The Journal of Open Source Software

ddtlcm: An R package for overcoming weak separation in Bayesian latent class analysis via tree-regularization

Mengbing Li [1¶], **Bolin Wu**[2], **Briana Stephenson**[3], **and Zhenke Wu** [1¶]

1 Department of Biostatistics, University of Michigan **2** Department of Computer Science, University of Michigan **3** Department of Biostatistics, Harvard University ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

In many scientific contexts, populations of interest can often be partitioned into meaningful homogeneous subgroups, which may not be directly observed. In finite samples, this statistical task based on subjects' measured features is termed "clustering". In particlar, latent class models (LCMs) is such a model-based clustering tool frequently used by social, behavioral, and medical researchers to cluster sampled individuals based on categorical responses. Traditional applications of LCMs often focus on scenarios where clear class separation is evident, resulting in well-defined and easily distinguishable classes. However, in numerous real-world applications, weak class separation is a common challenge and long-standing issue. For example, nutritional epidemiologists often encounter weak class separation when deriving dietary patterns from dietary intake assessment, which are defined by a few classes each containing a vector of probabilities of exposure to a set of diet components (Li et al., 2023). LCM-derived dietary patterns can exhibit strong similarities, or weak separation, resulting in numerical and inferential instabilities that challenge scientific interpretation. This issue is exacerbated in small-sized subpopulations.

To address these issues, we have developed an R package `ddtlcm` that empowers LCMs to account for weak class separation. This package implements a tree-regularized Bayesian LCM that leverages statistical strength between latent classes to make better estimates using limited data. With a tree-structured prior distribution over class profiles, classes that share proximity to one another in the tree are shrunk towards ancestral classes *a priori*, with the degree of shrinkage varying across pre-specified major item groups. The `ddtlcm` package takes data on multivariate binary responses over items in pre-specified major groups, and generates statistics and visualizations based on the inferred tree structures and LCM parameters. Overall, `ddtlcm` provides tools specifically designed to enhance the robustness and interpretability of LCMs in the presence of weak class separation, particularly useful for small sample sizes.

Statement of Need

A number of packages are capable of fitting LCMs in R. For example, `poLCA` (Linzer & Lewis, 2011) is a fully featured package that fits LCMs and latent class regression on polytomous outcome variables. `BayesLCA` (White & Murphy, 2014) is designed for LCMs in a Bayesian setting, incorporating three algorithms: expectation-maximization, Gibbs sampling, and variational Bayes approximation. `randomLCA` (Beath, 2017) provides tools to perform LCMs with individual-specific random effects. These packages focus on LCMs where the class profiles are well-separated. Directly applying these packages to data that suffer from weak separation may result in large standard deviations of the class profiles, tendency to merge similar classes, and inaccurate individual class membership assignments. These phenomena are excerbated especially when the sample size is small.

The Journal of Open Source Software

The package `ddtlcm` implements the tree-regularized LCM proposed in Li et al. (2023), a general framework to facilitate the sharing of information between classes to make better estimates of parameters using limited data. The model addresses weak separation for small sample sizes by (1) sharing statistical strength between classes guided by an unknown tree, and (2) accounting for varying degrees of shrinkage across major item groups. The proposed model uses a Dirichlet diffusion tree (DDT) process (Neal, 2003) as a fully probabilistic device to specify a prior distribution for the class profiles on the leaves of an unknown tree (hence termed "DDT-LCM"). Classes positioned closer on the tree exhibit more profile similarities. The degrees of separation by major item groups are modeled by item-group-specific diffusion variances.

Usage

In the following, we use an example list of model parameters, named "data_hchs", that comes with the package to demonstrate the utility of `ddtlcm` in deriving weakly separated latent class profiles. We start with demonstrating how a simulated dataset is generated. We next apply the primary model fitting function to the simulated dataset. Finally we summarize the fitted model and visualize the result.

Data Loading

The "data_hchs" contains model parameters obtained from applying DDT-LCM to dietary assessment data described in Li et al. (2023). We use it as a semi-synthetic data-generating mechanism to mimic the weak separation issue in the real world. Specifically, the data set includes a tree named "tree_phylo" (class "phylo"), a list of $J = 78$ food item labels, and a list of $G = 7$ pre-defined major food groups to which the food items belong. The food groups are dairy, fat, fruit, grain, meat, sugar, and vegetables.

```
install.packages("ddtlcm")
library(ddtlcm)
# load the data
data(data_hchs)
# unlist the elements into variables in the global environment
list2env(setNames(data_hchs, names(data_hchs)), envir = globalenv())
# look at items in group 1
g <- 1
# indices of the items in group 1
item_membership_list[g]
```

```
[[1]]
 [1]  1  2  3  4  5  6  7  8  9 10 11
```

```
# names of the items in group 1. The name of the list element is
# the major food group
item_name_list[g]
```

```
$Dairy
 [1] "dairy_1"  "dairy_2"  "dairy_3"  "dairy_4"  "dairy_5"
    "dairy_6"  "dairy_7"  "dairy_8"  "dairy_9"  "dairy_10"
[11] "dairy_11"
```

Data Simulation

Data simulation given the true parameter values in `data_hchs` is handled by the `simulate_lcm_given_tree()` function. Following the dietary assessment example, we simulate a multivariate binary data matrix of $N = 496$ subjects over the $J = 78$ food items, from

$K = 6$ latent classes along "tree_phylo". The resulting class profiles are weakly separated. Note that the number of latent classes equals the number of leaves in "tree_phylo".

```
# number of individuals
N <- 496
# random seed to generate node parameters given the tree
seed_parameter = 1
# random seed to generate multivariate binary observations from LCM
seed_response = 1
# simulate data given the parameters
sim_data <- simulate_lcm_given_tree(tree_phylo, N,
    class_probability, item_membership_list, Sigma_by_group,
    root_node_location = 0, seed_parameter = 1, seed_response = 1)
```

Model Fitting

The primary model fitting function is `ddtlcm_fit()`, which implements a hybrid Metropolis-Hastings-within-Gibbs algorithm to sample from the posterior distribution of model parameters. We assume that the number of latent classes $K = 6$ is known. To use `ddtlcm_fit()`, we need to specify the number of classes (`K`), a matrix of multivariate binary observations (`data`), a list of item group memberships (`item_membership_list`), and the number of posterior samples to collect (`total_iters`). For the purpose of quick illustration, here we specify a small number `total_iters = 100`.

```
set.seed(999)
# number of latent classes, or number of leaves on the tree
K <- 6
result_hchs <- ddtlcm_fit(K = K, data = sim_data$response_matrix,
  item_membership_list = item_membership_list, total_iters = 100)
print(result_hchs)
```

```
-------------------------------------------
DDT-LCM with K = 6 latent classes run on 496 observations and 78
items in 7 major groups. 100 iterations of posterior samples drawn.
-------------------------------------------
```

Model Summary

We next summarize the posterior samples using the generic function `summary()`. We discard the first 50 iterations as burn-in's (`burnin = 30`). To deal with identifiability of finite mixture models, we perform post-hoc label switching using the Equivalence Classes Representatives (ECR) method by specifying `relabel = TRUE`. To save space in the document, we do not print the summary result here (`be_quiet = TRUE`).

```
burnin <- 50
summarized_result <- summary(result_hchs, burnin, relabel = TRUE,
  be_quiet = TRUE)
```

Visualization

A generic `plot()` function is available in the package to visualize the summarized result. By specifying `plot_option = "all"`, we plot both the *maximum a posterior* (MAP) tree and the class profiles. Plotting only the tree or the class profiles is also available through `plot_option = "profile"` or `plot_option = "tree"`.

Figure 1 displays the result obtained from the above model summary. On the left shows the MAP tree structure over the latent classes. The numbers on the branches indicate the corresponding branch lengths. On the right shows class profiles for the $J = 78$ food items

belonging to $G = 7$ pre-defined major food groups, distinguished by different colors. The description of individual items is provided in Table S6.1 in the supplement of (Li et al., 2023). The numbers after the class labels in the titles of the subplots indicate class prevalences along with 95% credible intervals. Error bars show the 95% credible intervals of item response probabilities from the posterior distribution.

```
plot(x = summarized_result, item_name_list = item_name_list,
  plot_option = "all")
```

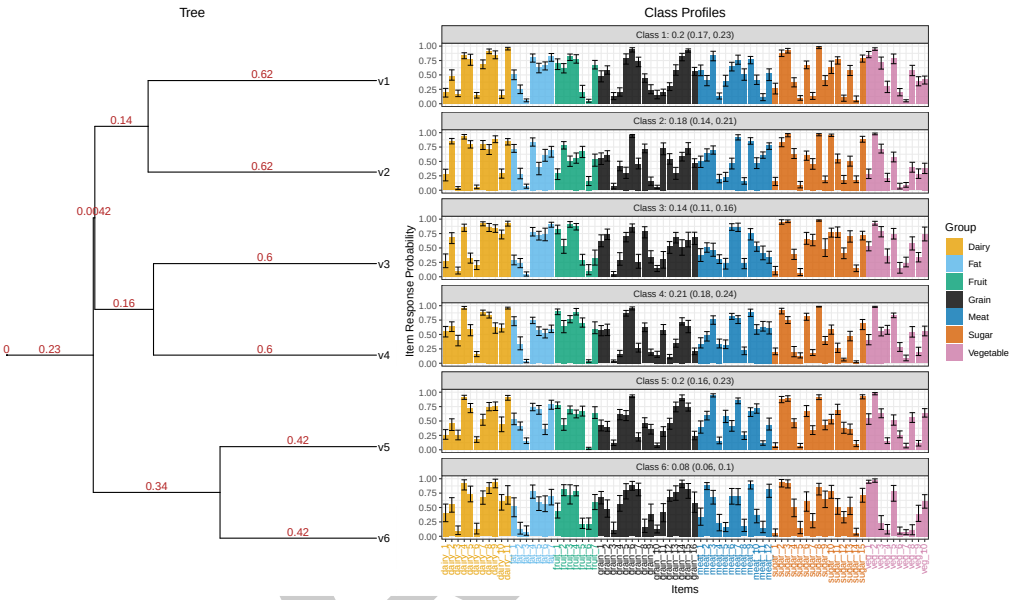

Figure 1: Posterior summary of DDT-LCM with $K = 6$ latent classes.

Interactive Illustration with RShiny

We have also developed a Shiny app accompanying the R package, accesible at (https://bolinw.shinyapps.io/ddtlcm_app/), which provides interactivity via point-and-click to allow users to explore and visualize the results of the "DDT-LCM" model implemented in our package.

The app is divided into three modes, each of which is denoted by a radio button on the left. The three modes allow users to 1) simulate data using user-specified parameters or exemplar parameters mimicking a real data set, 2) upload raw multivariate binary observed data matrix, or 3) upload posterior samples collected from pervious fit of the DDT-LCM. Users can explore the app to fully understand the properties of the model, analyze their own data, save the fitted results, and produce visualizations.

On the right hand side of the interface, three tabs are available during an actual data analysis. The "Analysis" tab visualizes tree structure over latent classes and class profiles for a set of food items grouped into major food categories, and allows users to download figures. The "Parameter" tab displays the detailed values of the estimated model parameters and users can explore the posterior distribution of model parameters. The "Data" tab shows the binary data matrix for easier examination of the analyzed data. Model fitting results can be downloaded. In addition, for simulated data, there is another tab "Truth" displaying true paramter settings. Overall, the Shiny app is a front-end interface that may enhance the accessibility and usability of the underlying "ddtlcm" package.

Conclusions

We developed an R package `ddtlcm` that addresses a long-standing weak class separation issue in latent class analysis, greatly enhancing numerical and inferential stability relative to existing popular packages. This paper offers a step-by-step example that contextualizes the workflow in a semi-syntheic nutrition data. Details about usage and more elaborate examples can be found online at (https://github.com/limengbinggz/ddtlcm).

Acknowledgements

This work is partially supported by a Michigan Institute for Data Science seed grant. The authors delare no conflicts of interest.

References

Beath, K. J. (2017). randomLCA: An r package for latent class with random effects analysis. *Journal of Statistical Software*, *81*, 1–25.

Li, M., Stephenson, B., & Wu, Z. (2023). Tree-regularized bayesian latent class analysis for improving weakly separated dietary pattern subtyping in small-sized subpopulations. *arXiv Preprint arXiv:2306.04700*.

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

Neal, R. M. (2003). Density modeling and clustering using Dirichlet diffusion trees. *Bayesian Statistics*, *7*, 619–629.

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