol{X}}_i = [0,0,0,2,1,1]^\top$ is given by:

$$\begin{aligned}&P(\underline{\boldsymbol{X}}_i = [0,0,0,2,1,1]^\top | c_i = 1, \theta_1, \underline{\boldsymbol{\delta}}_1)\\ &\quad = P([r_{i12}, r_{i16}, r_{i17}] = [14,1,0] | c_i = 1, \theta_1, \underline{\boldsymbol{\delta}}_1)\\ &\quad = \theta_{1,2,14}\theta_{1,6,1}\theta_{1,7,0}\\ &\quad = (0.09)(0.28)(0.26) \approx 0.0066.\end{aligned}$$

When looking at the joint distribution of r_{icd} and $r_{icd'}$ it is useful to use Kronecker products. This allows us to consider a vector of probabilities rather than considering one pattern value at a time.

Definition 2.1 (Kronecker product)**.** Let vectors $\underline{\boldsymbol{Y}}_l$ and $\underline{\boldsymbol{Y}}'_l$ be of size m and m' respectively. Their Kronecker product is given by:

$$\underline{\boldsymbol{Y}} \otimes \underline{\boldsymbol{Y}}' = [Y_1Y'_1, \ldots, Y_1Y'_{m'}, Y_2Y'_1, \ldots, Y_2Y'_{m'}, \ldots, \ldots, Y_mY'_{m'}]^\perp \in \mathbb{R}^{mm'}. \tag{8}$$

We also define the column-wise Kronecker product, sometimes called the Khatri–Rao product:

$$[\underline{\boldsymbol{Y}}_1, \ldots, \underline{\boldsymbol{Y}}_k] \otimes^* [\underline{\boldsymbol{Y}}'_1, \ldots, \underline{\boldsymbol{Y}}'_k] = \left[(\underline{\boldsymbol{Y}}_1 \otimes \underline{\boldsymbol{Y}}'_1), \ldots, (\underline{\boldsymbol{Y}}_k \otimes \underline{\boldsymbol{Y}}'_k)\right]. \tag{9}$$

An immediate use of the Kronecker products is to describe the joint distribution of $r_{i,c,d\in S}$ for $S \subseteq \mathbb{Z}_D$. There exists some permutation matrix $\mathbf{\Pi}$ where:

$$P(r_{i,c,d\in S}|c_i = c, \theta, \underline{\boldsymbol{\delta}}_c) = P\left(\underline{\boldsymbol{V}}\left(\bigcup_{d\in S} J(c,d)\right)^\top \underline{\boldsymbol{X}}_i | c_i = c, \theta, \underline{\boldsymbol{\delta}}_c\right)$$

$$\begin{bmatrix} P(\underline{\boldsymbol{V}}(\bigcup_{d\in S} J(c,d))^\top \underline{\boldsymbol{X}}_i & = 0 & |c_i = c, \theta, \underline{\boldsymbol{\delta}}_c) \\ P(\underline{\boldsymbol{V}}(\bigcup_{d\in S} J(c,d))^\top \underline{\boldsymbol{X}}_i & = 1 & |c_i = c, \theta, \underline{\boldsymbol{\delta}}_c) \\ \cdots & & \\ P(\underline{\boldsymbol{V}}(\bigcup_{d\in S} J(c,d))^\top \underline{\boldsymbol{X}}_i & = \prod_{d\in S} R_{cd} - 1 & |c_i = c, \theta, \underline{\boldsymbol{\delta}}_c) \end{bmatrix} = \mathbf{\Pi} \bigotimes_{d\in S} \underline{\boldsymbol{\theta}}_{cd}.$$

That is, up to reordering, the Kronecker product $\bigotimes_{d\in S} \underline{\boldsymbol{\theta}}_{cd}$ describes the distribution of $r_{i,c,d\in S}$. We now introduce a property called Kronecker Separability.

Definition 2.2. Domain (c,d) with probabilities $\underline{\boldsymbol{\theta}}_{c,d}$ is Kronecker separable if these probabilities could be formed from two groups of independent items.

Loosely speaking, $\underline{\boldsymbol{\theta}}_{cd}$ is Kronecker separable if it can be expressed as a Kronecker product of two probability vectors. If there exist a bipartition of items $J_0 \bigsqcup J_1 = J(c,d)$

with probabilities $\underline{\boldsymbol{\theta}}^{(0)}, \underline{\boldsymbol{\theta}}^{(1)}$ where $\underline{\boldsymbol{\theta}}^{(0)} \otimes \underline{\boldsymbol{\theta}}^{(1)}$ equals $\underline{\boldsymbol{\theta}}_{cd}$ then $\underline{\boldsymbol{\theta}}_{cd}$ is Kronecker separable. When establishing equality, we force the terms of $\underline{\boldsymbol{\theta}}^{(0)} \otimes \underline{\boldsymbol{\theta}}^{(1)}$ to be reordered to match the patterns of r_{icd}.

Example 3. Suppose $J(c, d) := \{0, 1\}$ with $Q_1 = Q_2 := 2$. If $\underline{\boldsymbol{\theta}}_{cd} = [0.25, 0.25, 0.25, 0.25]^\top = [0.5, 0.5] \otimes [0.5, 0.5]$ then $\underline{\boldsymbol{\theta}}_{cd}$ is Kronecker separable. If $\underline{\boldsymbol{\theta}}_{cd} = [0.5, 0, 0.5, 0]^\top$ then both items are dependent and this is not Kronecker separable.

Kronecker separability is important because if a domain is Kronecker separable then it can be split into two domains without changing the distribution of $\underline{\boldsymbol{X}}$. Conversely if a domain is not Kronecker separable then splitting the domain would change the distribution.

3 Identifiability

An important issue for mixture models is identifiability. Without identifiability there may be many choices of parameters $\omega := (\underline{\boldsymbol{\pi}}, \theta, \boldsymbol{\Delta}) \in \Theta$ which fit our responses equally well, making inference problematic. We define identifiability as follows:

Definition 3.1. *Unique distribution.* A specific choice of parameters $\omega := (\underline{\boldsymbol{\pi}}, \theta, \boldsymbol{\Delta}) \in \Theta$ has a unique distribution if no other choice of $\omega \in \Theta$ produces the same distribution of $\boldsymbol{X}|\omega$. That is, ω has a unique distribution if for all ω' we have $\boldsymbol{X}|\omega' \overset{d}{=} \boldsymbol{X}|\omega$ only when $\omega' = \omega$. We consider ω and ω' to be the same when they have the same values up to relabeling of class and domain identifiers δ_{jc}. *Strict identifiability.* A model is strictly identifiable if every choice of $\omega \in \Theta$ has a unique distribution. *Generic Identifiability.* A model is generically identifiable if only a measure zero subset of ω does not have a unique distribution. This is with respect to the standard Lebesgue measure.

Identifiability, especially strict identifiability, can fail for categorical mixture models (Carreira-Perpiñán and Renals, 2000). Allman et al. (2009) provides some useful results for determining generic identifiability for traditional LCM. These are based on algebraic results from Kruskal (1977). Our proofs are inspired by Allman, but generally work directly with Kruskal's theorem. For details see online Appendix B.

Theorem 3.2 (Identifiability)**.** *A DLCM is generically identifiable if the following conditions are met:*

1. *Domain structure $\boldsymbol{\Delta}$ is restricted. For each allowed $\boldsymbol{\Delta}$ there must be a tripartition of items J_0, J_1, J_2 which fulfill the following. First the partitioned items must be conditionally independent for all classes: $\boldsymbol{X}_{\cdot,J_0} \perp\!\!\!\perp \boldsymbol{X}_{\cdot,J_1} \perp\!\!\!\perp \boldsymbol{X}_{\cdot,J_2}$. Second the following inequality must hold:*

$$min(\kappa_0, C) + min(\kappa_1, C) + min(\kappa_2, C) \geq 2C + 2, \tag{10}$$

$$\kappa_k := \prod_{j \in J_k} Q_j. \tag{11}$$

2. *Probabilities θ are restricted such that every θ_{cd} is Kronecker inseparable.*
3. $\pi_c > 0$ *for all* c

Corollary 3.3. *If and only if inequality (10) holds when all items are conditionally independent then the DLCM is generically identifiable across some nonzero space of domain structures.*

Proof. We can tripartition items most flexibly when all items are conditionally independent. As domain structures increasingly group items, there are additional restrictions on how items can be partitioned making (10) increasingly harder to fulfill. Therefore (10) holds for at least one domain structure if and only if it holds when every item is in its own separate domain. Conditions two and three in Theorem 3.2 are always possible to fulfill by removing a measure zero subset of our parameter space. □

Remark 1. Generally if you can identify one valid tripartition of items, this implies many possible valid domain structures. At a minimum, any domain structure which is formed by grouping items from the same partition is allowed: $J(c,d) \in \mathrm{PowerSet}(J_0) \cup \mathrm{PowerSet}(J_1) \cup \mathrm{PowerSet}(J_2)$.

Remark 2. To verify whether a particular domain structure permits generic identifiability, the following procedure can be used. Take each item and break them into as many conditionally independent blocks as possible. For homogeneous DLCMs every domain is conditionally independent. For heterogeneous DLCMs, 'pooled domains' can be used; see description in Appendix B. These blocks can then be tripartitioned. The goal is for the first two tripartitions to have at least $\kappa_k \geq C$ response patterns, and the third tripartition has at least $\kappa_k \geq 2$ response patterns. For convenience in our R package we have a deterministic greedy function which attempts to find an appropriate tripartition and verify identifiability.

4 DLCM Distributions

In this section we introduce the DLCM Bayes priors and the full conditional distributions. By design these distributions have a lot in common with the traditional LCM.

Our DLCM Bayes parameters use the following priors:

$$c_i | \underline{\boldsymbol{\pi}} \sim Cat(\underline{\boldsymbol{\pi}}), \tag{12}$$

$$\underline{\boldsymbol{\pi}} \sim Dirichlet(\underline{\boldsymbol{\alpha}}^{(c)}), \tag{13}$$

$$\underline{\boldsymbol{\theta}}_{cd} | \boldsymbol{\Delta} \sim Dirichlet(\alpha^{(\theta)} \underline{\mathbf{1}}_{R_{cd}}) = Dirichlet([\alpha^{(\theta)}, \ldots, \alpha^{(\theta)}]^\top). \tag{14}$$

To ensure identifiability (14) is restricted so $\underline{\boldsymbol{\theta}}_{cd}$ is not Kronecker separable, removing a measure zero space. These priors produce the following posteriors:

Theorem 4.1. *The DLCM full conditional distributions for* c_i, $\underline{\boldsymbol{\pi}}$, *and* $\underline{\boldsymbol{\theta}}_{cd}$ *are:*

$$P(c_i = c | \underline{\boldsymbol{X}}_i, \underline{\boldsymbol{\pi}}, \boldsymbol{\theta}, \boldsymbol{\Delta}) = \frac{\pi_c p(\underline{\boldsymbol{r}}_{c,i} | c_i = c, \theta_c, \underline{\boldsymbol{\delta}}_c)}{\sum_{c'=0}^{C-1} \pi_{c'} p(\underline{\boldsymbol{r}}_{c,i} | c_i = c', \theta_{c'}, \underline{\boldsymbol{\delta}}_{c'})}, \tag{15}$$

$$(\underline{\boldsymbol{\pi}}|\boldsymbol{X}, \theta, \boldsymbol{\Delta}, \underline{\boldsymbol{c}}) = (\underline{\boldsymbol{\pi}}|\underline{\boldsymbol{c}}) \sim Dirichlet(\underline{\boldsymbol{\alpha}}^{(c)} + \underline{\boldsymbol{n}}^{(c)}), \tag{16}$$

$$(\underline{\boldsymbol{\theta}}_{cd}|\boldsymbol{X}, \underline{\boldsymbol{\pi}}, \boldsymbol{\Delta}, \underline{\boldsymbol{c}}) = (\underline{\boldsymbol{\theta}}_{cd}|\boldsymbol{r}_{c,\cdot,d}, \underline{\boldsymbol{c}}, \underline{\boldsymbol{\delta}}_c) \sim Dirichlet(\alpha^{(\theta)}\underline{\mathbf{1}}_{R_{cd}} + \underline{\boldsymbol{n}}_{cd}^{(\theta)}), \tag{17}$$

where $\underline{\mathbf{1}}_k := [1, \ldots, 1]^\top$ *is a k-vector of ones and* $n_c^{(c)} := \sum_{i=0}^{n-1} I(c_i = c)$ *and* $n_{cdr}^{(\theta)} := \sum_{i=0}^{n-1} I(r_{icd} = r, c_i = c)$ *are elements of* $\underline{\boldsymbol{n}}^{(c)}$ *and* $\underline{\boldsymbol{n}}_{cd}^{(\theta)}$*, respectively.*

Proof. The proof of this theorem is standard, following similar lines to traditional LCM. Details can be found in online Appendix C. □

These posteriors are similar to traditional LCMs. For comparison see traditional LCM full conditional distributions given in online Appendix C. This has an important consequence for homogeneous DLCMs. For a homogeneous domain structure let $\boldsymbol{r}$ be a $n \times D$ matrix with rows $\underline{\boldsymbol{r}}_{i,c=0}$. Conditional on this domain structure $\boldsymbol{\Delta}$, the homogeneous posteriors are equivalent to traditional LCM posteriors on $\boldsymbol{r}$ in place of $\boldsymbol{X}$. In other words, for fixed $\boldsymbol{\Delta}$ we transform $\boldsymbol{X}$ into $\boldsymbol{r}$ by coding locally dependent items into new single-item domain patterns. Then we apply a traditional LCM to this transformed dataset. Homogeneous DLCMs stochastically 'search' for a promising transformation of $\boldsymbol{X}$ which supports good fit. Heterogeneous DLCMs further extend this idea allowing different classes to code responses differently.

Missing data can be handled one of two ways. If missing at random, missing values can be imputed. In an MCMC iteration, missing values would be generated based on the current class of that observation using the current response probabilities θ_c. If some but not all of the responses to a domain are unknown, the values will be imputed conditionally. This is done by sampling from probability vector $\underline{\tilde{\boldsymbol{\theta}}}_{cd}$ where $\tilde{\theta}_{cdr} \propto \theta_{cdr}$ when the response r is consistent with the observed values and $\tilde{\theta}_{cdr} = 0$ when the response r is inconsistent. If the missing values are not at random, create a new category value for each item indicating that the response is 'missing'.

If there are covariates $\underline{\boldsymbol{z}}_i$ influencing class membership, these can be incorporated by modifying $\underline{\boldsymbol{\pi}}$. In the revised model $P(c_i = c|\underline{\boldsymbol{z}}_i, \underline{\boldsymbol{\beta}}_c, \underline{\boldsymbol{\kappa}}) := \pi(\underline{\boldsymbol{\beta}}_c^\top \underline{\boldsymbol{z}}_i, c, \underline{\boldsymbol{\kappa}})$ where $\underline{\boldsymbol{\beta}}_c$ and $\underline{\boldsymbol{\kappa}}$ are new Bayes parameters. Conditional on parameters $\underline{\boldsymbol{\beta}}_c$ and $\underline{\boldsymbol{\kappa}}$, class membership $(c_i|\underline{\boldsymbol{z}}_i, \underline{\boldsymbol{\beta}}_c, \underline{\boldsymbol{\kappa}}, \boldsymbol{\theta}, \boldsymbol{\Delta})$ can be updated with a multinomial Gibbs step analogous to (15) above. Conversely $\underline{\boldsymbol{\beta}}_c$ and $\underline{\boldsymbol{\kappa}}$ can be updated conditional on $\boldsymbol{c}$. This amounts to a regression problem on multinomial responses $\boldsymbol{c}$. Imai and van Dyk (2005) and Held and Holmes (2006) both provide methods for this step.

5 Domain Prior

In this section we examine the prior distribution of the domain structure $\boldsymbol{\Delta}$. We propose a prior we call the 'bucket prior' for domains. The support of $\boldsymbol{\Delta}$ is subject to certain restrictions. Namely $\boldsymbol{\Delta}$ is restricted by the identifiability conditions given by the Kruskal inequality (10) in Theorem 3.2. The other restrictions are discussed in the two subsections below. For simplicity, we start with the simplest case: homogeneous DLCMs with bucket priors.

5.1 Homogeneous Domain with Bucket Prior

In homogeneous DLCMs every class has the same domains: $\underline{\boldsymbol{\delta}}_c = \underline{\boldsymbol{\delta}}_{c'}$ for all $c, c' \in \mathbb{Z}_C$. Therefore, we choose some representative class c and fully explain the domain structure $\boldsymbol{\Delta}$ from $\underline{\boldsymbol{\delta}}_c$. In this subsection, we introduce our homogeneous domain bucket prior and show that it encourages parsimony by means of less complex domain structures.

Within the support, every choice of $\underline{\boldsymbol{\delta}}_c \in \mathbb{Z}_D^J$ is has equal prior probability. If we neglect restrictions for a moment, this prior has a certain interpretation. An individual item would be equally likely to be in any of the D domains in class c. In this way, class c's domains are analogous to J numbered balls (items) distributed randomly among D buckets (domains). We also have a restriction $\mathsf{MaxItems} \in \{1, \ldots, J\}$ which limits the greatest number of items which can be in any individual domain. Neither of these restrictions depend on the specific labeling of $\boldsymbol{\Delta}$, even in the heterogeneous case (see online Appendix E).

Although every allowable choice of $\underline{\boldsymbol{\delta}}_c$ is equally likely, not every partition of items $\{J(c, d) : d \in \mathbb{Z}_D\}$ is equally likely. Consider the following example:

Example 4. Consider a homogeneous DLCM on $J = 2$ items. Neglect identifiability restrictions for simplicity. If $D = 1$, then the two items will always be in the same domain. If $D = 2$ then the items are equally likely to be put together in the same domain versus put into separate domains. Suppose we have more domains: $D = 20$. Then every choice of $(\delta_{0,0}, \delta_{0,1}) \in \mathbb{Z}_{20}^2$ is equally likely, but only 20/200 cases put both items in the same domain. Generally, the more domains D there are, the less likely a pair of items will appear in the same domain.

When we apply restrictions we reduce the support, and then scale up all probabilities by a common normalizing constant. In general we have the following prior probabilities:

Theorem 5.1. *For a homogeneous DLCM, suppose $P(\boldsymbol{\Delta}) > 0$. Up to proportionality, the bucket prior probability of class c's domains is given by $P(\underline{\boldsymbol{\delta}}_c) \propto 1$ with:*

$$P(\{J(c, d) : d \in \mathbb{Z}_D\}) \propto \frac{D!}{(D - |\mathcal{D}_c|)! D^J}, \tag{18}$$

$$P(\{|J(c, d)| : d \in \mathbb{Z}_D\}) \propto \frac{J!}{\prod_{k=0}^{D-1} |J(c, d)|! \ \prod_{k=1}^{J} |\{d : |J(c, d)| = k\}|!} \ \frac{D!}{(D - |\mathcal{D}_c|)! \ D^J}, \tag{19}$$

where $\mathcal{D}_c := \{d : |J(c, d)| > 0\}$ identifies the nonempty domains in class c.

Proof. Follows a standard combinatorial argument found in online Appendix E. $\square$

From a practical standpoint, we want the simplest domain structures to have the highest prior probability. In the spirit of Occam's Razor, this would bias our DLCM towards the best-fitting parsimonious models. Generally speaking, the larger D the less likely a priori any two items will end up in the same domain. We formalize this idea with the following corollary.

Most common bucket domain structures in Example 5

$S_1 := \{\lvert J(c,d)\rvert : d \in \mathcal{D}_c\}$	$P(S_1)$	# of $S_2 := \{J(c,d) : d \in \mathbb{Z}_D\}$	$\ln P(S_2)$
1x20	61.6%	1	-0.48
2,1x18	30.8%	190	-6.42
3,1x17	0.5%	1,140	-12.37
2,2,1x16	6.2%	14,535	-12.37
3,2,1x15	0.2%	155,040	-18.31
2,2,2,1x14	0.6%	581,400	-18.31
4,1x16	<0.1%	4,845	-18.31
$\cdots$	<0.1%	$\cdots$	≤ -24.26

Table 1: The most common domain structures with $J = 20$ binary items, $D = J^2 - 1$, and a bucket domain prior. Let S_1 represent the size of each domain in a domain structure: $S_1 := \{\lvert J(c,d)\rvert : d \in \mathcal{D}_c\}$. Additionally let S_2 represent how items are partitioned: $S_2 := \{J(c,d) : d \in \mathbb{Z}_D\}$.

Corollary 5.2. *For any q and J, if D is such that:*

$$D \geq J + \frac{q}{2}J(J-1) - 1. \tag{20}$$

Then we have the following inequality under a bucket prior:

$$P(\{\lvert J(c,d)\rvert : d \in \mathcal{D}_c\} = \{1,1,\ldots,1\}) \geq qP(\{\lvert J(c,d)\rvert : d \in \mathcal{D}_c\} = \{2,1,1,\ldots,1\}). \tag{21}$$

If the terms of (21) are nonzero, then exact equality in (20) provides exact equality (21).

If D is large enough to satisfy (20) with $q = 1$, then the most likely value of $\{\lvert J(\delta,c)\rvert : \delta \in \mathcal{D}_c\}$ puts each item in its own separate domain. When every item is in its own domain, we have conditional independence and DLCM is equivalent to traditional LCM – the simplest form.

By default in our distributable package we use $q = 2$ and $D = J^2 - 1$. This makes our simplest form twice as likely as the next simplest. We use these defaults in both our simulations and real world applications described in Sections 7 and 9. We next employ a toy example to demonstrate how the aforementioned choice of defaults translates to the most common domain structures a priori.

Example 5. Suppose we have $J = 20$ binary items and set $D = J^2 - 1$. We fit with a homogeneous DLCM with a bucket prior. The most common domain structures a priori are given in Table 1.

If we use $C = 2$ classes in this example, only domain structures with fewer than three non-empty domains do not satisfy sufficient conditions for generic identifiability. These domain structures fail the Kruskal condition given in (10). If we chose $\underline{\boldsymbol{\delta}}_c$ at random without restriction there would only be a 8×10^{-48} prior chance of choosing a non-identifiable domain structure. Finally, note that $\ln P(S_2)$ in Table 1 represents a penalty term towards simpler domain structures.

We have discussed that the domain prior prioritizes simpler domains for large D. Another key question is when we prefer one domain structure over another. Suppose we have two sets of parameters $\omega = (\underline{\boldsymbol{\pi}}, \theta, \boldsymbol{\Delta})$ and $\omega' = (\underline{\boldsymbol{\pi}}', \theta', \boldsymbol{\Delta}')$. Suppose domain structure $\boldsymbol{\Delta}'$ dominates $\boldsymbol{\Delta}$. We say $\boldsymbol{\Delta}'$ dominates $\boldsymbol{\Delta}$ if every domain $J(c,d)$ in $\boldsymbol{\Delta}$ is the subset of some domain $J'(c,d')$ in $\boldsymbol{\Delta}'$. As a consequence $\boldsymbol{\Delta}$ is a special case of $\boldsymbol{\Delta}'$. For any θ, there exists a Kronecker separable θ' where $\underline{\boldsymbol{X}}_i|\omega' \overset{d}{=} \underline{\boldsymbol{X}}_i|\omega$.

The prior also gives greater weight to simpler domain structures. We disallow Kronecker separability, but if θ' was almost Kronecker separable that would be allowed. There exists a Kronecker non-separable sequence θ'_t where $\underline{\boldsymbol{X}}_i|\omega'_t \overset{d}{\to} \underline{\boldsymbol{X}}_i|\omega$ on t. So we can choose ω'_t where $\underline{\boldsymbol{X}}_i|\omega'_t$ is asymptotically close to $\underline{\boldsymbol{X}}_i|\omega$. Since $\boldsymbol{\Delta}'$ is formed by merging domains in $\boldsymbol{\Delta}$, we know that $\boldsymbol{\Delta}'$ has smaller prior probability:

$$\frac{P(\{J(c,d) : d \in \mathbb{Z}_D\})}{P(\{J'(c,d) : d \in \mathbb{Z}_D\})} = \frac{D!}{(D - |\mathcal{D}_c|)!D^J} \Big/ \frac{D!}{(D - |\mathcal{D}'_c|)!D^J} = \prod_{k=|\mathcal{D}'_c|}^{|\mathcal{D}_c|-1} (D-k). \tag{22}$$

This gives a strong bias towards the simpler model. As t increases, ω'_t becomes approximately Kronecker separable. In the fully Kronecker separable case where $\underline{\boldsymbol{X}}_i|\omega' \overset{d}{=} \underline{\boldsymbol{X}}_i|\omega$ we have the following ratio:

$$\frac{P(\boldsymbol{X}, \underline{\boldsymbol{\theta}}, \underline{\boldsymbol{\pi}} = \underline{\boldsymbol{\pi}}_0, \{J(c,d) :\in \mathbb{Z}_D\})}{P(\boldsymbol{X}, \underline{\boldsymbol{\theta}}', \underline{\boldsymbol{\pi}}' = \underline{\boldsymbol{\pi}}_0, \{J'(c,d) :\in \mathbb{Z}_D\})} = \frac{P(\theta|\boldsymbol{\Delta})}{P(\theta'|\boldsymbol{\Delta}')} \prod_{k=|\mathcal{D}'_c|}^{|\mathcal{D}_c|-1} (D-k). \tag{23}$$

5.2 Prior Knowledge of Domains

In some cases, researchers may have some prior knowledge on which items are likely to be in the same domain, and which items are likely to be in different domains. We enable this by creating a weighting term $w(\boldsymbol{\Delta})$. Our bucket prior with weighting is $P(\boldsymbol{\Delta}) \propto g_b(\boldsymbol{\Delta})w(\boldsymbol{\Delta})$, where $g_b(\boldsymbol{\Delta})$ is the unweighted bucket prior defined in the previous subsection. Practitioners can specify the weighting by setting a $J \times J$ upper triangular matrix $\boldsymbol{W}$. For a homogeneous DLCM our prior weights are given by:

$$w(\boldsymbol{\Delta}) := \prod_{j<j'} w_{jj'}^{I(\delta_{jc} = \delta_{j'c})}. \tag{24}$$

In this way if items j and j' are in the same domain then the prior probability $P(\boldsymbol{\Delta})$ is adjusted by $W_{jj'}$. This can be done to either encourage ($w_{jj'} > 1$) or discourage ($0 < w_{jj'} < 1$) two items being grouped together.

Note that if $w_{jj'}$ is too large this pair of items might be placed into the same domain even if this does not improve fit of $P(\boldsymbol{X}|\boldsymbol{\Delta}, \underline{\boldsymbol{c}})$. This is usually not preferred because we would like conditionally independent items to be placed into separate domains. For this reason we recommend limiting $w_{jj'} < D - J + 1$. This deters spurious domains.

Theorem 5.3. *Take a pair of items j_1 and j_2. Suppose two domain structures $\mathbf{\Delta}$ and $\tilde{\mathbf{\Delta}}$ are the same except for the domain of item j_1. Under $\mathbf{\Delta}$ item j_1 is in its own domain: $J(c, d_1) = \{j_1\}$. Under $\tilde{\mathbf{\Delta}}$ we place j_1 into the same domain as item j_2. Restricting $w_{j_1 j_2} < D - J + 1$ guarantees $P(\{\tilde{J}(c, d) : d \in \mathbb{Z}_D\}) < P(\{J(c, d) : d \in \mathbb{Z}_D\})$ for the bucket prior.*

Proof. Follows from (22). See Appendix E. □

Under Theorem 5.3, we see that the simpler domain structure with fewer dependencies will be given greater prior weight. This means we will only prefer the more complicated domain structure if it improves the fit of $P(\boldsymbol{X}|\mathbf{\Delta}, \underline{\boldsymbol{c}})$.

5.3 Heterogenous Domain Prior

The prior for heterogeneous domain structures are constructed as follows. Consider a heterogeneous DLCM. Before applying restrictions, each $\underline{\boldsymbol{\delta}}_c$ and $\underline{\boldsymbol{\delta}}_{c'}$ are independent for $c \neq c'$. In other words, every $\mathbf{\Delta} \in \mathbb{Z}_D^{J \times C}$ in the support has equal prior probability. When we apply restrictions we reduce the support and then scale up all probabilities by a normalizing constant. This implies that the probabilities of each $\underline{\boldsymbol{\delta}}_c$ are proportional to the probabilities given without restriction. Therefore, within the support, the prior probability of $\mathbf{\Delta}$ is proportional to the product of independent probabilities. For unweighted bucket priors these probabilities are given in Theorem 5.1.

In partially heterogeneous domain structures some, but not all, of the classes are restricted to have the same domains. More specifically, there is a hyperparameter $\underline{\boldsymbol{\mathcal{E}}} \in \mathbb{Z}_C^C$. If $\mathcal{E}_c = \mathcal{E}_{c'}$ then classes c, c' have the same domains: $\delta_{jc} = \delta_{jc'}$ $\forall j$. For each group of classes with the same domains, we choose a representative class c. Before applying restrictions, the domains $\underline{\boldsymbol{\delta}}_c$ are independent. In this way the partially heterogeneous domain prior is analogous to the heterogeneous domain prior if you consider just the representative classes.

6 MCMC

In this section we discuss our algorithm to approximate the posterior distribution of our parameters $\omega|\boldsymbol{X}$. This is done by way of MCMC sampling with a Metropolis-Hastings within Gibbs sampler. This sampler generates MaxItr observations ostensibly from $\omega|\boldsymbol{X}$. We denote the t'th iteration of the MCMC parameters as $\omega^{(t)} := (\underline{\boldsymbol{\pi}}^{(t)}, \theta^{(t)}, \mathbf{\Delta}^{(t)})$. The MCMC steps are as follows:

1. Use Metropolis-Hastings to sample $\mathbf{\Delta}^{(t)}$ collapsed on θ. This step is repeated a specified number of times (nDomainIters times).
2. Sample $\theta^{(t)}$ with Gibbs using full conditional distribution as given in Theorem 4.1.
3. Sample $\underline{\boldsymbol{\pi}}^{(t)}$ with Gibbs using full conditional distribution as given in Theorem 4.1.

4. Sample $\underline{\boldsymbol{c}}^{(t)}$ with Gibbs using full conditional distribution as given in Theorem 4.1. Alternatively collapsed Gibbs may be used, collapsing on θ and $\underline{\boldsymbol{\pi}}$ using Theorem C.2 in the online Appendix.

   Increment t and return to step 1 and repeat until MaxItr iterations have been reached.

The Gibbs steps sample one parameter conditional on the others. For instance, the next value of $\theta^{(t+1)}$ is generated by the distribution of $\theta^{(t+1)}|\boldsymbol{\Delta}^{(t)}, \underline{\boldsymbol{\pi}}^{(t)}, \underline{\boldsymbol{c}}^{(t)}, \boldsymbol{X}$ given in Theorem 4.1. For more information on Gibbs sampling see Gelfand (2000).

A Metropolis algorithm works roughly by proposing a new value of $\boldsymbol{\Delta}$. Depending on how likely the proposed value is, the algorithm will either stay at its current value or move to the new value with some probability. Chib and Greenberg (1995) provide some useful details.

The Metropolis-Hastings step for the domain structure requires a dedicated discussion. The space of all possible domain structures is quite large encompassing every possible partition of J items. Our Metropolis algorithm is built to efficiently search this space with this challenge in mind.

The proposal is done by taking two domains at random and mixing items between them. Every item in $J(c, d_1) \cup J(c, d_2)$ is equally likely to end up in either domain, up to some small restrictions such as identifiability. Since the mixing procedure allows for any partition of $J(c, d_1) \cup J(c, d_2)$, it can correct potentially large problems with a candidate domain structure and escape many local minima. Consider the following examples.

Example 6. Suppose the true domain structure of a dataset includes the domains $J_{\text{truth}}(c, 0) = A = \{a_0, a_1, a_2, a_3\}$ and $J_{\text{truth}}(c, 1) = B = \{b_0, b_1, b_2, b_3\}$.

- Suppose our MCMC algorithm is at $J(c, 0) = \{a_0, a_1\}$ and $J(c, 1) = \{a_2, a_3\}$. If $J(c, 0)$ and $J(c, 1)$ are mixed, one possible proposal is $J(c, 0) = A$ and $J(c, 1) = \emptyset$. Notice the importance of moving all of the items at once. If only a single item was moved the goodness of fit might have actually worsened. A proposal of $\{a_0\}$ and $\{a_1, a_2, a_3\}$ would be worse if the predictions of a_0 were poor enough.

- Suppose our MCMC algorithm is at $J(c, 0) = A \cup B$. If $J(c, 0)$ is mixed with $J(c, d) = \emptyset$ then a possible proposal is $J(c, 0) = A$ and $J(c, d) = B$. This splits the domain into its independent parts without losing the conditional dependence within each A and B.

- Suppose our MCMC algorithm is at $J(c, 0) = \{a_0, a_1, b_0, b_1\}$ and $J(c, 1) = \{a_2, a_3, b_2, b_3\}$. If $J(c, 0)$ and $J(c, 1)$ are mixed, one possible proposal is $J(c, 0) = A$ and $J(c, 1) = B$. Again notice the importance of moving all of the items at once. If only a single item was moved the goodness of fit might have actually worsened. A proposal of $\{a_0, b_0, b_1\}$ and $\{a_1, a_2, a_3, b_2, b_3\}$ would be worse if the predictions of a_0 were poor enough.

In practice a particular proposal might not operate as described above. A different pair of domains might be chosen to be mixed. Even if the described domains are mixed, the items might be partitioned differently. However since these proposals can occur, they indeed will occur over sufficiently many iterations.

Full details on the Metropolis step for updating domains can be found in online Appendix F.

7 Simulation Studies

We conducted Monte Carlo simulation studies to validate the accuracy of the proposed DLCM algorithms. Here, we will generate random datasets following a specific distribution. Then we see how well our models recover the true underlying distribution.

We considered three types of datasets. In the first case *Traditional Data*, we generate data following the distribution of a traditional LCM. All items are conditionally independent given latent classes. The goal of the first simulation is to demonstrate that the DLCM accurately recovers conditionally independent domain structures. In the second case *Homogeneous Data*, we generate data following the distribution of a homogeneous DLCM. Conditional dependence appears, but it is the same across classes. In the third case *Heterogeneous Data*, we generate data following the distribution of a heterogeneous DLCM. Conditional dependence exists and differs across classes. For simplicity these simulated datasets use Bernoulli data, $C = 2$ classes, and $J = 24$ items. In Section 9.3 we provide a real world example of DLCMs on polytomous data and more classes.

Within each of the three scenarios for a given sample size n, we generate 500 datasets. For each dataset, we fit a number of different models representing each combination of the below:

- Model: We fit three types of models: traditional LCMs, homogeneous DLCMs, and heterogeneous DLCMs.
- Domain Prior: Each DLCM is fit with two different types of domain priors: uniform, and bucket. The uniform prior assumes that every domain structure is equally likely, and serves as a baseline comparison similar to Marbac et al. (2014). The bucket prior is our proposed regularizing prior discussed in Section 5.
- Seeding Method: We fit each DLCM under one of two initial conditions: default and random. The default seeding method chooses the initial domains to be as simple as possible, putting each item into its own domain. It also clusters similar observations together. This is done by choosing C random centers and allocating each observation to the nearest center. The random seeding method chooses initial conditions unfavorably. We initialize domains at random using the uniform prior. In this way the initial domains will typically be far from the truth. We also seed class membership using independent Bernoulli variables. This will cause starting class membership to start far from the truth. We use the random seeding method to measure performance when initial conditions are poor.

Domain Structure Mode Accuracy, Percentage				Sample Size					
Data	Model	Domain Prior	Seed	$n = 100$	200	300	400	500	1,000
Traditional	Homogeneous	Uniform	Default	0	0	0	0	0	0
		Bucket		96	96	99	99	99	99
	Heterogeneous	Uniform		0	0	0	0	0	0
		Bucket		96	96	98	97	98	98
Homogeneous	Homogeneous	Uniform		0	0	0	0	0	8
		Bucket		98	98	96	97	98	97
	Heterogeneous	Uniform		0	0	0	0	0	0
		Bucket		8	54	88	95	97	97
Heterogeneous	Heterogeneous	Uniform		0	0	0	0	0	0
		Bucket		79	99	97	98	98	99
Homogeneous	Homogeneous	Bucket	Random	98	98	97	96	97	97
Heterogeneous	Heterogeneous	Bucket		79	98	97	98	98	100

Table 2: From simulation studies with $C = 2$ classes. Across 500 generated datasets, in what percent did the most common posterior domain structure match the truth? The uniform domain prior does not regularize and assumes every domain structure is equally likely. The random seed initializes the domain structures and classes randomly. When initialized randomly, the starting domain structures and classes are typically far from the truth.

More information on how the data was generated, how the models are tuned, and more detailed results can be found in online Appendix G (Bowers and Culpepper, 2024b).

We say that a simulation recovered the true domain structure if the most common domain structure across iterations matched the truth. This is shown in Table 2. In all cases, the baseline uniform prior does quite poor. We show good recovery for homogeneous DLCMs and heterogeneous DLCMs under the bucket prior. Both the default and random seeds show good recovery as well, indicating a lack of sensitivity to starting conditions. Note that heterogeneous models require larger sample sizes owing to their complexity, and restrictive priors require larger sample sizes owing to the strength of the prior. In Appendix G, we show that the DLCMs typically reach the true domain structure early, often in the first 100 MCMC iterations. Our simulations also ran quickly with $> 98\%$ of simulations with sample sizes of $n = 1{,}000$ completing in 2 minutes or less.[2] In Appendix G, we provide additional simulations showing a lack of sensitivity to hyperparameter D, the number of 'buckets' in the bucket and domain adjusted domain priors.

In this subsection we examine DLCM performance when the starting conditions are not chosen favorably. We initialize domains at random using the uniform prior. In this way the initial domains will typically be far from truth. We also seed class membership using independent Bernoulli variables. This will cause starting class membership to start far from the truth. In Table 2, the seed of 'random' indicates these alternate starting conditions are used. We see that performance using this versus the default seed are very similar.

[2]Simulations were conducted on a 2.1 GHz processor: Intel Xeon (Sapphire Rapids) 8468 CPU.

8 Evaluating DLCMs

In this section we provide information on how DLCM models can be tuned and described. This is relevant in Section 9 where these techniques are used in applications.

Model Selection When fitting a DLCM, there is a pivotal choice of number of classes and type of domains (homogeneous versus heterogeneous). To make this choice, we recommend building competing models and comparing goodness of fit. Traditional LCM need not be compared, as the DLCM does a good job of recovering this domain structure. In our work we use WAIC (Watanabe–Akaike Information Criterion) to compare competing models. WAIC can be viewed as an approximation of leave one out cross validation. It uses log pointwise predictive density (LPPD) to measure fit on the training data, and then applies a WAIC penalty to adjust for overfitting and model complexity (Gelman et al., 2013).

Domain Prior Tuning If there is prior knowledge of which items should, or should not, be in the same domain this can be incorporated into matrix $\boldsymbol{W}$ as described in Section 5.2. The choice of bucket versus domain adjusted prior can reasonably be prespecified based on a practitioner's preferences towards greater data adaptability or regularization respectively. Due to a lack of sensitivity to the number of buckets D, we recommend leaving this to the default value.

Convergence To measure MCMC convergence we use the Gelman-Rubin statistic. We jointly measure the convergence of class probabilities $\underline{\boldsymbol{\pi}}^{(t)}$, marginalized item probabilities $P(X_{ij}|c_i = c, \omega^{(t)})$, and total log-likelihood $\ln P(\boldsymbol{X}|\omega^{(t)})$. A Gelman-Rubin value less than 1.1 is commonly considered representative of satisfactory convergence.

Heterogeneous DLCMs Under heterogeneous DLCMs, different classes may have different domains and operate off of different joint variables. Despite these differences, classes can be readily compared and contrasted. For any subset of items $\mathcal{J}$, we can always calculate $P(\underline{\boldsymbol{X}}_{i,\mathcal{J}}|c_i = c, \theta)$ for any class c. This means we can compare response probabilities across classes even when $\mathcal{J}$ does not match any domain in a class as follows. Fix a class c'. Under c', this probability can be found by breaking $\mathcal{J}$ into conditionally independent sets of items: $\mathcal{J} \cap J(c', d)$. The probability of items $\mathcal{J} \cap J(c', d)$ can be found by marginalizing $\underline{\boldsymbol{\theta}}_{cd}$. These conditionally independent probabilities can then be multiplied together to form $P(\underline{\boldsymbol{X}}_{i,\mathcal{J}}|c_i = c', \theta)$. Our R package provides a convenient way to make these calculations and compare probabilities across heterogeneous classes.

Describing Dependence When items are placed into the same domain, we want to be able to characterize the dependence between related items. We offer two measures to assess this relationship. These work by calculating the marginal probabilities of each item under the fitted model. The product of these marginal probabilities is then taken as a proxy for $\underline{\boldsymbol{\theta}}_{cd}$ under conditional independence. These conditional independent probabilities are denoted $\underline{\boldsymbol{\theta}}_{cd}^{(I)}$. By calculating the odds ratios and risk differences between $\underline{\boldsymbol{\theta}}_{cd}$

and $\underline{\boldsymbol{\theta}}_{cd}^{(I)}$ one can identify which individual patterns have higher or lower probabilities under dependence. These metrics are readily available in our R package.

Measuring Dependence For DLCMs it is possible to build a correlation-style metric measuring the overall level of dependence found in domain (c, d). We provide three metrics based on Kullback-Leibler (KL) divergence; the last measured on the scale $[0, 1]$. We measure the KL divergence between conditionally dependent probabilities $\underline{\boldsymbol{\theta}}_{cd}$ and conditionally independent probabilities $\underline{\boldsymbol{\theta}}_{cd}^{(I)}$. On log scale, KL divergence measures the expected likelihood ratio of $\underline{\boldsymbol{x}}_{iJ(c,d)}$ under $\underline{\boldsymbol{\theta}}_{cd}$ versus $\underline{\boldsymbol{\theta}}_{cd}^{(I)}$ when $\underline{\boldsymbol{\theta}}_{cd}$ is the true model. The higher the KL divergence the stronger the dependence. For domain (c, d), KL divergence is defined as:

$$D_{KL}(\underline{\boldsymbol{\theta}}_{cd}||\underline{\boldsymbol{\theta}}_{cd}^{(I)}) := \sum_{\underline{\boldsymbol{\chi}}} P(\underline{\boldsymbol{x}}_{iJ(c,d)} = \underline{\boldsymbol{\chi}}|c_i = c, \underline{\boldsymbol{\theta}}_{cd}) \ln \left(\frac{P(\underline{\boldsymbol{x}}_{iJ(c,d)} = \underline{\boldsymbol{\chi}}|c_i = c, \underline{\boldsymbol{\theta}}_{cd})}{P(\underline{\boldsymbol{x}}_{iJ(c,d)} = \underline{\boldsymbol{\chi}}|c_i = c, \underline{\boldsymbol{\theta}}_{cd}^{(I)})} \right). \tag{25}$$

For homogeneous DLCMs we can examine the KL divergence of domain d across all classes:

$$D_{KL}(\boldsymbol{\theta}_d, \underline{\boldsymbol{\pi}}||\boldsymbol{\theta}_d^{(I)}, \underline{\boldsymbol{\pi}}) := \sum_c \pi_c D_{KL}(\underline{\boldsymbol{\theta}}_{cd}||\underline{\boldsymbol{\theta}}_{cd}^{(I)}). \tag{26}$$

As defined in (25), the value of $D_{KL}(\underline{\boldsymbol{\theta}}_{cd}||\underline{\boldsymbol{\theta}}_{cd}^{(I)})$ is always bounded. We have an explicit expression for the upper bound when either all items in $J(c, d)$ have the same number of values Q_j, there are exactly two items, or when at most one item $j \in J(c, d)$ has $Q_j > \min\{Q_j : j \in J(c, d)\}$ (see Appendix D). Under these conditions, we can scale KL divergence so that it varies from zero (independence) to one (perfect dependence). We call this rescaled value the KL ratio. These metrics are readily available in our R package.

9 Real World Applications

In this section we illustrate the effectiveness of DLCMs in three real world examples. We consider applications for datasets related to issues in education, pediatric health, and adolescent sociology. The education application examines pre-post testing and highlights how DLCMs can identify local dependence between two time-points. The pediatric medical application examines a time series and highlights how DLCMs can identify local dependence within each time-point. Finally the sociological example contains overlapping questions and highlights how DLCMs can identify structural zeros.

We fit the DLCMs to each application dataset with four chains, 2,000 warmup iterations, 10,000 main iterations, and $\mathsf{nHomoItrs} = 600$. When updating classes $\underline{\boldsymbol{c}}$ our Gibbs step collapses on θ and $\underline{\boldsymbol{\pi}}$. Models are fitted with between $C = 1$ and $C = 8$ classes, and goodness of fit is compared based on WAIC. Otherwise these examples were executed with the same hyperparameters as the simulation studies.

As a point of comparison we also fit a latent tree model built under a package provided by Obermeyer (2017) and modeled emulating Zhang and Poon (2016). We restrict model hyperparameters to guarantee 'regularity', a necessary condition for identifiability, based on the conditions given in Chen et al. (2012). To measure goodness of fit of this latent tree model we use the total log-likelihood formed from five-fold cross validation. In the below applications, we highlight the interpretability of DLCMs, one area which latent tree models can struggle.

9.1 Education Application

In this experiment, participants' skill with probability theory was assessed. The study followed a pre-post design. First, participants are given a 12-question pre-test (questions B101–B112). Then participants were randomly given one of two treatments. Finally, participants are given a 12-question post-test with matched items (B201–B212). Each post-test question mirrored a pre-test question with slightly different numbers or labels. For example, B105 from the pre-test matches with B205 in the post-test. All questions are listed in online Appendix H (Bowers and Culpepper, 2024c). This data was collected by Anselmi et al. (2013) and is freely available in the 'pks' R package (Heller and Wickelmaier, 2013). For each question we examined the Bernoulli responses: 1 for correct and 0 for incorrect.

Subjects were eliminated if they responded too quickly, responded too slowly, or did not answer every question. This left $n = 345$ participants considered. Both treatments were assessed within the same latent class model without differentiation. A potential goal was to find relationships between our latent classes and the treatments.

In addition to the DLCMs, we also fit a confirmatory model. This model assumes a priori that paired items are dependent, puts these pairs into the same domain, and leaves the domains as fixed. When all models were compared, the homogeneous DLCM with three classes and a bucket prior performed best. In Table 3 we see that the homogeneous DLCM has both a higher likelihood (LPPD) and smaller model complexity (WAIC penalty) compared to the traditional model. The homogeneous DLCM outperforms the latent tree model with a cross validated total log-likelihood of $-2{,}670$ and $-2{,}684$ respectively. This homogeneous DLCM shows good MCMC convergence with a multivariate Gelman-Rubin statistic of 1.02.

Model Name	Prior	# of Classes	LPPD	WAIC Penalty	WAIC
Traditional LCM	–	7	−2,503	127	5,260
Confirmatory LCM	–	5	−2,483	116	5,198
Homogeneous DLCM	Bucket	3	−2,502	83	5,170
Heterogenous DLCM	Bucket	4	−2,506	95	5,202

Table 3: Education Application. Goodness of fit for top models. Between $C = 1$ and $C = 8$ classes are evaluated.

The homogeneous DLCM fits well with three classes. There are proficient students (80% of participants), beginners (17%), and students who performed worse in their post-test than their pre-test (3%). Each latent class has a roughly even amount of subjects from each treatment, and a chi-squared test shows no evidence of dependence between class and treatment ($p = 0.6$). Information on these classes including response probabilities can be found in online Appendix H.

Table 4 reports the most frequently visited domain structures. The mode domain structure contains three pairs of pre/post items: {b105,b205}; {b108,b208}, {b109,b209}. For these paired items, the participants typically got a pair both right or both wrong. The homogeneous DLCM also grouped some pre-test items which were especially difficult: {b104,b110,b111,b112}. In this domain participants have heavy concentrations on 'all right' and 'all wrong', possibly owing to the skill level needed to solve these problems. See online Appendix H for details.

Domains ({Domain1}; {Domain2}; . . .)	% of Iterations
{b104,b110,b111,b112}; {b105,b205}; {b108,b208}; {b109,b209}	87.3%
{b104,b110,b111}; {b105,b205}; {b108,b208}; {b109,b209}	4.0%
{b104,b110,b111,b112}; {b105,b205}; {b108,b208}; {b109,b209}; {b106,b107}	2.5%

Table 4: Education Application. Most Common Homogeneous Domain Structures. Domains with a single item omitted. This homogeneous DLCM is fitted with $C = 3$ classes and a bucket prior.

Compared to a traditional model, the homogeneous DLCM produced both a simpler and more accurate fit. The homogeneous DLCM was able to identify paired questions in the pre-post design and incorporate their conditional dependence into its model.

			Odds Ratios of Different Response Patterns			
Domain	KL Divergence	KL Ratio	Response=(0,0)	(1,0)	(0,1)	(1,1)
{b104,b110,b111,b112}	0.193	0.093				
{b105,b205}	0.049	0.071	2.4	1/1.7	1/1.6	1.3
{b108,b208}	0.061	0.088	3.7	1/1.8	1/1.6	1.3
{b109,b209}	0.134	0.193	2.2	1/2.8	1/2.0	1.5

Table 5: Education Application. The higher the KL divergence, the greater the level of dependence. KL ratio scales KL divergence from zero (conditional independence) to one (perfect dependence). The odds ratios compare the odds of a particular response under dependence (numerator) to conditional independence (denominator). These metrics are collapsed on class membership. Class-by-class metrics can also be interesting (see Appendix H). Calculations above are based on the posterior expected value of θ and $\underline{\boldsymbol{\pi}}$ under a homogeneous DLCM with $C = 3$ classes and a bucket prior.

9.2 Medical Application

This application is drawn from a Childhood Prevention Study (CAPS) by Mihrshahi et al. (2001) (available in the R randomLCA package). This is a longitudinal study which investigated children originally under two years of age. Every six months for two years these children were tracked for nighttime coughing, wheezing, itchy rashes, and flexural dermatitis. This creates four time periods where we indicate the absence (0) or presence (1) of each symptom. We removed any observations with missing data leaving $n = 533$ subjects.

When models were compared, a homogeneous DLCM with bucket prior fit best. The homogeneous DLCM was effective with four classes. Traditional LCM performed worst requiring eight or more classes. The homogeneous DLCM has both less model complexity and improved fit compared to the traditional model (Table 6). The latent tree model and homogeneous DLCM perform almost identically with −4,381.5 and −4,381.4 cross validated total log-likelihood respectively. We have adequate convergence with a multivariate Gelman-Rubin statistic of < 1.025 for the homogeneous DLCM.

Model Name	Prior	# of Classes	LPPD	WAIC Penalty	WAIC
Traditional LCM	–	8	−4,383	136	9,037
Homogeneous DLCM	Bucket	4	−4,271	78	8,698
Heterogenous DLCM	Bucket	4	−4,286	94	8,760

Table 6: Medical Application. Goodness of fit for top models. Between $C = 1$ and $C = 8$ classes are evaluated.

Homogeneous DLCM provides four classes: bad lungs (30% of subjects), bad skin (17%), bad all symptoms (14%), and good all symptoms (39%). Symptoms generally improve as time goes on. See Table 7 in the supplemental appendix for symptom prevalence within each class.

The most common domain structure can be found in Table 7. This domain structure pairs related symptoms for the same visit. The two lung issues are related: nighttime coughing and wheezing. Similarly, the two skin issues are related: itchy rashes and flexural dermatitis. Paired symptoms are typically comorbid. See online Appendix I for details.

Domains	% of Iterations
{IR.1,FD.1}; {IR.2,FD.2}; {IR.3,FD.3}; {IR.4,FD.4}; {NC.1,W.1}; {NC.3,W.3}; {NC.4,W.4};	95.3%
{IR.1,FD.1}; {IR.2,FD.2}; {IR.3,FD.3}; {IR.4,FD.4}; {NC.1,W.1}; {NC.2,W.2}; {NC.3,W.3}; {NC.4,W.4}	4.7%
All Others	$< 0.1\%$

Table 7: Medical Application. Most Common Homogeneous Domains. NC=NightCough, W=Wheeze, IR=ItchyRash, FD=FlexDerma. Domains with a single item omitted. This homogeneous DLCM is fitted with $C = 4$ classes and a bucket prior.

Overall, the homogeneous model was able to produce a simpler and more accurate model compared to the traditional LCM. In the context of longitudinal data, the homogenous DLCM was able to identify dependence between questions at the same time-point.

9.3 Sociology Application

This application was drawn from a CDC Youth Risk Behavior Survey (YRBS) (Centers for Disease Control and Prevention, 2017). We isolated thirteen questions about sexual violence and sexual risk (e.g. STDs) answered by high school women. Each response is

Domain	KL Divergence	KL Ratio	Odds Ratios of Different Response Patterns Response=(0,0)	(1,0)	(0,1)	(1,1)
{IR.1, FD.1}	0.153	0.220	1.5	1/1.9	1/5.1	1.7
{IR.2, FD.2}	0.138	0.199	1.4	1/1.8	1/4.7	1.7
{IR.3, FD.3}	0.098	0.141	1.3	1/1.7	1/3.7	1.8
{IR.4, FD.4}	0.076	0.110	1.3	1/1.5	1/3.2	1.9
{NC.1, W.1}	0.039	0.057	1.3	1/1.3	1/1.7	1.3
{NC.3, W.3}	0.047	0.068	1.2	1/1.3	1/2.0	1.5
{NC.4, W.4}	0.046	0.067	1.2	1/1.3	1/2.5	1.6

Table 8: Medical Application. The higher the KL divergence, the greater the level of dependence. KL ratio scales KL divergence from zero (conditional independence) to one (perfect dependence). The odds ratios compare the odds of a particular response under dependence (numerator) to conditional independence (denominator). These metrics are collapsed on class membership. Class-by-class metrics can also be interesting (see Appendix H). Calculations above are based on the posterior expected value of θ and $\underline{\boldsymbol{\pi}}$ under a homogeneous DLCM with $C = 4$ classes and a bucket prior. NC=NightCough, W=Wheeze, IR=ItchyRash, FD=FlexDerma.

polytomous with 2–5 possible responses depending on the question. We removed any observations with missing data leaving $n = 1{,}295$ subjects.

When models were compared, heterogeneous DLCM with three classes and a bucket prior fit best. For details see Table 9. The latent tree model and homogeneous DLCM perform almost identically with $-6{,}069.4$ and $-6{,}069.6$ cross validated total-log-likelihood respectively. The heterogeneous DLCM has good convergence with a multivariate Gelman-Rubin statistic of < 1.01.

Model Name	Prior	# of Classes	LPPD	WAIC Penalty	WAIC
Traditional LCM	–	7	−5,690	92	11,565
Homogeneous DLCM	Bucket	3	−5,687	52	11,479
Heterogenous DLCM	Bucket	3	−5,652	57	11,419

Table 9: Sociology Application. Goodness of fit for top models. Between $C = 1$ and $C = 8$ classes are evaluated.

The heterogeneous DLCM identified three classes: class 0 'Sexually Active' (27% of participants), class 1 'Not Sexually Active' (56%), and class 2 'At Risk' (17%). The marginal response probabilities can be found in online Appendix J.

The most common domain structures can be found in Table 10. Questions Q20 and Q21 are two questions about frequency of sexual violence (see Table 11). They form an almost triangular structure with $Q20 \geq Q21$. Questions Q64 and Q65 are overlapping questions about contraceptive use. For details on all questions see online Appendix J.

Class 0 'At Risk' has the most complex local dependencies with domains {Q19, Q20,Q21} and {Q64,Q65}. Class 2 'Sexually Active' is almost as complex with domains {Q20,Q21} and {Q64,Q65}. Dependence on Q19 was not important because 97% of participants answered this question 'no'. Class 1 'Not Sexually Active' is the least com-

	Domains ({Domain1}; {Domain2}; ...)	% of Iterations
Class0:	{Q20,Q21}; {Q64,Q65};	
Class1:	{Q20,Q21};	76.4%
Class2:	{Q19,Q20,Q21}; {Q64,Q65};	
Class0:	{Q20,Q21}; {Q64,Q65};	
Class1:	{Q20,Q21};	19.1%
Class2:	{Q19,Q20,Q21}; {Q64,Q65}; {Q61,Q62};	
	All others	$< 5\%$

Table 10: Sociology Application. Most common heterogeneous domain structures. Domains with a single item are omitted. This heterogeneous DLCM is fitted with $C = 3$ classes and a bucket prior.

plex with domain {Q20,Q21}. Responses in class 1 were very concentrated with 11/13 questions each having $\geq 95\%$ of their responses associated to a single value indicating they never had sex (online Appendix J).

Selected Questions from Survey.

Q19: Have you ever been physically forced to have sexual intercourse when you did not want to?
Q20: During the past 12 months, how many times did anyone force you to do sexual things that you did not want to do?
Q21: During the past 12 months, how many times did someone you were dating or going out with force you to do sexual things that you did not want to do?

Q64: The last time you had sexual intercourse, did you or your partner use a condom?
Q65: The last time you had sexual intercourse, what one method did you or your partner use to prevent pregnancy?

Table 11: Sociology Application. Brackets indicate items that are commonly put into the same domain. For other questions see online Appendix J.

The heterogeneous DLCM performed well in this example. In domain {Q20,Q21}, we recover near structural zeros. In domain {Q64,Q65}, we identify overlapping questions. Different classes have different domains because some classes are heavily concentrated on certain values, reducing the need to manage dependence in corresponding questions.

10 Future Work

One avenue of future exploration is introducing a more informative prior on θ. Currently we use a flat Dirichlet prior on θ, and conduct regularization by way of our domain structure $\boldsymbol{\Delta}$. This regularization could be improved via decreasing the prior density of θ near the Kronecker separable case. This would ensure dependence in θ when items are put into the same domain. Future research could consider developing a type of penalized complexity prior (e.g. see Simpson et al., 2017) to effectively shrink more

Domain	Class0	Class1	Class2
KL Divergence			
{Q20,Q21}	0.164	0.146	–
{Q19,Q20,Q21}	–	–	0.418
{Q64,Q65}	0.417	–	0.334
KL Ratio			
{Q20,Q21}	0.149	0.133	–
{Q64,Q65}	0.601	–	0.482

Table 12: Sociology Application. The higher the KL divergence, the greater the level of dependence. KL ratio scales KL divergence from zero (conditional independence) to one (perfect dependence). Calculations above are based on the posterior expected value of θ under a heterogeneous DLCM with $C = 3$ classes and a bucket prior. Domain {Q19,Q20,Q21} is not included under KL Ratio because it fails the sufficient conditions for this ratio given in Section 8.

complex domains structures away from problematic Kronecker separable cases. This can be achieved by having the prior probability of θ depend on the KL divergence given in (25). To implement such an approach certain computational hurdles would need to be cleared. Currently we collapsed on θ when updating domains $\boldsymbol{\Delta}$, and this depends on conjugacies with the Dirichlet distribution. These conjugacies would not necessarily hold under this enhancement.

In a future update to the R package, we plan on allowing covariates and missing data into the code. Enhancing the package in this way will increase the applications in which it can be readily applied.

11 Conclusion

In the presence of conditional dependence, traditional LCMs tend to overfit with too many classes. Even with additional classes, traditional LCMs may suffer from model mis-specification leading to poor goodness of fit.

We proposed a Domain LCM (DLCM) model to account for these dependencies. The DLCM works by grouping together conditionally dependent items into conditionally independent domains. We verified the generic identifiability of this model. We also demonstrated the effectiveness of DLCMs in simulation studies and real world applications. In applications we demonstrated that DLCMs are particularly effective at analyzing time series data, pre-post testing, overlapping items, and structural zeros.

Acknowledgments

We would like to thank Theren Williams, Eric Wayman, and Dr. Kristen Lee for providing valuable feedback on writing style.

Supplementary Material

Appendices A–F (DOI: 10.1214/24-BA1433SUPPA; .pdf). The Theory Supplemental File contains Appendices A–F. Appendix A Motivating Example describes how a single dependent item can double the amount of classes a traditional LCM requires. Appendix B Identifiability provides identifiability proofs. Appendix C DLCM Posteriors provides proofs for the full conditional distributions used in the DLCM. Appendix D KL Divergence Maxima provides proofs for the KL ratio calculation. Appendix E Domain Prior provides proofs related to the distribution of the domain priors. Appendix F MCMC describes the Monte Carlo Markov Chain steps in detail.

Appendix G (DOI: 10.1214/24-BA1433SUPPB; .pdf). The Simulations Supplemental File provides Appendix G Simulation Studies. Additional results are the simulation studies are given here.

Appendices H–J (DOI: 10.1214/24-BA1433SUPPC; .pdf). The Applications Supplemental File provides Appendices H–J. These appendices provide additional information about the education application, the medical application, and the sociology application respectively.

References

Agresti, A. (2018). *An Introduction to Categorical Data Analysis*. Wiley Series in Probability and Statistics. Standards Information Network, 3 edition. MR3887567. 2

Allman, E. S., Matias, C., and Rhodes, J. A. (2009). "Identifiability of parameters in latent structure models with many observed variables." *Annals of Statistics*, 37(6A): 3099–3132. MR2549554. doi: https://doi.org/10.1214/09-AOS689. 8

Asparouhov, T. and Muthen, B. (2011). "Using Bayesian priors for more flexible latent class analysis." In *Section on Government Statistics*. URL https://www.statmodel.com/download/BayesLCA%20-%20official%20version.pdf 3

Bartholomew, D. J., Knott, M., and Moustaki, I. (2011). *Latent variable models and factor analysis*, Chapter 1.4. Wiley Series in Probability and Statistics. Hoboken, NJ: Wiley-Blackwell, 3 edition. MR2849614. doi: https://doi.org/10.1002/9781119970583. 2

Bowers, J. and Culpepper, S. (2024a). "Supplementary Material A–F for "Domain latent class models"." *Bayesian Analysis*. doi: https://doi.org/10.1214/24-BA1433SUPPA. 2

Bowers, J. and Culpepper, S. (2024b). "Supplementary Material G for "Domain latent class models"." *Bayesian Analysis*. doi: https://doi.org/10.1214/24-BA1433SUPPB. 17

Bowers, J. and Culpepper, S. (2024c). "Supplementary Material H–J for "Domain latent class models"." *Bayesian Analysis*. doi: https://doi.org/10.1214/24-BA1433SUPPC. 20

Cagnone, S. and Viroli, C. (2012). "A factor mixture analysis model for multivariate binary data." *Statistical Modelling*, 12(3): 257–277. MR3179502. doi: https://doi.org/10.1177/1471082X1101200303. 3

Carreira-Perpiñán, M. Á. and Renals, S. (2000). "Practical identifiability of finite mixtures of multivariate Bernoulli distributions." *Neural Computation*, 12(1): 141–152. doi: https://doi.org/10.1162/089976600300015925. 8

Centers for Disease Control and Prevention (2017). "2017 Youth Risk Behavior Survey Data." Technical report, CDC. 22

Chen, T., Zhang, N. L., Liu, T., Poon, K. M., and Wang, Y. (2012). "Model-based multidimensional clustering of categorical data." *Artificial Intelligence*, 176(1): 2246–2269. MR2896724. doi: https://doi.org/10.1016/j.artint.2011.09.003. 3, 20

Chib, S. and Greenberg, E. (1995). "Understanding the Metropolis-Hastings Algorithm." *The American Statistician*, 49(4): 327–335. doi: https://doi.org/10.1080/00031305.1995.10476177. 15

Daggy, J., Xu, H., Hui, S., and Grannis, S. (2014). "Evaluating latent class models with conditional dependence in record linkage." *Statistics in Medicine*, 33(24): 4250–4265. MR3267408. doi: https://doi.org/10.1002/sim.6230. 3

Gelfand, A. E. (2000). "Gibbs sampling." *Journal of the American Statistical Association*, 95(452): 1300–1304. MR1825281. doi: https://doi.org/10.2307/2669775. 15

Gelman, A., Hwang, J., and Vehtari, A. (2013). "Understanding predictive information criteria for Bayesian models." *Statistics and Computing*, 24(6): 997–1016. MR3253850. doi: https://doi.org/10.1007/s11222-013-9416-2. 18

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(5): 1179–1259. doi: https://doi.org/10.1086/225676. 2

Gu, Y. and Dunson, D. B. (2023). "Bayesian Pyramids: Identifiable multilayer discrete latent structure models for discrete data." *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(2): 399–426. URL https://arxiv.org/abs/2101.10373. MR4726971. doi: https://doi.org/10.1093/jrsssb/qkad010. 3

Held, L. and Holmes, C. C. (2006). "Bayesian auxiliary variable models for binary and multinomial regression." *Bayesian Analysis*, 1(1). MR2227368. doi: https://doi.org/10.1214/06-BA105. 10

Heller, J. and Wickelmaier, F. (2013). "Minimum discrepancy estimation in probabilistic knowledge structures." *Electronic Notes in Discrete Mathematics*, 42: 49–56. MR3081845. doi: https://doi.org/10.1016/j.endm.2013.05.145. 20

Imai, K. and van Dyk, D. A. (2005). "A Bayesian analysis of the multinomial probit model using marginal data augmentation." *Journal of Econometrics*, 124(2): 311–334. MR2125369. doi: https://doi.org/10.1016/j.jeconom.2004.02.002. 10

Kruskal, J. B. (1977). "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics." *Linear Algebra and Its Applications*, 18(2): 95–138. MR0444690. doi: https://doi.org/10.1016/0024-3795(77)90069-6. 8

Marbac, M., Biernacki, C., and Vandewalle, V. (2014). "Finite mixture model of conditional dependencies modes to cluster categorical data." 4, 16

Mihrshahi, S., Peat, J. K., Webb, K., Tovey, E. R., Marks, G. B., Mellis, C. M., and Leeder, S. R. (2001). "The Childhood Asthma Prevention Study (CAPS)." *Controlled Clinical Trials*, 22(3): 333–354. doi: https://doi.org/10.1016/s0197-2456(01)00112-x. 21

Obermeyer, F. (2017). "A Bayesian latent tree model of high-dimensional heterogeneous data." URL https://github.com/posterior/treecat 20

Simpson, D., Rue, H., Riebler, A., Martins, T. G., and Sørbye, S. H. (2017). "Penalising model component complexity: a principled, practical approach to constructing priors." *Statistical Science*, 32(1). MR3634300. doi: https://doi.org/10.1214/16-STS576. 24

Uebersax, J. (2009). "A Practical Guide to Conditional Dependence in Latent Class Models." https://www.john-uebersax.com/stat/condep.htm. 3

Zhang, N. L. and Poon, L. K. M. (2016). "Latent Tree Analysis." URL https://arxiv.org/abs/1610.00085 20